QorIQ LS1088A Reference Manual

Also supports LS1044A, LS1084A and LS1048A

Document Number: LS1088ARM
Rev. 1, 11/2020

Contents

Section number **Title** **Page**

Chapter 1
Overview

1.1 Introduction....227

1.2 Block diagram....227

1.3 Features summary....228

1.4 Core and CPU clusters....230

1.4.1 Cortex-A53 core features....230

1.5 Memory complex....231

1.5.1 DDR memory controller....231

1.6 Nonvolatile memory interfaces....232

1.6.1 Integrated flash controller (IFC) 2.0....232

1.6.1.1 NAND flash features....232

1.6.1.2 NOR flash features....232

1.6.1.3 General-purpose chip-select machine (GPCM)....233

1.6.2 Enhanced secure digital host controller (eSDHC)....233

1.6.3 Serial peripheral interface (SPI)....233

1.6.4 Quad serial peripheral interface (QuadSPI)....234

1.7 Universal serial bus (USB) 3.0....235

1.8 QUICC Engine....235

1.9 High-speed peripheral interface complex (HSSI)....236

1.9.1 PCI Express....236

1.9.2 Serial Advanced Technology Attachment (SATA)....236

1.10 Data Path Acceleration Architecture (DPAA2)....237

1.10.1 DPAA2 management complex....237

1.10.2 Other DPAA2 components....238

1.10.2.1 Wire rate IO processor (WRIOP)....238

1.10.2.1.1 Classifier and table lookup unit (CTLU)....240

1.10.2.1.1.1 Table lookup unit (TLU) features....241
1.10.2.1.1.2 QoS mapping and policer features....241
1.10.2.1.1.3 Parser and attributes select features....242
1.10.2.2 Advanced IO processor (AIOP)....242
1.10.2.2.1 AIOP features....243
1.10.2.3 Queue manager (QMan)....244
1.10.2.4 Buffer manager (BMan)....244
1.10.2.5 SEC....245
1.10.2.6 Queue direct memory access (qDMA)....246
1.11 Advanced power management....247
1.11.1 Transparent power management....247
1.11.2 Arm power management....248
1.12 Resource partitioning and QorIQ trust architecture....248
1.12.1 System memory management unit (SMMU)....249
1.12.2 Secure boot and sensitive data protection....249
1.12.3 Service processor....250
1.13 Debug support....251

Chapter 2
Memory Map

2.1 Memory Map Overview....253
2.2 System Memory Map....253
2.3 CCSR Address Map....255
2.4 DCSR address map....263
2.5 Source ID Assignments....263
2.6 OCRAM Initialization....270
2.7 DRAM Region Remapping....270
2.8 IFC, QSPI Address Remapping....271

Chapter 3
Signal Descriptions

3.1 Signals Introduction....273

3.2 Signals Overview....273

3.3 Output Signal States During Reset....297

Chapter 4
Reset, Clocking, and Initialization

4.1 Reset, clocking, and initialization overview....299

4.2 External signal descriptions....299

4.2.1 System control signals....300

4.2.2 External Clock Signals....301

4.3 Reset register descriptions....302

4.3.1 RST_CCSR memory map....302

4.3.2 Reset Control Register (RSTCR)....302

4.3.2.1 Offset....302

4.3.2.2 Function....303

4.3.2.3 Diagram....303

4.3.2.4 Fields....303

4.3.3 Service Processor Reset Control Register (RSTCRSP)....304

4.3.3.1 Offset....304

4.3.3.2 Function....304

4.3.3.3 Diagram....304

4.3.3.4 Fields....304

4.3.4 Reset Request Mask Register (RSTRQMR1)....305

4.3.4.1 Offset....305

4.3.4.2 Function....305

4.3.4.3 Diagram....305

4.3.4.4 Fields....306

4.3.5 Reset Request Status Register (RSTRQSR1)....307

4.3.5.1 Offset....307

4.3.5.2 Function....307

4.3.5.3 Diagram....308
4.3.5.4 Fields....308
4.3.6 Reset Request WDT Mask Register (RSTRQWDTMRL)....310
4.3.6.1 Offset....310
4.3.6.2 Function....310
4.3.6.3 Diagram....310
4.3.6.4 Fields....311
4.3.7 Reset Request WDT Status Register (RSTRQWDTSRL)....312
4.3.7.1 Offset....312
4.3.7.2 Function....312
4.3.7.3 Diagram....312
4.3.7.4 Fields....312
4.3.8 Boot Release Register (BRRL)....313
4.3.8.1 Offset....313
4.3.8.2 Function....313
4.3.8.3 Diagram....314
4.3.8.4 Fields....314
4.3.9 Core Enable Boot Release Status Register (BRCORENBR)....315
4.3.9.1 Offset....315
4.3.9.2 Function....315
4.3.9.3 Diagram....315
4.3.9.4 Fields....316
4.3.10 RCW Request Register (RCW_REQR)....316
4.3.10.1 Offset....316
4.3.10.2 Function....317
4.3.10.3 Diagram....317
4.3.10.4 Fields....317
4.3.11 RCW Completion Register (RCW_COMPLETIONR)....317
4.3.11.1 Offset....317

4.3.11.2 Function ... 318
4.3.11.3 Diagram ... 318
4.3.11.4 Fields ... 318
4.3.12 PBI Request Register (PBI_REQR) ... 318
4.3.12.1 Offset ... 318
4.3.12.2 Function ... 319
4.3.12.3 Diagram ... 319
4.3.12.4 Fields ... 319
4.3.13 PBI Completion Register (PBI_COMPLETIONR) ... 319
4.3.13.1 Offset ... 319
4.3.13.2 Function ... 320
4.3.13.3 Diagram ... 320
4.3.13.4 Fields ... 320
4.3.14 Core Reset Status Register n (CRSTSR0 - CRSTSR7) ... 321
4.3.14.1 Offset ... 321
4.3.14.2 Function ... 321
4.3.14.3 Diagram ... 321
4.3.14.4 Fields ... 322
4.3.15 Core Reset Status Register n (CRSTSR8 - CRSTSR11) ... 322
4.3.15.1 Offset ... 322
4.3.15.2 Function ... 323
4.3.15.3 Diagram ... 323
4.3.15.4 Fields ... 324
4.3.16 QMBM Warm Reset Control Register (QMBM_WARMRST) ... 325
4.3.16.1 Offset ... 325
4.3.16.2 Function ... 325
4.3.16.3 Diagram ... 326
4.3.16.4 Fields ... 326
4.3.17 SOC_IP Warm Reset Request Register (SOCIP_WARMRST) ... 326

4.3.17.1 Offset....326
4.3.17.2 Function.... 327
4.3.17.3 Diagram....327
4.3.17.4 Fields....327
4.3.18 IP Block Revision 1 Register (IP_REV1)....328
4.3.18.1 Offset....328
4.3.18.2 Function.... 328
4.3.18.3 Diagram....328
4.3.18.4 Fields....328
4.3.19 IP Block Revision 2 Register (IP_REV2)....329
4.3.19.1 Offset....329
4.3.19.2 Function.... 329
4.3.19.3 Diagram....329
4.3.19.4 Fields....329
4.4 CGU core cluster register descriptions.... 330
4.4.1 CGU Core to Hardware Accelerator (HWA) Block Cluster Group memory map.... 330
4.4.2 HWAaCSR (HWA1CSR - HWA3CSR)....330
4.4.2.1 Offset....330
4.4.2.2 Function.... 331
4.4.2.3 Diagram....331
4.4.2.4 Fields....331
4.5 CGU Platform register descriptions....332
4.5.1 CGU Platform CCSR memory map....332
4.5.2 Platform Clock Control/Status Register (CLKCSR).... 332
4.5.2.1 Offset....332
4.5.2.2 Function.... 333
4.5.2.3 Diagram....333
4.5.2.4 Fields....333
4.6 CGU DDR register descriptions.... 334

Section number	Title	Page

4.6.1 CGU DDR CCSR memory map 334

4.6.2 DDR Clock Control/Status Register (CLKCSR) 334

4.6.2.1 Offset 334

4.6.2.2 Function 334

4.6.2.3 Diagram 335

4.6.2.4 Fields 335

4.7 CORE CLUSTER UNIT CCU register descriptions 335

4.7.1 CORE CLUSTER UNIT CCU memory map 335

4.7.2 Core Cluster a Clock Control/Status Register (CLKC1CSR - CLKC2CSR) 336

4.7.2.1 Offset 336

4.7.2.2 Function 336

4.7.2.3 Diagram 336

4.7.2.4 Fields 336

4.8 Functional description 337

4.8.1 Reset Operations 337

4.8.2 Reset Actions 338

4.8.3 Sources of Reset Requests 339

4.8.4 Power-On Reset Sequence Overview 339

4.8.5 Power-On Reset Sequence Detailed Description 340

4.8.6 Power-On Reset Sequence Timing 343

4.8.7 Hard Reset Sequence (HRESET_B) 344

4.8.7.1 Selecting the RCW Source and "Flash Mode" 344

4.8.7.2 NOR Flash Configuration for RCW and Boot 346

4.8.7.3 RCW, PBI, and Boot Options 347

4.8.7.4 Power-on reset configuration 348

4.8.7.5 General-purpose input 349

4.8.7.6 IFC external transceiver enable polarity select 349

4.8.7.7 RCW Loading Phase - Service Processor Brings in RCW Data 349

4.8.7.8 Reset Control Word (RCW) Field Descriptions 350

4.8.7.8.1 RCW ... 350
4.8.7.9 Hard Coded RCW Options ... 375
4.8.7.9.1 RCW settings for hard coded options ... 375
4.8.8 IP Logic Clock Distribution and Configuration ... 382

Chapter 5
Interrupt Assignments

5.1 Introduction ... 387
5.2 Internal Interrupt Sources ... 387
5.3 ISC register descriptions ... 394
5.3.1 ISC memory map ... 394
5.3.2 External Interrupt Pins Status Register (IRQPSR) ... 394
5.3.2.1 Offset ... 394
5.3.2.2 Function ... 395
5.3.2.3 Diagram ... 395
5.3.2.4 Fields ... 395
5.3.3 External Interrupt Control Register (IRQCR) ... 397
5.3.3.1 Offset ... 397
5.3.3.2 Function ... 398
5.3.3.3 Diagram ... 398
5.3.3.4 Fields ... 398
5.3.4 Thread WatchDog Reset Timer Status Register (TWDTSR0) ... 401
5.3.4.1 Offset ... 401
5.3.4.2 Function ... 401
5.3.4.3 Diagram ... 401
5.3.4.4 Fields ... 401
5.3.5 Thread WatchDog Reset Timer Control Register (TWDTCR) ... 404
5.3.5.1 Offset ... 404
5.3.5.2 Function ... 404
5.3.5.3 Diagram ... 404

5.3.5.4 Fields....405

5.3.6 SoC WatchDog Reset Timer Status Register (SWDTSR)....405

5.3.6.1 Offset....405

5.3.6.2 Function....405

5.3.6.3 Diagram....405

5.3.6.4 Fields....406

Chapter 6
Arm Modules

6.1 Introduction....409

6.2 Arm® Cortex®-A53....410

6.3 Arm generic interrupt controller (GIC)....411

6.4 System memory management unit (MMU-500)....412

6.5 Arm Generic Timer (Global Generic Reference Timer)....414

Chapter 7
Service Processor

7.1 General Format of Pre-Boot Initialization Image....415

7.1.1 PBI Command List....415

7.1.2 Header Format for a CCSR Configuration Write....416

7.1.3 Header Format for an AltConfig Write Command....416

7.1.4 Format of General Commands....417

7.1.5 Use of Addresses in PBI Commands....417

7.1.6 CRC Checking of PBI Commands....418

7.2 PBI Command Detailed Descriptions....418

7.2.1 Preamble....419

7.2.2 Configuration Writes....419

7.2.2.1 CCSR Write....419

7.2.2.2 AltConfig Write....421

7.2.3 Block Copy Command....422

7.2.4 Load RCW....423

7.2.5 Load Alternate Configuration Window.... 424
7.2.6 Load Condition.... 426
7.2.7 Load Security Header.... 426
7.2.8 Load Boot 1 CSF Header Pointer.... 427
7.2.9 Poll.... 427
7.2.10 Wait.... 428
7.2.11 Jump.... 429
7.2.12 Jump Conditional.... 429
7.2.13 CRC and Stop.... 430
7.2.14 Stop.... 430
7.3 Error Codes.... 431

Chapter 8
Device Configuration

8.1 Introduction.... 435
8.2 Features.... 435
8.3 Device Config register descriptions.... 435
8.3.1 DCFG_ccsr memory map.... 436
8.3.2 POR Status Register 1 (PORSR1).... 437
8.3.2.1 Offset.... 438
8.3.2.2 Function.... 438
8.3.2.3 Diagram.... 438
8.3.2.4 Fields.... 438
8.3.3 POR Status Register 2 (PORSR2).... 439
8.3.3.1 Offset.... 439
8.3.3.2 Function.... 439
8.3.3.3 Diagram.... 440
8.3.3.4 Fields.... 440
8.3.4 General-Purpose POR Configuration Register (GPPORCR1).... 440
8.3.4.1 Offset.... 440

8.3.4.2 Function 441
8.3.4.3 Diagram 441
8.3.4.4 Fields 441
8.3.5 Fuse Status Register (FUSESR) 441
8.3.5.1 Offset 441
8.3.5.2 Function 442
8.3.5.3 Diagram 442
8.3.5.4 Fields 442
8.3.6 Device Disable Register 1 (DEVDISR1) 443
8.3.6.1 Offset 443
8.3.6.2 Function 443
8.3.6.3 Diagram 444
8.3.6.4 Fields 444
8.3.7 Device Disable Register 2 (DEVDISR2) 445
8.3.7.1 Offset 445
8.3.7.2 Function 445
8.3.7.3 Diagram 446
8.3.7.4 Fields 446
8.3.8 Device Disable Register 3 (DEVDISR3) 447
8.3.8.1 Offset 447
8.3.8.2 Function 447
8.3.8.3 Diagram 447
8.3.8.4 Fields 447
8.3.9 Device Disable Register 4 (DEVDISR4) 448
8.3.9.1 Offset 448
8.3.9.2 Function 448
8.3.9.3 Diagram 449
8.3.9.4 Fields 449
8.3.10 Device Disable Register 5 (DEVDISR5) 450

8.3.10.1 Offset....450
8.3.10.2 Function.... 450
8.3.10.3 Diagram....450
8.3.10.4 Fields....451
8.3.11 Device Disable Register 6 (DEVDISR6)....452
8.3.11.1 Offset....452
8.3.11.2 Function.... 452
8.3.11.3 Diagram....452
8.3.11.4 Fields....453
8.3.12 Core Disable Register (COREDISR)....453
8.3.12.1 Offset....453
8.3.12.2 Function.... 453
8.3.12.3 Diagram....454
8.3.12.4 Fields....454
8.3.13 System Version Register (SVR).... 455
8.3.13.1 Offset....455
8.3.13.2 Function.... 455
8.3.13.3 Diagram....455
8.3.13.4 Fields....456
8.3.14 Reset Configuration Word Status Register 1 (RCWSR1).... 457
8.3.14.1 Offset....457
8.3.14.2 Function.... 457
8.3.14.3 Diagram....457
8.3.14.4 Fields....458
8.3.15 Reset Configuration Word Status Register 2 (RCWSR2).... 458
8.3.15.1 Offset....458
8.3.15.2 Function.... 459
8.3.15.3 Diagram....459
8.3.15.4 Fields....459

Section number	Title	Page

8.3.16 Reset Configuration Word Status Register 3 (RCWSR3) 460

8.3.16.1 Offset 460

8.3.16.2 Function 460

8.3.16.3 Diagram 460

8.3.16.4 Fields 460

8.3.17 Reset Configuration Word Status Register 4 (RCWSR4) 461

8.3.17.1 Offset 461

8.3.17.2 Function 461

8.3.17.3 Diagram 461

8.3.17.4 Fields 462

8.3.18 Reset Configuration Word Status Register 5 (RCWSR5) 462

8.3.18.1 Offset 462

8.3.18.2 Function 463

8.3.18.3 Diagram 463

8.3.18.4 Fields 463

8.3.19 Reset Configuration Word Status Register 6 (RCWSR6) 464

8.3.19.1 Offset 464

8.3.19.2 Function 465

8.3.19.3 Diagram 465

8.3.19.4 Fields 465

8.3.20 Reset Configuration Word Status Register 7 (RCWSR7) 466

8.3.20.1 Offset 466

8.3.20.2 Function 466

8.3.20.3 Diagram 466

8.3.20.4 Fields 467

8.3.21 Reset Configuration Word Status Register 8 (RCWSR8) 468

8.3.21.1 Offset 468

8.3.21.2 Function 468

8.3.21.3 Diagram 468

8.3.21.4 Fields....469
8.3.22 Reset Configuration Word Status Register 9 (RCWSR9).... 469
8.3.22.1 Offset....470
8.3.22.2 Function.... 470
8.3.22.3 Diagram....470
8.3.22.4 Fields....470
8.3.23 Reset Configuration Word Status Register 10 (RCWSR10).... 471
8.3.23.1 Offset....471
8.3.23.2 Function.... 472
8.3.23.3 Diagram....472
8.3.23.4 Fields....472
8.3.24 Reset Configuration Word Status Register 11 (RCWSR11).... 473
8.3.24.1 Offset....473
8.3.24.2 Function.... 473
8.3.24.3 Diagram....473
8.3.24.4 Fields....474
8.3.25 Reset Configuration Word Status Register 12 (RCWSR12).... 474
8.3.25.1 Offset....474
8.3.25.2 Function.... 474
8.3.25.3 Diagram....474
8.3.25.4 Fields....475
8.3.26 Reset Configuration Word Status Register 13 (RCWSR13).... 476
8.3.26.1 Offset....476
8.3.26.2 Function.... 476
8.3.26.3 Diagram....476
8.3.26.4 Fields....477
8.3.27 Reset Configuration Word Status Register 14 (RCWSR14).... 479
8.3.27.1 Offset....479
8.3.27.2 Function.... 479

8.3.27.3 Diagram....479
8.3.27.4 Fields....480
8.3.28 Reset Configuration Word Status Register 15 (RCWSR15)....481
8.3.28.1 Offset....481
8.3.28.2 Function....481
8.3.28.3 Diagram....481
8.3.28.4 Fields....482
8.3.29 Reset Configuration Word Status Register 16 (RCWSR16)....483
8.3.29.1 Offset....483
8.3.29.2 Function....483
8.3.29.3 Diagram....483
8.3.29.4 Fields....484
8.3.30 Reset Configuration Word Status Register 17 (RCWSR17)....485
8.3.30.1 Offset....485
8.3.30.2 Function....485
8.3.30.3 Diagram....485
8.3.30.4 Fields....485
8.3.31 Reset Configuration Word Status Register 18 (RCWSR18)....485
8.3.31.1 Offset....486
8.3.31.2 Function....486
8.3.31.3 Diagram....486
8.3.31.4 Fields....486
8.3.32 Reset Configuration Word Status Register 19 (RCWSR19)....486
8.3.32.1 Offset....486
8.3.32.2 Function....487
8.3.32.3 Diagram....487
8.3.32.4 Fields....487
8.3.33 Reset Configuration Word Status Register 20 (RCWSR20)....487
8.3.33.1 Offset....487

8.3.33.2 Function ... 488
8.3.33.3 Diagram ... 488
8.3.33.4 Fields ... 488
8.3.34 Reset Configuration Word Status Register 21 (RCWSR21) ... 488
8.3.34.1 Offset ... 488
8.3.34.2 Function ... 489
8.3.34.3 Diagram ... 489
8.3.34.4 Fields ... 489
8.3.35 Reset Configuration Word Status Register 22 (RCWSR22) ... 489
8.3.35.1 Offset ... 489
8.3.35.2 Function ... 490
8.3.35.3 Diagram ... 490
8.3.35.4 Fields ... 490
8.3.36 Reset Configuration Word Status Register 23 (RCWSR23) ... 490
8.3.36.1 Offset ... 490
8.3.36.2 Function ... 491
8.3.36.3 Diagram ... 491
8.3.36.4 Fields ... 491
8.3.37 Reset Configuration Word Status Register 24 (RCWSR24) ... 491
8.3.37.1 Offset ... 491
8.3.37.2 Function ... 492
8.3.37.3 Diagram ... 492
8.3.37.4 Fields ... 492
8.3.38 Reset Configuration Word Status Register 25 (RCWSR25) ... 492
8.3.38.1 Offset ... 492
8.3.38.2 Function ... 493
8.3.38.3 Diagram ... 493
8.3.38.4 Fields ... 493
8.3.39 Reset Configuration Word Status Register 26 (RCWSR26) ... 493

8.3.39.1 Offset....493
8.3.39.2 Function.... 494
8.3.39.3 Diagram....494
8.3.39.4 Fields....494
8.3.40 Reset Configuration Word Status Register 27 (RCWSR27).... 494
8.3.40.1 Offset....494
8.3.40.2 Function.... 495
8.3.40.3 Diagram....495
8.3.40.4 Fields....495
8.3.41 Reset Configuration Word Status Register 28 (RCWSR28).... 496
8.3.41.1 Offset....496
8.3.41.2 Function.... 496
8.3.41.3 Diagram....496
8.3.41.4 Fields....497
8.3.42 Reset Configuration Word Status Register 29 (RCWSR29).... 497
8.3.42.1 Offset....497
8.3.42.2 Function.... 497
8.3.42.3 Diagram....498
8.3.42.4 Fields....498
8.3.43 Reset Configuration Word Status Register 30 (RCWSR30).... 499
8.3.43.1 Offset....499
8.3.43.2 Function.... 499
8.3.43.3 Diagram....499
8.3.43.4 Fields....500
8.3.44 Reset Configuration Word Status Register 31 (RCWSR31).... 501
8.3.44.1 Offset....501
8.3.44.2 Function.... 502
8.3.44.3 Diagram....502
8.3.44.4 Fields....502

Section number	Title	Page

8.3.45 Reset Configuration Word Status Register 32 (RCWSR32)..... 502

8.3.45.1 Offset.....502

8.3.45.2 Function..... 503

8.3.45.3 Diagram.....503

8.3.45.4 Fields.....503

8.3.46 Scratch Read / Write Register n (SCRATCHRW1 - SCRATCHRW11)..... 503

8.3.46.1 Offset.....503

8.3.46.2 Function..... 504

8.3.46.3 Diagram.....504

8.3.46.4 Fields.....504

8.3.47 Scratch Read / Write Register n (SCRATCHRW12)..... 504

8.3.47.1 Offset.....504

8.3.47.2 Function..... 505

8.3.47.3 Diagram.....505

8.3.47.4 Fields.....505

8.3.48 Scratch Read / Write Register n (SCRATCHRW13 - SCRATCHRW16)..... 505

8.3.48.1 Offset.....505

8.3.48.2 Function..... 506

8.3.48.3 Diagram.....506

8.3.48.4 Fields.....506

8.3.49 Scratch Read Register n (SCRATCHW1R1 - SCRATCHW1R4)..... 506

8.3.49.1 Offset.....506

8.3.49.2 Function..... 507

8.3.49.3 Diagram.....507

8.3.49.4 Fields.....507

8.3.50 Boot location pointer low-order address (BOOTLOCPTRL).....507

8.3.50.1 Offset.....507

8.3.50.2 Function..... 508

8.3.50.3 Diagram.....508

8.3.50.4 Fields....508
8.3.51 Boot location pointer high-order address (BOOTLOCPTRH)....508
8.3.51.1 Offset....508
8.3.51.2 Function....509
8.3.51.3 Diagram....509
8.3.51.4 Fields....509
8.3.52 USBa_AMQR Register (USB1_AMQR - USB2_AMQR)....509
8.3.52.1 Offset....509
8.3.52.2 Function....510
8.3.52.3 Diagram....510
8.3.52.4 Fields....510
8.3.53 SDMM1_AMQR Register (SDMM1_AMQR)....511
8.3.53.1 Offset....511
8.3.53.2 Function....511
8.3.53.3 Diagram....511
8.3.53.4 Fields....512
8.3.54 SATA n Access Management Qualifier register (SATA1_AMQR)....512
8.3.54.1 Offset....513
8.3.54.2 Function....513
8.3.54.3 Diagram....513
8.3.54.4 Fields....513
8.3.55 MISC n Access Management Qualifier register (MISC1_AMQR - MISC3_AMQR)....514
8.3.55.1 Offset....514
8.3.55.2 Function....514
8.3.55.3 Diagram....514
8.3.55.4 Fields....515
8.3.56 SPARE n Access Management Qualifier register (SPARE1_AMQR - SPARE2_AMQR)....515
8.3.56.1 Offset....516
8.3.56.2 Function....516

8.3.56.3 Diagram....516
8.3.56.4 Fields....516
8.3.57 Topology Initiator Type n Register (TP_ITYP0 - TP_ITYP5)....517
8.3.57.1 Offset....517
8.3.57.2 Function....517
8.3.57.3 Diagram....517
8.3.57.4 Fields....518
8.3.58 Core Cluster n Topology Register (TP_CLUSTER1L - TP_CLUSTER2L)....518
8.3.58.1 Offset....518
8.3.58.2 Function....518
8.3.58.3 Diagram....519
8.3.58.4 Fields....519
8.3.59 QSPI Control Register (QSPICR1)....520
8.3.59.1 Offset....520
8.3.59.2 Diagram....520
8.3.59.3 Fields....520
8.3.60 I/O Quality of Services Register (IOQOSCR1 - IOQOSCR8)....521
8.3.60.1 Offset....521
8.3.60.2 Function....521
8.3.60.3 Diagram....521
8.3.60.4 Fields....522
8.3.61 FlexTimer Control Register (FTMCR)....522
8.3.61.1 Offset....522
8.3.61.2 Function....523
8.3.61.3 Diagram....523
8.3.61.4 Fields....523
8.3.62 Core Disable Status Register (COREDISABLESR)....524
8.3.62.1 Offset....524
8.3.62.2 Function....524

8.3.62.3 Diagram....524
8.3.62.4 Fields....524
8.3.63 Secure Debug configuration register (SDBGCR)....525
8.3.63.1 Offset....525
8.3.63.2 Function....525
8.3.63.3 Diagram....526
8.3.63.4 Fields....526
8.3.64 IP Block Revision Register 1 (IPBRR1)....526
8.3.64.1 Offset....526
8.3.64.2 Function....526
8.3.64.3 Diagram....527
8.3.64.4 Fields....527
8.3.65 IP Block Revision Register 2 (IPBRR2)....527
8.3.65.1 Offset....527
8.3.65.2 Function....527
8.3.65.3 Diagram....528
8.3.65.4 Fields....528
8.4 Pin Control Analog Register Descriptions....528
8.4.1 Pin Control Analog Memory Map....528
8.4.2 Hyst Analog Control 1 (hyst_ana_cntrl1)....529
8.4.2.1 Offset....529
8.4.2.2 Diagram....529
8.4.2.3 Fields....529
8.4.3 Hyst Analog Control 2 (hyst_ana_cntrl2)....532
8.4.3.1 Offset....532
8.4.3.2 Diagram....532
8.4.3.3 Fields....533
8.4.4 Hyst Analog Control 3 (hyst_ana_cntrl3)....534
8.4.4.1 Offset....534

8.4.4.2 Diagram....534
8.4.4.3 Fields....535
8.4.5 VSEL Control 1 (vsel_cntrl1)....538
8.4.5.1 Offset....538
8.4.5.2 Diagram....538
8.4.5.3 Fields....538
8.5 Pin Control Memory Map/Register Definition....539
8.5.1 DDR Clock Disable Register (DCFG_PIN_CTRL_PIN_CTRL_DDRCLKDR)....540
8.5.2 IFC Clock Disable Register (DCFG_PIN_CTRL_PIN_CTRL_IFCCLKDR)....541

Chapter 9
Data Path Acceleration Architecture (DPAA2) Overview and Implementation

9.1 LS1088A-Specific DPAA Implementation Details....543
9.1.1 Queue Buffer Manager (QBMan) Implementation....543
9.1.2 Advanced Input Output Processor (AIOP) Implementation....544
9.1.3 Ethernet Wire Rate IO Processor (eWRIOP) Implementation....545
9.1.4 Security and Encryption Engine (SEC) Implementation....546
9.1.5 Queue direct memory access (qDMA)....548

Chapter 10
QUICC Engine Block

10.1 Introduction....551
10.2 QUICC Engine block....551
10.3 QUICC Engine implementation details....552
10.3.1 System interface....554
10.3.1.1 System interface-data paths....555
10.3.1.2 System interface-interrupt configuration....556
10.3.1.3 System interface-bus arbitration....557
10.3.2 Configuration....557
10.3.2.1 Configuration-parameter RAM....557
10.3.3 Multiplexing and timers....558

10.3.3.1 Multiplexing and timers-NMSI configuration.... 558

10.3.3.2 Multiplexing and timers-baud-rate generators (BRGs), BRG configuration registers 1-4 (BRGCn).... 560

10.3.4 Unified communications controllers (UCCs).... 562

10.3.4.1 Unified communications controllers (UCCs)-UCC page base address.... 562

10.3.5 HDLC controller.... 562

10.3.5.1 HDLC controller-introduction.... 563

10.3.6 Transparent controller.... 563

10.3.7 UCC for fast protocols.... 563

10.3.8 UCC for slow protocols.... 563

10.3.9 UART mode and asynchronous HDLC.... 563

10.3.10 BISYNC mode.... 564

10.3.11 Serial interface with time-slot assigner.... 564

10.3.12 Multi-channel controller on UCC (UMCC).... 564

Chapter 11
Cache Coherent Interconnect

11.1 The CCI-400 module as implemented on the chip.... 565

11.1.1 LS1088A CCI module integration.... 565

11.1.2 Connections to the CCI-400 interconnect.... 565

11.1.3 LS1088A transaction shareability configuration.... 567

11.2 CCI400 register descriptions.... 567

11.2.1 CCI400 Registers memory map.... 567

11.2.2 Control Override Register (Control_Override_Register).... 568

11.2.2.1 Offset.... 568

11.2.2.2 Function.... 568

11.2.2.3 Diagram.... 568

11.2.2.4 Fields.... 569

11.2.3 Speculation Control Register (Speculation_Control_Register).... 570

11.2.3.1 Offset.... 570

11.2.3.2 Function.... 570
11.2.3.3 Diagram....570
11.2.3.4 Fields....571
11.2.4 Secure Access Register (Secure_Access_Register)....572
11.2.4.1 Offset....572
11.2.4.2 Function.... 572
11.2.4.3 Diagram....572
11.2.4.4 Fields....573
11.2.5 Status Register (Status_Register)....573
11.2.5.1 Offset....573
11.2.5.2 Function.... 573
11.2.5.3 Diagram....574
11.2.5.4 Fields....574
11.2.6 Imprecise Error Register (Imprecise_Error_Register)....575
11.2.6.1 Offset....575
11.2.6.2 Function.... 575
11.2.6.3 Diagram....575
11.2.6.4 Fields....576
11.2.7 Snoop Control Registers (Snoop_Control_Register_S0 - Snoop_Control_Register_S4).... 576
11.2.7.1 Offset....576
11.2.7.2 Function.... 577
11.2.7.3 Diagram....577
11.2.7.4 Fields....578
11.2.8 Shareable Override Registers (Shareable_Override_Register_S0 - Shareable_Override_Register_S4).....579
11.2.8.1 Offset....579
11.2.8.2 Function.... 579
11.2.8.3 Diagram....580
11.2.8.4 Fields....580
11.3 Functional Description....580

11.3.1 About the functions....581
11.3.2 Snoop connectivity and control....581
11.3.2.1 Removing a master from the coherent domain....582
11.3.3 Speculative fetch....582
11.3.4 Security....583
11.3.4.1 Internal programmers view....583
11.3.4.2 Security of master interfaces....584
11.3.5 Error responses....584
11.3.5.1 Imprecise errors....584
11.3.6 Barriers....585
11.3.7 DVM messages....585

Chapter 12
DDR Memory Controller

12.1 DDR Introduction....587
12.2 DDR Features....588
12.2.1 DDR Modes of Operation....589
12.3 DDR External Signal Descriptions....589
12.3.1 DDR Signals Overview....589
12.3.2 DDR Detailed Signal Descriptions....591
12.3.2.1 Memory Interface Signals....591
12.3.2.2 Clock Interface Signals....596
12.4 DDR register descriptions....596
12.4.1 DDR memory map....597
12.4.2 Chip select a memory bounds (CS0_BNDS - CS3_BNDS)....599
12.4.2.1 Offset....599
12.4.2.2 Function....599
12.4.2.3 Diagram....599
12.4.2.4 Fields....600
12.4.3 Chip select a configuration (CS0_CONFIG - CS3_CONFIG)....600

12.4.3.1 Offset....600
12.4.3.2 Function....600
12.4.3.3 Diagram....601
12.4.3.4 Fields....601
12.4.4 DDR SDRAM timing configuration 3 (TIMING_CFG_3)....603
12.4.4.1 Offset....603
12.4.4.2 Function....603
12.4.4.3 Diagram....603
12.4.4.4 Fields....604
12.4.5 DDR SDRAM timing configuration 0 (TIMING_CFG_0)....606
12.4.5.1 Offset....606
12.4.5.2 Function....606
12.4.5.3 Diagram....606
12.4.5.4 Fields....607
12.4.6 DDR SDRAM timing configuration 1 (TIMING_CFG_1)....610
12.4.6.1 Offset....610
12.4.6.2 Function....610
12.4.6.3 Diagram....610
12.4.6.4 Fields....610
12.4.7 DDR SDRAM timing configuration 2 (TIMING_CFG_2)....613
12.4.7.1 Offset....613
12.4.7.2 Function....613
12.4.7.3 Diagram....614
12.4.7.4 Fields....614
12.4.8 DDR SDRAM control configuration (DDR_SDRAM_CFG)....616
12.4.8.1 Offset....616
12.4.8.2 Function....617
12.4.8.3 Diagram....617
12.4.8.4 Fields....617

12.4.9 DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)....620
12.4.9.1 Offset....620
12.4.9.2 Function....620
12.4.9.3 Diagram....620
12.4.9.4 Fields....621
12.4.10 DDR SDRAM mode configuration (DDR_SDRAM_MODE)....623
12.4.10.1 Offset....623
12.4.10.2 Function....624
12.4.10.3 Diagram....624
12.4.10.4 Fields....624
12.4.11 DDR SDRAM mode configuration 2 (DDR_SDRAM_MODE_2)....624
12.4.11.1 Offset....625
12.4.11.2 Function....625
12.4.11.3 Diagram....625
12.4.11.4 Fields....625
12.4.12 DDR SDRAM mode control (DDR_SDRAM_MD_CNTL)....626
12.4.12.1 Offset....626
12.4.12.2 Function....626
12.4.12.3 Diagram....627
12.4.12.4 Fields....627
12.4.13 DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL)....629
12.4.13.1 Offset....629
12.4.13.2 Function....629
12.4.13.3 Diagram....629
12.4.13.4 Fields....630
12.4.14 DDR SDRAM data initialization (DDR_DATA_INIT)....630
12.4.14.1 Offset....630
12.4.14.2 Function....630
12.4.14.3 Diagram....631

12.4.14.4 Fields....631
12.4.15 DDR SDRAM clock control (DDR_SDRAM_CLK_CNTL)....631
12.4.15.1 Offset....631
12.4.15.2 Function....631
12.4.15.3 Diagram....632
12.4.15.4 Fields....632
12.4.16 DDR training initialization address (DDR_INIT_ADDR)....633
12.4.16.1 Offset....633
12.4.16.2 Function....633
12.4.16.3 Diagram....633
12.4.16.4 Fields....634
12.4.17 DDR training initialization extended address (DDR_INIT_EXT_ADDRESS)....634
12.4.17.1 Offset....634
12.4.17.2 Function....634
12.4.17.3 Diagram....634
12.4.17.4 Fields....635
12.4.18 DDR SDRAM timing configuration 4 (TIMING_CFG_4)....635
12.4.18.1 Offset....635
12.4.18.2 Function....635
12.4.18.3 Diagram....636
12.4.18.4 Fields....636
12.4.19 DDR SDRAM timing configuration 5 (TIMING_CFG_5)....639
12.4.19.1 Offset....639
12.4.19.2 Function....639
12.4.19.3 Diagram....639
12.4.19.4 Fields....640
12.4.20 DDR SDRAM timing configuration 7 (TIMING_CFG_7)....641
12.4.20.1 Offset....641
12.4.20.2 Function....641

12.4.20.3 Diagram....642
12.4.20.4 Fields....642
12.4.21 DDR ZQ calibration control (DDR_ZQ_CNTL)....644
12.4.21.1 Offset....644
12.4.21.2 Function....644
12.4.21.3 Diagram....645
12.4.21.4 Fields....645
12.4.22 DDR write leveling control (DDR_WRLVL_CNTL)....647
12.4.22.1 Offset....647
12.4.22.2 Function....647
12.4.22.3 Diagram....647
12.4.22.4 Fields....647
12.4.23 DDR Self Refresh Counter (DDR_SR_CNTR)....650
12.4.23.1 Offset....650
12.4.23.2 Function....650
12.4.23.3 Diagram....650
12.4.23.4 Fields....651
12.4.24 DDR Register Control Words 1 (DDR_SDRAM_RCW_1)....651
12.4.24.1 Offset....651
12.4.24.2 Function....652
12.4.24.3 Diagram....652
12.4.24.4 Fields....652
12.4.25 DDR Register Control Words 2 (DDR_SDRAM_RCW_2)....653
12.4.25.1 Offset....653
12.4.25.2 Function....653
12.4.25.3 Diagram....653
12.4.25.4 Fields....654
12.4.26 DDR write leveling control 2 (DDR_WRLVL_CNTL_2)....654
12.4.26.1 Offset....654

12.4.26.2 Function 654
12.4.26.3 Diagram 655
12.4.26.4 Fields 655
12.4.27 DDR write leveling control 3 (DDR_WRLVL_CNTL_3) 658
12.4.27.1 Offset 658
12.4.27.2 Function 658
12.4.27.3 Diagram 658
12.4.27.4 Fields 659
12.4.28 DDR Register Control Words 3 (DDR_SDRAM_RCW_3) 661
12.4.28.1 Offset 661
12.4.28.2 Function 662
12.4.28.3 Diagram 662
12.4.28.4 Fields 662
12.4.29 DDR Register Control Words 4 (DDR_SDRAM_RCW_4) 663
12.4.29.1 Offset 663
12.4.29.2 Function 663
12.4.29.3 Diagram 663
12.4.29.4 Fields 663
12.4.30 DDR Register Control Words 5 (DDR_SDRAM_RCW_5) 664
12.4.30.1 Offset 664
12.4.30.2 Function 664
12.4.30.3 Diagram 664
12.4.30.4 Fields 664
12.4.31 DDR Register Control Words 6 (DDR_SDRAM_RCW_6) 665
12.4.31.1 Offset 665
12.4.31.2 Function 665
12.4.31.3 Diagram 665
12.4.31.4 Fields 666
12.4.32 DDR SDRAM mode configuration 3 (DDR_SDRAM_MODE_3) 666

12.4.32.1 Offset....666
12.4.32.2 Function.... 666
12.4.32.3 Diagram....667
12.4.32.4 Fields....667
12.4.33 DDR SDRAM mode configuration 4 (DDR_SDRAM_MODE_4).... 668
12.4.33.1 Offset....668
12.4.33.2 Function.... 668
12.4.33.3 Diagram....668
12.4.33.4 Fields....668
12.4.34 DDR SDRAM mode configuration 5 (DDR_SDRAM_MODE_5).... 669
12.4.34.1 Offset....669
12.4.34.2 Function.... 669
12.4.34.3 Diagram....669
12.4.34.4 Fields....670
12.4.35 DDR SDRAM mode configuration 6 (DDR_SDRAM_MODE_6).... 670
12.4.35.1 Offset....670
12.4.35.2 Function.... 670
12.4.35.3 Diagram....671
12.4.35.4 Fields....671
12.4.36 DDR SDRAM mode configuration 7 (DDR_SDRAM_MODE_7).... 671
12.4.36.1 Offset....671
12.4.36.2 Function.... 672
12.4.36.3 Diagram....672
12.4.36.4 Fields....672
12.4.37 DDR SDRAM mode configuration 8 (DDR_SDRAM_MODE_8).... 673
12.4.37.1 Offset....673
12.4.37.2 Function.... 673
12.4.37.3 Diagram....673
12.4.37.4 Fields....673

Section number	Title	Page

12.4.38 DDR SDRAM mode configuration 9 (DDR_SDRAM_MODE_9)..... 674

12.4.38.1 Offset.....674

12.4.38.2 Function..... 674

12.4.38.3 Diagram.....674

12.4.38.4 Fields.....675

12.4.39 DDR SDRAM mode configuration 10 (DDR_SDRAM_MODE_10)..... 675

12.4.39.1 Offset.....675

12.4.39.2 Function..... 675

12.4.39.3 Diagram.....676

12.4.39.4 Fields.....676

12.4.40 DDR SDRAM mode configuration 11 (DDR_SDRAM_MODE_11)..... 676

12.4.40.1 Offset.....676

12.4.40.2 Function..... 677

12.4.40.3 Diagram.....677

12.4.40.4 Fields.....677

12.4.41 DDR SDRAM mode configuration 12 (DDR_SDRAM_MODE_12)..... 678

12.4.41.1 Offset.....678

12.4.41.2 Function..... 678

12.4.41.3 Diagram.....678

12.4.41.4 Fields.....678

12.4.42 DDR SDRAM mode configuration 13 (DDR_SDRAM_MODE_13)..... 679

12.4.42.1 Offset.....679

12.4.42.2 Function..... 679

12.4.42.3 Diagram.....679

12.4.42.4 Fields.....680

12.4.43 DDR SDRAM mode configuration 14 (DDR_SDRAM_MODE_14)..... 680

12.4.43.1 Offset.....680

12.4.43.2 Function..... 681

12.4.43.3 Diagram.....681

12.4.43.4 Fields....681
12.4.44 DDR SDRAM mode configuration 15 (DDR_SDRAM_MODE_15)....681
12.4.44.1 Offset....682
12.4.44.2 Function....682
12.4.44.3 Diagram....682
12.4.44.4 Fields....682
12.4.45 DDR SDRAM mode configuration 16 (DDR_SDRAM_MODE_16)....683
12.4.45.1 Offset....683
12.4.45.2 Function....683
12.4.45.3 Diagram....683
12.4.45.4 Fields....683
12.4.46 DDR SDRAM timing configuration 8 (TIMING_CFG_8)....684
12.4.46.1 Offset....684
12.4.46.2 Function....684
12.4.46.3 Diagram....684
12.4.46.4 Fields....685
12.4.47 DDR SDRAM control configuration 3 (DDR_SDRAM_CFG_3)....688
12.4.47.1 Offset....688
12.4.47.2 Function....689
12.4.47.3 Diagram....689
12.4.47.4 Fields....689
12.4.48 DQ mapping register 0 (DDR_DQ_MAP0)....692
12.4.48.1 Offset....692
12.4.48.2 Function....692
12.4.48.3 Diagram....694
12.4.48.4 Fields....694
12.4.49 DQ mapping register 1 (DDR_DQ_MAP1)....695
12.4.49.1 Offset....695
12.4.49.2 Function....695

12.4.49.3 Diagram....695
12.4.49.4 Fields....696
12.4.50 DQ mapping register 2 (DDR_DQ_MAP2)....696
12.4.50.1 Offset....696
12.4.50.2 Function....696
12.4.50.3 Diagram....696
12.4.50.4 Fields....697
12.4.51 DQ mapping register 3 (DDR_DQ_MAP3)....697
12.4.51.1 Offset....697
12.4.51.2 Function....698
12.4.51.3 Diagram....698
12.4.51.4 Fields....698
12.4.52 DDR Debug Status Register 1 (DDRDSR_1)....698
12.4.52.1 Offset....698
12.4.52.2 Function....699
12.4.52.3 Diagram....699
12.4.52.4 Fields....699
12.4.53 DDR Debug Status Register 2 (DDRDSR_2)....700
12.4.53.1 Offset....700
12.4.53.2 Function....700
12.4.53.3 Diagram....700
12.4.53.4 Fields....700
12.4.54 DDR Control Driver Register 1 (DDRCDR_1)....701
12.4.54.1 Offset....701
12.4.54.2 Function....701
12.4.54.3 Diagram....702
12.4.54.4 Fields....702
12.4.55 DDR Control Driver Register 2 (DDRCDR_2)....703
12.4.55.1 Offset....703

12.4.55.2 Function ... 703
12.4.55.3 Diagram ... 704
12.4.55.4 Fields ... 704
12.4.56 DDR IP block revision 1 (DDR_IP_REV1) ... 705
12.4.56.1 Offset ... 705
12.4.56.2 Function ... 706
12.4.56.3 Diagram ... 706
12.4.56.4 Fields ... 706
12.4.57 DDR IP block revision 2 (DDR_IP_REV2) ... 706
12.4.57.1 Offset ... 706
12.4.57.2 Function ... 707
12.4.57.3 Diagram ... 707
12.4.57.4 Fields ... 707
12.4.58 DDR Memory Test Control Register (DDR_MTCR) ... 707
12.4.58.1 Offset ... 708
12.4.58.2 Function ... 708
12.4.58.3 Diagram ... 708
12.4.58.4 Fields ... 708
12.4.59 DDR Memory Test Pattern n Register (DDR_MTP0 - DDR_MTP9) ... 710
12.4.59.1 Offset ... 710
12.4.59.2 Function ... 710
12.4.59.3 Diagram ... 710
12.4.59.4 Fields ... 710
12.4.60 DDR Memory Test Start Extended Address (DDR_MT_ST_EXT_ADDR) ... 711
12.4.60.1 Offset ... 711
12.4.60.2 Function ... 711
12.4.60.3 Diagram ... 711
12.4.60.4 Fields ... 711
12.4.61 DDR Memory Test Start Address (DDR_MT_ST_ADDR) ... 712

12.4.61.1 Offset....712
12.4.61.2 Function.... 712
12.4.61.3 Diagram....712
12.4.61.4 Fields....713
12.4.62 DDR Memory Test End Extended Address (DDR_MT_END_EXT_ADDR).... 713
12.4.62.1 Offset....713
12.4.62.2 Function.... 713
12.4.62.3 Diagram....713
12.4.62.4 Fields....714
12.4.63 DDR Memory Test End Address (DDR_MT_END_ADDR).... 714
12.4.63.1 Offset....714
12.4.63.2 Function.... 714
12.4.63.3 Diagram....714
12.4.63.4 Fields....715
12.4.64 Memory data path error injection mask high (DATA_ERR_INJECT_HI)....715
12.4.64.1 Offset....715
12.4.64.2 Diagram....715
12.4.64.3 Fields....716
12.4.65 Memory data path error injection mask low (DATA_ERR_INJECT_LO)....716
12.4.65.1 Offset....716
12.4.65.2 Diagram....716
12.4.65.3 Fields....717
12.4.66 Memory data path error injection mask ECC (ECC_ERR_INJECT)....717
12.4.66.1 Offset....717
12.4.66.2 Function.... 717
12.4.66.3 Diagram....717
12.4.66.4 Fields....718
12.4.67 Memory data path read capture high (CAPTURE_DATA_HI).... 719
12.4.67.1 Offset....719

12.4.67.2 Function ... 719
12.4.67.3 Diagram ... 719
12.4.67.4 Fields ... 719
12.4.68 Memory data path read capture low (CAPTURE_DATA_LO) ... 719
12.4.68.1 Offset ... 720
12.4.68.2 Function ... 720
12.4.68.3 Diagram ... 720
12.4.68.4 Fields ... 720
12.4.69 Memory data path read capture ECC (CAPTURE_ECC) ... 720
12.4.69.1 Offset ... 720
12.4.69.2 Function ... 721
12.4.69.3 Diagram ... 721
12.4.69.4 Fields ... 721
12.4.70 Memory error detect (ERR_DETECT) ... 722
12.4.70.1 Offset ... 722
12.4.70.2 Function ... 722
12.4.70.3 Diagram ... 722
12.4.70.4 Fields ... 722
12.4.71 Memory error disable (ERR_DISABLE) ... 724
12.4.71.1 Offset ... 724
12.4.71.2 Function ... 724
12.4.71.3 Diagram ... 724
12.4.71.4 Fields ... 725
12.4.72 Memory error interrupt enable (ERR_INT_EN) ... 725
12.4.72.1 Offset ... 726
12.4.72.2 Function ... 726
12.4.72.3 Diagram ... 726
12.4.72.4 Fields ... 726
12.4.73 Memory error attributes capture (CAPTURE_ATTRIBUTES) ... 727

12.4.73.1 Offset....727
12.4.73.2 Function....727
12.4.73.3 Diagram....728
12.4.73.4 Fields....728
12.4.74 Memory error address capture (CAPTURE_ADDRESS)....729
12.4.74.1 Offset....729
12.4.74.2 Function....729
12.4.74.3 Diagram....729
12.4.74.4 Fields....730
12.4.75 Memory error extended address capture (CAPTURE_EXT_ADDRESS)....730
12.4.75.1 Offset....730
12.4.75.2 Function....730
12.4.75.3 Diagram....730
12.4.75.4 Fields....730
12.4.76 Single-Bit ECC memory error management (ERR_SBE)....731
12.4.76.1 Offset....731
12.4.76.2 Function....731
12.4.76.3 Diagram....731
12.4.76.4 Fields....731
12.5 DDR Functional Description....732
12.5.1 DDR SDRAM Interface Operation....736
12.5.1.1 Supported DDR SDRAM Organizations....736
12.5.2 DDR SDRAM Address Multiplexing....737
12.5.3 DDR SDRAM Registered DIMM Mode....740
12.5.4 DDR SDRAM Write Timing Adjustments....741
12.5.5 DDR SDRAM Refresh....741
12.5.5.1 DDR SDRAM Refresh Timing....742
12.5.5.2 DDR SDRAM Refresh and Power-Saving Modes....743
12.5.6 DDR Data Beat Ordering....743

12.5.7 Page Mode and Logical Bank Retention....744
12.5.8 Error Checking and Correcting (ECC)....744
12.5.9 Error Management....746
12.5.10 DDR Rapid Clear of Memory....748
12.6 Initialization/Application Information....748
12.6.1 Programming Summary....752
12.6.2 DDR SDRAM Initialization Sequence....755
12.6.3 Using Forced Self-Refresh Mode to Implement a Battery-Backed RAM System....756
12.6.3.1 Software Based Self-Refresh Scheme....756
12.6.3.2 Bypassing Re-initialization During Battery-Backed Operation....756

Chapter 13
DUART

13.1 The DUART module as implemented on the chip....757
13.2 Overview....757
13.2.1 Features....758
13.2.2 Modes of operation....759
13.3 DUART external signal descriptions....759
13.4 DUART register descriptions....760
13.4.1 DUART memory map....761
13.4.2 UART divisor least significant byte register (UDLB1 - UDLB2)....762
13.4.2.1 Offset....762
13.4.2.2 Function....762
13.4.2.3 Diagram....763
13.4.2.4 Fields....764
13.4.3 UART receiver buffer register (URBR1 - URBR2)....764
13.4.3.1 Offset....764
13.4.3.2 Function....764
13.4.3.3 Diagram....764
13.4.3.4 Fields....765

13.4.4 UART transmitter holding register (UTHR1 - UTHR2)..... 765
13.4.4.1 Offset.....765
13.4.4.2 Function..... 765
13.4.4.3 Diagram.....765
13.4.4.4 Fields.....766
13.4.5 UART divisor most significant byte register (UDMB1 - UDMB2)..... 766
13.4.5.1 Offset.....766
13.4.5.2 Function..... 766
13.4.5.3 Diagram.....766
13.4.5.4 Fields.....767
13.4.6 UART interrupt enable register (UIER1 - UIER2).....767
13.4.6.1 Offset.....767
13.4.6.2 Function..... 767
13.4.6.3 Diagram.....767
13.4.6.4 Fields.....768
13.4.7 UART alternate function register (UAFR1 - UAFR2)..... 768
13.4.7.1 Offset.....768
13.4.7.2 Function..... 769
13.4.7.3 Diagram.....769
13.4.7.4 Fields.....769
13.4.8 UART FIFO control register (UFCR1 - UFCR2).....769
13.4.8.1 Offset.....769
13.4.8.2 Function..... 770
13.4.8.3 Diagram.....770
13.4.8.4 Fields.....770
13.4.9 UART interrupt ID register (UIIR1 - UIIR2)..... 771
13.4.9.1 Offset.....771
13.4.9.2 Function..... 771
13.4.9.3 Diagram.....772

13.4.9.4 Fields....772
13.4.10 UART line control register (ULCR1 - ULCR2)....773
13.4.10.1 Offset....773
13.4.10.2 Function....773
13.4.10.3 Diagram....774
13.4.10.4 Fields....774
13.4.11 UART modem control register (UMCR1 - UMCR2)....775
13.4.11.1 Offset....775
13.4.11.2 Function....775
13.4.11.3 Diagram....776
13.4.11.4 Fields....776
13.4.12 UART line status register (ULSR1 - ULSR2)....776
13.4.12.1 Offset....776
13.4.12.2 Function....777
13.4.12.3 Diagram....777
13.4.12.4 Fields....777
13.4.13 UART modem status register (UMSR1 - UMSR2)....778
13.4.13.1 Offset....778
13.4.13.2 Function....779
13.4.13.3 Diagram....779
13.4.13.4 Fields....779
13.4.14 UART scratch register (USCR1 - USCR2)....779
13.4.14.1 Offset....779
13.4.14.2 Function....780
13.4.14.3 Diagram....780
13.4.14.4 Fields....780
13.4.15 UART DMA status register (UDSR1 - UDSR2)....780
13.4.15.1 Offset....780
13.4.15.2 Function....781

13.4.15.3 Diagram....782
13.4.15.4 Fields....782
13.5 Functional description....782
13.5.1 Serial interface....783
13.5.1.1 START bit....783
13.5.1.2 Data transfer....783
13.5.1.3 Parity bit....784
13.5.1.4 STOP bit....784
13.5.2 Baud-rate generator logic....784
13.5.3 Local loopback mode....785
13.5.4 Errors....785
13.5.4.1 Framing error....785
13.5.4.2 Parity error....786
13.5.4.3 Overrun error....786
13.5.5 FIFO mode....786
13.5.5.1 FIFO interrupts....787
13.5.5.2 Interrupt control logic....787
13.6 Initialization/Application information....787

Chapter 14
Enhanced Secured Digital Host Controller (eSDHC)

14.1 Overview....789
14.1.1 eSDHC features summary....791
14.1.2 Modes and operations....792
14.1.2.1 Data transfer modes....792
14.2 External signals....793
14.2.1 External signals overview....793
14.2.2 eSDHC signal descriptions....794
14.3 eSDHC register descriptions....795
14.3.1 eSDHC memory map....795

14.3.2 SDMA system address register/Block attributes 2 (DSADDR_BLKATTR2)....796
14.3.2.1 Offset....796
14.3.2.2 Function....796
14.3.2.3 Diagram....796
14.3.2.4 Fields....797
14.3.3 Block attributes register (BLKATTR)....797
14.3.3.1 Offset....797
14.3.3.2 Function....798
14.3.3.3 Diagram....798
14.3.3.4 Fields....798
14.3.4 Command argument register (CMDARG)....799
14.3.4.1 Offset....799
14.3.4.2 Function....799
14.3.4.3 Diagram....799
14.3.4.4 Fields....800
14.3.5 Transfer type register (XFERTYP)....800
14.3.5.1 Offset....800
14.3.5.2 Function....800
14.3.5.3 Diagram....802
14.3.5.4 Fields....802
14.3.6 Command response 0 register (CMDRSP0)....804
14.3.6.1 Offset....804
14.3.6.2 Function....805
14.3.6.3 Diagram....805
14.3.6.4 Fields....805
14.3.7 Command response 1 register (CMDRSP1)....805
14.3.7.1 Offset....805
14.3.7.2 Function....806
14.3.7.3 Diagram....806

14.3.7.4 Fields....806
14.3.8 Command response 2 register (CMDRSP2)....806
14.3.8.1 Offset....806
14.3.8.2 Function....807
14.3.8.3 Diagram....807
14.3.8.4 Fields....807
14.3.9 Command Response 3 register (CMDRSP3)....807
14.3.9.1 Offset....807
14.3.9.2 Function....808
14.3.9.3 Diagram....809
14.3.9.4 Fields....809
14.3.10 Buffer data port register (DATPORT)....809
14.3.10.1 Offset....809
14.3.10.2 Function....810
14.3.10.3 Diagram....810
14.3.10.4 Fields....810
14.3.11 Present state register (PRSSTAT)....810
14.3.11.1 Offset....810
14.3.11.2 Function....811
14.3.11.3 Diagram....811
14.3.11.4 Fields....811
14.3.12 Protocol control register (PROCTL)....815
14.3.12.1 Offset....815
14.3.12.2 Function....815
14.3.12.3 Diagram....815
14.3.12.4 Fields....816
14.3.13 System Control Register when ESDHCCTL[CRS=0] (SYSCTL_ESDHCCTL_CRS_0)....818
14.3.13.1 Offset....818
14.3.13.2 Function....819

14.3.13.3 Diagram....819
14.3.13.4 Fields....819
14.3.14 System Control Register when ESDHCCTL[CRS=1] (SYSCTL_ESDHCCTL_CRS_1)....822
14.3.14.1 Offset....822
14.3.14.2 Function....822
14.3.14.3 Diagram....822
14.3.14.4 Fields....822
14.3.15 Interrupt status register (IRQSTAT)....825
14.3.15.1 Offset....825
14.3.15.2 Function....825
14.3.15.3 Diagram....826
14.3.15.4 Fields....826
14.3.16 Interrupt status enable register (IRQSTATEN)....830
14.3.16.1 Offset....830
14.3.16.2 Function....830
14.3.16.3 Diagram....830
14.3.16.4 Fields....831
14.3.17 Interrupt signal enable register (IRQSIGEN)....832
14.3.17.1 Offset....833
14.3.17.2 Function....833
14.3.17.3 Diagram....833
14.3.17.4 Fields....833
14.3.18 Auto CMD Error Status Register / System Control 2 Register (AUTOCERR_SYSCTL2)....835
14.3.18.1 Offset....835
14.3.18.2 Function....835
14.3.18.3 Diagram....836
14.3.18.4 Fields....837
14.3.19 Host controller capabilities register (HOSTCAPBLT)....838
14.3.19.1 Offset....838

14.3.19.2 Function ... 839
14.3.19.3 Diagram ... 839
14.3.19.4 Fields ... 839
14.3.20 Watermark level register (WML) ... 840
14.3.20.1 Offset ... 840
14.3.20.2 Function ... 841
14.3.20.3 Diagram ... 841
14.3.20.4 Fields ... 841
14.3.21 Force event register (FEVT) ... 842
14.3.21.1 Offset ... 842
14.3.21.2 Function ... 842
14.3.21.3 Diagram ... 843
14.3.21.4 Fields ... 843
14.3.22 ADMA error status register (ADMAES) ... 844
14.3.22.1 Offset ... 845
14.3.22.2 Function ... 845
14.3.22.3 Diagram ... 846
14.3.22.4 Fields ... 846
14.3.23 ADMA system address low register (ADSADDRL) ... 846
14.3.23.1 Offset ... 847
14.3.23.2 Function ... 847
14.3.23.3 Diagram ... 847
14.3.23.4 Fields ... 847
14.3.24 ADMA system address high register (ADSADDRH) ... 848
14.3.24.1 Offset ... 848
14.3.24.2 Function ... 848
14.3.24.3 Diagram ... 848
14.3.24.4 Fields ... 848
14.3.25 Host controller version register (HOSTVER) ... 849

14.3.25.1 Offset....849
14.3.25.2 Function....849
14.3.25.3 Diagram....849
14.3.25.4 Fields....849
14.3.26 DMA error address low register (DMAERRADDRL)....850
14.3.26.1 Offset....850
14.3.26.2 Function....850
14.3.26.3 Diagram....850
14.3.26.4 Fields....851
14.3.27 DMA error address high register (DMAERRADDRH)....851
14.3.27.1 Offset....851
14.3.27.2 Function....851
14.3.27.3 Diagram....851
14.3.27.4 Fields....851
14.3.28 DMA error attribute register (DMAERRATTR)....852
14.3.28.1 Offset....852
14.3.28.2 Function....852
14.3.28.3 Diagram....852
14.3.28.4 Fields....852
14.3.29 Host controller capabilities register 2 (HOSTCAPBLT2)....853
14.3.29.1 Offset....853
14.3.29.2 Function....853
14.3.29.3 Diagram....853
14.3.29.4 Fields....854
14.3.30 Tuning block control register (TBCTL)....855
14.3.30.1 Offset....855
14.3.30.2 Function....855
14.3.30.3 Diagram....855
14.3.30.4 Fields....856

14.3.31 Tuning block status register (TBSTAT)..... 857
14.3.31.1 Offset..... 857
14.3.31.2 Function..... 857
14.3.31.3 Diagram..... 857
14.3.31.4 Fields..... 858
14.3.32 Tuning block pointer register (TBPTR)..... 858
14.3.32.1 Offset..... 858
14.3.32.2 Function..... 858
14.3.32.3 Diagram..... 858
14.3.32.4 Fields..... 859
14.3.33 SD direction control register (SDDIRCTL)..... 859
14.3.33.1 Offset..... 859
14.3.33.2 Function..... 860
14.3.33.3 Diagram..... 860
14.3.33.4 Fields..... 860
14.3.34 SD Clock Control Register (SDCLKCTL)..... 860
14.3.34.1 Offset..... 860
14.3.34.2 Function..... 861
14.3.34.3 Diagram..... 861
14.3.34.4 Fields..... 861
14.3.35 eSDHC control register (ESDHCCTL)..... 862
14.3.35.1 Offset..... 862
14.3.35.2 Function..... 862
14.3.35.3 Diagram..... 862
14.3.35.4 Fields..... 863
14.4 Functional description..... 864
14.4.1 System interface and control unit (SysICU)..... 865
14.4.1.1 Data buffer..... 866
14.4.1.1.1 Write operation sequence..... 867

14.4.1.1.2 Read operation sequence....867
14.4.1.1.3 Data buffer and block size.... 868
14.4.1.1.4 Dividing large data transfer.... 871
14.4.1.1.5 Byte order (endianness) of buffer data port register.... 872
14.4.1.2 DMA system interface.... 873
14.4.1.2.1 DMA burst length.... 874
14.4.1.2.2 System master interface.... 874
14.4.1.3 Single DMA (SDMA).... 875
14.4.1.3.1 SDMA error.... 875
14.4.1.4 Advanced DMA (ADMA).... 875
14.4.1.4.1 ADMA concept and descriptor format.... 875
14.4.1.4.2 ADMA interrupt....880
14.4.1.4.3 ADMA error....880
14.4.2 SD interface and control unit (SDICU) 881
14.4.2.1 Command CRC.... 881
14.4.2.2 Data CRC.... 882
14.4.2.3 Tuning block (SDICU)....882
14.4.3 Register bank 884
14.4.4 Clock and reset module....884
14.4.4.1 Clock generator.... 885
14.4.5 SD monitor....886
14.4.5.1 SDIO card interrupt....887
14.4.5.1.1 Interrupts in 1-bit mode.... 887
14.4.5.1.2 Interrupt in 4-bit mode....887
14.4.5.1.3 Card interrupt handling....888
14.4.5.2 Card insertion and removal detection.... 889
14.4.5.3 Power management and wake up events....889
14.4.5.3.1 Setting wake up events....890
14.5 Initialization/application of eSDHC.... 891

14.5.1 Command send and response receive basic operation.....891
14.5.2 Card identification mode.....892
14.5.2.1 Card detect.....892
14.5.2.2 Reset.....893
14.5.2.3 Voltage validation.....894
14.5.2.4 Card registry.....896
14.5.3 Card access.....897
14.5.3.1 clocking constraints.....897
14.5.3.2 Block write.....898
14.5.3.2.1 Normal write.....898
14.5.3.2.2 Write with pause.....900
14.5.3.3 Block read.....901
14.5.3.3.1 Normal read.....901
14.5.3.3.2 Read with pause.....902
14.5.3.4 Suspend resume.....904
14.5.3.4.1 Suspend.....904
14.5.3.4.2 Resume.....904
14.5.3.5 ADMA usage.....905
14.5.3.6 Tuning block procedure.....905
14.5.3.6.1 Tuning procedure for hardware tuning modes.....906
14.5.3.6.2 Tuning procedure for software tuning mode.....907
14.5.3.7 DDR.....907
14.5.3.8 Transfer error.....908
14.5.3.8.1 CRC transfer error.....908
14.5.3.8.2 DMA transfer error.....908
14.5.3.8.3 ADMA transfer error.....908
14.5.3.8.4 Auto CMD12 error.....909
14.5.3.9 Card interrupt.....909
14.5.4 Switch function.....910

14.5.4.1 Query, enable and disable SDIO high speed mode....910
14.5.4.2 Query, enable and disable SD high speed mode....911
14.5.4.3 Query, enable and disable eMMC high speed mode....911
14.5.4.4 Set eMMC bus width....912
14.5.5 ADMA operation....912
14.5.5.1 ADMA1 operation....912
14.5.5.2 ADMA2 operation....913
14.6 Interfacing Card....913
14.7 Commands for /SD/SDIO and eMMC device....914
14.8 Software restrictions....921
14.8.1 Software polling procedure....921
14.8.2 Suspend operation....921
14.8.3 Data port access....921
14.8.4 Multi-block read....922
14.8.5 ADMA address....922
14.8.6 Allowed operations after stop at block gap....922
14.8.7 SDIO card interrupt during soft reset....922
14.8.8 Soft reset for data not allowed when SD clock is disabled....922
14.8.9 Data transfer with Auto CMD12 Enable....923

Chapter 15
FlexTimer Module

15.1 FlexTimer as Implemented on the chip....925
15.2 Introduction....926
15.2.1 FlexTimer philosophy....926
15.2.2 Features....927
15.2.3 Modes of operation....928
15.2.4 Block diagram....928
15.3 FTM signal descriptions....932
15.4 Memory map and register definition....932

15.4.1 Memory map....932
15.4.2 Register descriptions....932
15.4.3 Status And Control (FTM*x*_SC)....938
15.4.4 Counter (FTM*x*_CNT)....940
15.4.5 Modulo (FTM*x*_MOD)....940
15.4.6 Channel (n) Status And Control (FTM*x*_C*n*SC)....941
15.4.7 Channel (n) Value (FTM*x*_C*n*V)....943
15.4.8 Counter Initial Value (FTM*x*_CNTIN)....944
15.4.9 Capture And Compare Status (FTM*x*_STATUS)....945
15.4.10 Features Mode Selection (FTM*x*_MODE)....947
15.4.11 Synchronization (FTM*x*_SYNC)....948
15.4.12 Initial State For Channels Output (FTM*x*_OUTINIT)....951
15.4.13 Output Mask (FTM*x*_OUTMASK)....952
15.4.14 Function For Linked Channels (FTM*x*_COMBINE)....954
15.4.15 Deadtime Insertion Control (FTM*x*_DEADTIME)....958
15.4.16 FTM External Trigger (FTM*x*_EXTTRIG)....959
15.4.17 Channels Polarity (FTM*x*_POL)....961
15.4.18 Fault Mode Status (FTM*x*_FMS)....963
15.4.19 Input Capture Filter Control (FTM*x*_FILTER)....964
15.4.20 Configuration (FTM*x*_CONF)....965
15.4.21 Synchronization Configuration (FTM*x*_SYNCONF)....966
15.4.22 FTM Inverting Control (FTM*x*_INVCTRL)....968
15.4.23 FTM Software Output Control (FTM*x*_SWOCTRL)....969
15.4.24 FTM PWM Load (FTM*x*_PWMLOAD)....971
15.5 Functional description....972
15.5.1 Clock source....973
15.5.1.1 Counter clock source....973
15.5.2 Prescaler....974
15.5.3 Counter....974

15.5.3.1 Up counting....974
15.5.3.2 Up-down counting....977
15.5.3.3 Free running counter....978
15.5.3.4 Counter reset....979
15.5.3.5 When the TOF bit is set....979
15.5.4 Input Capture mode....980
15.5.4.1 Filter for Input Capture mode....981
15.5.5 Output Compare mode....982
15.5.6 Edge-Aligned PWM (EPWM) mode....984
15.5.7 Center-Aligned PWM (CPWM) mode....985
15.5.8 Combine mode....987
15.5.8.1 Asymmetrical PWM....995
15.5.9 Complementary mode....995
15.5.10 Registers updated from write buffers....996
15.5.10.1 CNTIN register update....996
15.5.10.2 MOD register update....997
15.5.10.3 CnV register update....997
15.5.11 PWM synchronization....998
15.5.11.1 Hardware trigger....998
15.5.11.2 Software trigger....999
15.5.11.3 Boundary cycle and loading points....1000
15.5.11.4 MOD register synchronization....1001
15.5.11.5 CNTIN register synchronization....1004
15.5.11.6 C(n)V and C(n+1)V register synchronization....1005
15.5.11.7 OUTMASK register synchronization....1005
15.5.11.8 INVCTRL register synchronization....1008
15.5.11.9 SWOCTRL register synchronization....1009
15.5.11.10 FTM counter synchronization....1011
15.5.12 Inverting....1014

15.5.13 Software output control....1015
15.5.14 Deadtime insertion....1017
15.5.14.1 Deadtime insertion corner cases....1018
15.5.15 Output mask....1020
15.5.16 Polarity control....1020
15.5.17 Initialization....1021
15.5.18 Features priority....1021
15.5.19 Channel trigger output....1022
15.5.20 Initialization trigger....1024
15.5.21 Capture Test mode....1025
15.5.22 Dual Edge Capture mode....1026
15.5.22.1 One-Shot Capture mode....1027
15.5.22.2 Continuous Capture mode....1028
15.5.22.3 Pulse width measurement....1028
15.5.22.4 Period measurement....1030
15.5.22.5 Read coherency mechanism....1032
15.5.23 Debug mode....1033
15.5.24 Intermediate load....1034
15.5.25 Global time base (GTB)....1036
15.5.25.1 Enabling the global time base (GTB)....1037
15.6 Reset overview....1038
15.7 FTM Interrupts....1039
15.7.1 Timer Overflow Interrupt....1040
15.7.2 Channel (n) Interrupt....1040
15.8 Initialization Procedure....1040

Chapter 16
General Purpose I/O (GPIO)

16.1 The GPIO module as implemented on the chip....1043
16.2 GPIO overview....1043

Section number	Title	Page

16.3 GPIO features summary....1044

16.4 GPIO signal descriptions....1045

16.5 GPIO register descriptions....1045

16.5.1 GPIO memory map....1045

16.5.2 GPIO direction register (GPDIR)....1046

16.5.2.1 Offset....1046

16.5.2.2 Function....1046

16.5.2.3 Diagram....1046

16.5.2.4 Fields....1046

16.5.3 GPIO open drain register (GPODR)....1047

16.5.3.1 Offset....1047

16.5.3.2 Function....1047

16.5.3.3 Diagram....1047

16.5.3.4 Fields....1047

16.5.4 GPIO data register (GPDAT)....1048

16.5.4.1 Offset....1048

16.5.4.2 Function....1048

16.5.4.3 Diagram....1048

16.5.4.4 Fields....1048

16.5.5 GPIO interrupt event register (GPIER)....1049

16.5.5.1 Offset....1049

16.5.5.2 Function....1049

16.5.5.3 Diagram....1049

16.5.5.4 Fields....1050

16.5.6 GPIO interrupt mask register (GPIMR)....1050

16.5.6.1 Offset....1050

16.5.6.2 Function....1050

16.5.6.3 Diagram....1051

16.5.6.4 Fields....1051

16.5.7 GPIO interrupt control register (GPICR)....1051
16.5.7.1 Offset....1051
16.5.7.2 Function....1051
16.5.7.3 Diagram....1051
16.5.7.4 Fields....1052
16.5.8 GPIO Input Buffer Enable Register (GPIBE)....1052
16.5.8.1 Offset....1052
16.5.8.2 Function....1052
16.5.8.3 Diagram....1053
16.5.8.4 Fields....1053

Chapter 17
Integrated Flash Controller (IFC)

17.1 IFC overview....1055
17.1.1 IFC features summary....1056
17.1.1.1 NAND flash controller features....1057
17.1.1.2 NOR flash features....1058
17.1.1.3 GPCM and GASIC features....1059
17.1.2 IFC modes of operation....1059
17.2 External signal descriptions....1059
17.2.1 Internal connectivity of WP/RB signal....1064
17.2.1.1 Internal connectivity of WP signal....1064
17.2.1.2 Internal connectivity of RB signal....1065
17.3 IFC register descriptions....1066
17.3.1 IFC memory map....1066
17.3.2 IFC Revision Control register (REV)....1075
17.3.2.1 Offset....1075
17.3.2.2 Function....1075
17.3.2.3 Diagram....1076
17.3.2.4 Fields....1076

Section number	Title	Page

17.3.3 Extended Chip Select Property Registers (CSPR0_EXT - CSPR7_EXT) 1076

17.3.3.1 Offset 1076

17.3.3.2 Function 1077

17.3.3.3 Diagram 1077

17.3.3.4 Fields 1077

17.3.4 Chip-select Property register n (CSPR0 - CSPR7) 1077

17.3.4.1 Offset 1078

17.3.4.2 Function 1078

17.3.4.3 Diagram 1078

17.3.4.4 Fields 1078

17.3.5 Address Mask register (AMASK0 - AMASK7) 1079

17.3.5.1 Offset 1079

17.3.5.2 Function 1080

17.3.5.3 Diagram 1082

17.3.5.4 Fields 1082

17.3.6 Chip-Select Option register - GPCM (CSOR0_GPCM - CSOR7_GPCM) 1082

17.3.6.1 Offset 1082

17.3.6.2 Function 1082

17.3.6.3 Diagram 1083

17.3.6.4 Fields 1083

17.3.7 Chip-Select Option register - NAND Flash Mode (CSOR0_NAND - CSOR7_NAND) 1086

17.3.7.1 Offset 1086

17.3.7.2 Function 1086

17.3.7.3 Diagram 1086

17.3.7.4 Fields 1087

17.3.8 Chip-Select Option register - NOR Flash Mode (CSOR0_NOR - CSOR7_NOR) 1089

17.3.8.1 Offset 1089

17.3.8.2 Function 1089

17.3.8.3 Diagram 1089

17.3.8.4 Fields.....1090

17.3.9 Extended Chip-Select Option register - NAND Flash Mode (CSOR0_EXT - CSOR7_EXT).....1092

17.3.9.1 Offset.....1093

17.3.9.2 Function.....1093

17.3.9.3 Diagram.....1093

17.3.9.4 Fields.....1093

17.3.10 Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS0_GPCM - FTIM0_CS7_GPCM)..1094

17.3.10.1 Offset.....1094

17.3.10.2 Function.....1094

17.3.10.3 Diagram.....1095

17.3.10.4 Fields.....1095

17.3.11 Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS0_NAND - FTIM0_CS7_NAND).....1096

17.3.11.1 Offset.....1096

17.3.11.2 Function.....1096

17.3.11.3 Diagram.....1097

17.3.11.4 Fields.....1097

17.3.12 Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS0_NAND_ASYNC_MODE - FTIM0_CS7_NAND_ASYNC_MODE).....1098

17.3.12.1 Offset.....1098

17.3.12.2 Function.....1098

17.3.12.3 Diagram.....1098

17.3.12.4 Fields.....1099

17.3.13 Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS0_NOR - FTIM0_CS7_NOR).....1100

17.3.13.1 Offset.....1100

17.3.13.2 Function.....1100

17.3.13.3 Diagram.....1100

17.3.13.4 Fields.....1101

17.3.14 Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS0_GPCM - FTIM1_CS7_GPCM)..1101

17.3.14.1 Offset....1101
17.3.14.2 Function....1102
17.3.14.3 Diagram....1102
17.3.14.4 Fields....1102
17.3.15 Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS0_NAND - FTIM1_CS7_NAND)....1103
17.3.15.1 Offset....1103
17.3.15.2 Function....1103
17.3.15.3 Diagram....1103
17.3.15.4 Fields....1104
17.3.16 Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS0_NAND_ASYNC_MODE - FTIM1_CS7_NAND_ASYNC_MODE)....1104
17.3.16.1 Offset....1104
17.3.16.2 Function....1104
17.3.16.3 Diagram....1105
17.3.16.4 Fields....1105
17.3.17 Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS0_NOR - FTIM1_CS7_NOR)....1105
17.3.17.1 Offset....1106
17.3.17.2 Function....1106
17.3.17.3 Diagram....1106
17.3.17.4 Fields....1106
17.3.18 Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS0_GPCM - FTIM2_CS7_GPCM)....1107
17.3.18.1 Offset....1107
17.3.18.2 Function....1107
17.3.18.3 Diagram....1107
17.3.18.4 Fields....1108
17.3.19 Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS0_NAND - FTIM2_CS7_NAND)....1108
17.3.19.1 Offset....1109
17.3.19.2 Function....1109

17.3.19.3 Diagram....1109
17.3.19.4 Fields....1109
17.3.20 Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS0_NAND_ASYNC_MODE - FTIM2_CS7_NAND_ASYNC_MODE)....1110
17.3.20.1 Offset....1110
17.3.20.2 Function....1110
17.3.20.3 Diagram....1110
17.3.20.4 Fields....1111
17.3.21 Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS0_NOR - FTIM2_CS7_NOR)....1111
17.3.21.1 Offset....1111
17.3.21.2 Function....1111
17.3.21.3 Diagram....1112
17.3.21.4 Fields....1112
17.3.22 Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS0_GPCM - FTIM3_CS7_GPCM)....1113
17.3.22.1 Offset....1113
17.3.22.2 Function....1113
17.3.22.3 Diagram....1113
17.3.22.4 Fields....1114
17.3.23 Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS0_NAND - FTIM3_CS7_NAND)....1114
17.3.23.1 Offset....1114
17.3.23.2 Function....1115
17.3.23.3 Diagram....1115
17.3.23.4 Fields....1115
17.3.24 Flash Timing register 3 for CSn - NOR Flash Mode (FTIM3_CS_NOR)....1115
17.3.24.1 Offset....1115
17.3.24.2 Function....1116
17.3.24.3 Diagram....1116
17.3.24.4 Fields....1116

17.3.25 Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM03_CS_NAND_ASYNC_MODE - IFC_FTIM73_CS_NAND_ASYNC_MODE).....1117

17.3.25.1 Offset.....1117

17.3.25.2 Function.....1117

17.3.25.3 Diagram.....1117

17.3.25.4 Fields.....1118

17.3.26 Ready Busy Status for each Chip Select (RB_STAT).....1118

17.3.26.1 Offset.....1118

17.3.26.2 Diagram.....1118

17.3.26.3 Fields.....1119

17.3.27 Ready Busy Map register (RB_MAP).....1120

17.3.27.1 Offset.....1120

17.3.27.2 Function.....1120

17.3.27.3 Diagram.....1120

17.3.27.4 Fields.....1121

17.3.28 Write Protect Map register (WP_MAP).....1123

17.3.28.1 Offset.....1123

17.3.28.2 Function.....1123

17.3.28.3 Diagram.....1123

17.3.28.4 Fields.....1123

17.3.29 General Control register (GCR).....1125

17.3.29.1 Offset.....1125

17.3.29.2 Function.....1125

17.3.29.3 Diagram.....1125

17.3.29.4 Fields.....1126

17.3.30 Common Event and Error Status register (CM_EVTER_STAT).....1126

17.3.30.1 Offset.....1127

17.3.30.2 Function.....1127

17.3.30.3 Diagram.....1127

17.3.30.4 Fields....1127
17.3.31 Common Event and Error Enable register (CM_EVTER_EN)....1128
17.3.31.1 Offset....1128
17.3.31.2 Function....1128
17.3.31.3 Diagram....1128
17.3.31.4 Fields....1128
17.3.32 Common Event and Error Interrupt Enable register (CM_EVTER_INTR_EN)....1129
17.3.32.1 Offset....1129
17.3.32.2 Function....1129
17.3.32.3 Diagram....1129
17.3.32.4 Fields....1129
17.3.33 Common Transfer Error Attributes register 0 (CM_ERATTR0)....1130
17.3.33.1 Offset....1130
17.3.33.2 Function....1130
17.3.33.3 Diagram....1130
17.3.33.4 Fields....1130
17.3.34 Common Transfer Error Attributes register 1 (CM_ERATTR1)....1131
17.3.34.1 Offset....1131
17.3.34.2 Function....1131
17.3.34.3 Diagram....1131
17.3.34.4 Fields....1132
17.3.35 Clock Control register (CCR)....1132
17.3.35.1 Offset....1132
17.3.35.2 Diagram....1132
17.3.35.3 Fields....1133
17.3.36 Clock Status register (CSR)....1134
17.3.36.1 Offset....1134
17.3.36.2 Diagram....1134
17.3.36.3 Fields....1134

17.3.37 DDR Clock Control register Low (DDR_CCR_LOW) 1135
17.3.37.1 Offset 1135
17.3.37.2 Function 1135
17.3.37.3 Diagram 1135
17.3.37.4 Fields 1135
17.3.38 NAND Configuration register (NCFGR) 1137
17.3.38.1 Offset 1137
17.3.38.2 Function 1137
17.3.38.3 Diagram 1137
17.3.38.4 Fields 1137
17.3.39 NAND Flash Command register 0 (NAND_FCR0) 1139
17.3.39.1 Offset 1139
17.3.39.2 Function 1139
17.3.39.3 Diagram 1139
17.3.39.4 Fields 1140
17.3.40 NAND Flash Command register 1 (NAND_FCR1) 1140
17.3.40.1 Offset 1140
17.3.40.2 Diagram 1140
17.3.40.3 Fields 1141
17.3.41 Flash Row Address register n (ROW0 - ROW3) 1141
17.3.41.1 Offset 1141
17.3.41.2 Function 1141
17.3.41.3 Diagram 1142
17.3.41.4 Fields 1142
17.3.42 Flash COL Address register n (COL0 - COL3) 1142
17.3.42.1 Offset 1142
17.3.42.2 Function 1143
17.3.42.3 Diagram 1143
17.3.42.4 Fields 1144

17.3.43 Flash COL Address register for 2 KB Large-Page Device (COL0_2KB - COL3_2KB).....1144
17.3.43.1 Offset.....1144
17.3.43.2 Function.....1145
17.3.43.3 Diagram.....1145
17.3.43.4 Fields.....1145
17.3.44 Flash COL Address register for 4 KB Large-Page Device (COL0_4KB - COL3_4KB).....1146
17.3.44.1 Offset.....1146
17.3.44.2 Function.....1146
17.3.44.3 Diagram.....1147
17.3.44.4 Fields.....1147
17.3.45 Flash Byte Count register for NAND Flash (NAND_BC).....1147
17.3.45.1 Offset.....1147
17.3.45.2 Function.....1148
17.3.45.3 Diagram.....1148
17.3.45.4 Fields.....1148
17.3.46 NAND Flash Instruction register 0 (NAND_FIR0).....1149
17.3.46.1 Offset.....1149
17.3.46.2 Function.....1149
17.3.46.3 Diagram.....1151
17.3.46.4 Fields.....1151
17.3.47 NAND Flash Instruction register 1 (NAND_FIR1).....1152
17.3.47.1 Offset.....1152
17.3.47.2 Function.....1152
17.3.47.3 Diagram.....1152
17.3.47.4 Fields.....1152
17.3.48 NAND Flash Instruction register 2 (NAND_FIR2).....1153
17.3.48.1 Offset.....1153
17.3.48.2 Function.....1153
17.3.48.3 Diagram.....1153

17.3.48.4 Fields....1153
17.3.49 NAND Chip-Select register (NAND_CSEL)....1154
17.3.49.1 Offset....1154
17.3.49.2 Function....1154
17.3.49.3 Diagram....1154
17.3.49.4 Fields....1155
17.3.50 NAND Operation Sequence Start (NANDSEQ_STRT)....1155
17.3.50.1 Offset....1155
17.3.50.2 Function....1155
17.3.50.3 Diagram....1156
17.3.50.4 Fields....1156
17.3.51 NAND Event and Error Status register (NAND_EVTER_STAT)....1157
17.3.51.1 Offset....1157
17.3.51.2 Function....1157
17.3.51.3 Diagram....1158
17.3.51.4 Fields....1158
17.3.52 NAND Page Read Completion Event Status register (PGRDCMPL_EVT_STAT)....1159
17.3.52.1 Offset....1159
17.3.52.2 Function....1160
17.3.52.3 Diagram....1160
17.3.52.4 Fields....1161
17.3.53 NAND Event and Error Enable register (NAND_EVTER_EN)....1161
17.3.53.1 Offset....1161
17.3.53.2 Function....1162
17.3.53.3 Diagram....1162
17.3.53.4 Fields....1162
17.3.54 NAND Event and Error Interrupt Enable register (NAND_EVTER_INTR_EN)....1163
17.3.54.1 Offset....1163
17.3.54.2 Function....1163

17.3.54.3 Diagram....1163
17.3.54.4 Fields....1164
17.3.55 NAND Transfer Error Attributes register 0 (NAND_ERATTR0)....1165
17.3.55.1 Offset....1165
17.3.55.2 Function....1165
17.3.55.3 Diagram....1165
17.3.55.4 Fields....1165
17.3.56 NAND Transfer Error Attributes register 1 (NAND_ERATTR1)....1166
17.3.56.1 Offset....1166
17.3.56.2 Function....1166
17.3.56.3 Diagram....1166
17.3.56.4 Fields....1167
17.3.57 NAND Flash Status register (NAND_FSR)....1167
17.3.57.1 Offset....1167
17.3.57.2 Function....1167
17.3.57.3 Diagram....1167
17.3.57.4 Fields....1168
17.3.58 ECC Status and Result of Flash Operation register 0 (ECCSTAT0)....1168
17.3.58.1 Offset....1168
17.3.58.2 Function....1168
17.3.58.3 Diagram....1169
17.3.58.4 Fields....1170
17.3.59 ECC Status and Result of Flash Operation register 1 (ECCSTAT1)....1171
17.3.59.1 Offset....1171
17.3.59.2 Function....1171
17.3.59.3 Diagram....1171
17.3.59.4 Fields....1172
17.3.60 ECC Status and Result of Flash Operation Register2 (ECCSTAT2)....1172
17.3.60.1 Offset....1173

17.3.60.2 Function ... 1173
17.3.60.3 Diagram ... 1173
17.3.60.4 Fields ... 1173
17.3.61 ECC Status and Result of Flash Operation Register3 (ECCSTAT3) ... 1174
17.3.61.1 Offset ... 1174
17.3.61.2 Function ... 1174
17.3.61.3 Diagram ... 1174
17.3.61.4 Fields ... 1175
17.3.62 ECC Status and Result of Flash Operation Register4 (ECCSTAT4) ... 1175
17.3.62.1 Offset ... 1176
17.3.62.2 Function ... 1176
17.3.62.3 Diagram ... 1176
17.3.62.4 Fields ... 1176
17.3.63 ECC Status and Result of Flash Operation Register5 (ECCSTAT5) ... 1177
17.3.63.1 Offset ... 1177
17.3.63.2 Function ... 1177
17.3.63.3 Diagram ... 1177
17.3.63.4 Fields ... 1178
17.3.64 ECC Status and Result of Flash Operation Register6 (ECCSTAT6) ... 1178
17.3.64.1 Offset ... 1178
17.3.64.2 Function ... 1178
17.3.64.3 Diagram ... 1178
17.3.64.4 Fields ... 1179
17.3.65 ECC Status and Result of Flash Operation Register7 (ECCSTAT7) ... 1179
17.3.65.1 Offset ... 1179
17.3.65.2 Function ... 1180
17.3.65.3 Diagram ... 1180
17.3.65.4 Fields ... 1180
17.3.66 NAND Control register (NANDCR) ... 1181

17.3.66.1 Offset....1181
17.3.66.2 Diagram....1181
17.3.66.3 Fields....1181
17.3.67 NAND Autoboot Trigger register (NAND_AUTOBOOT_TRGR)....1182
17.3.67.1 Offset....1182
17.3.67.2 Function....1182
17.3.67.3 Diagram....1182
17.3.67.4 Fields....1183
17.3.68 NAND Flash Memory Data register (NAND_MDR)....1183
17.3.68.1 Offset....1183
17.3.68.2 Function....1184
17.3.68.3 Diagram....1184
17.3.68.4 Fields....1184
17.3.69 Nand DLL Low Config 0 Register (NAND_DLL_LOW_CFG0)....1185
17.3.69.1 Offset....1185
17.3.69.2 Function....1185
17.3.69.3 Diagram....1185
17.3.69.4 Fields....1185
17.3.70 Nand DLL Low Config 1 Register (NAND_DLL_LOW_CFG1)....1186
17.3.70.1 Offset....1186
17.3.70.2 Function....1186
17.3.70.3 Diagram....1186
17.3.70.4 Fields....1187
17.3.71 NAND DLL Low Status Register (NAND_DLL_LOW_STAT)....1188
17.3.71.1 Offset....1188
17.3.71.2 Function....1188
17.3.71.3 Diagram....1188
17.3.71.4 Fields....1189
17.3.72 NOR Event and Error Status register (NOR_EVTER_STAT)....1190

17.3.72.1 Offset....1190
17.3.72.2 Function.... 1190
17.3.72.3 Diagram....1190
17.3.72.4 Fields....1191
17.3.73 NOR Event and Error Enable register (NOR_EVTER_EN).... 1191
17.3.73.1 Offset....1191
17.3.73.2 Function.... 1191
17.3.73.3 Diagram....1191
17.3.73.4 Fields....1192
17.3.74 NOR Event and Error Interrupt enable register (NOR_EVTER_INTR_EN)....1192
17.3.74.1 Offset....1192
17.3.74.2 Function.... 1193
17.3.74.3 Diagram....1193
17.3.74.4 Fields....1193
17.3.75 NOR Transfer Error Attributes register 0 (NOR_ERATTR0)....1194
17.3.75.1 Offset....1194
17.3.75.2 Function.... 1194
17.3.75.3 Diagram....1194
17.3.75.4 Fields....1195
17.3.76 NOR Transfer Error Attribute register 1 (NOR_ERATTR1).... 1195
17.3.76.1 Offset....1195
17.3.76.2 Function.... 1196
17.3.76.3 Diagram....1196
17.3.76.4 Fields....1196
17.3.77 NOR Transfer Error Attribute register 2 (NOR_ERATTR2).... 1196
17.3.77.1 Offset....1196
17.3.77.2 Function.... 1197
17.3.77.3 Diagram....1197
17.3.77.4 Fields....1197

17.3.78 NOR Control register (NORCR)....1198
17.3.78.1 Offset....1198
17.3.78.2 Diagram....1198
17.3.78.3 Fields....1198
17.3.79 GPCM Event and Error Status register (GPCM_EVTER_STAT)....1200
17.3.79.1 Offset....1200
17.3.79.2 Function....1200
17.3.79.3 Diagram....1200
17.3.79.4 Fields....1200
17.3.80 GPCM Event and Error Enable register (GPCM_EVTER_EN)....1201
17.3.80.1 Offset....1201
17.3.80.2 Function....1201
17.3.80.3 Diagram....1201
17.3.80.4 Fields....1202
17.3.81 GPCM Event and Error Interrupt enable register (GPCM_EVTER_INTR_EN)....1202
17.3.81.1 Offset....1202
17.3.81.2 Function....1203
17.3.81.3 Diagram....1203
17.3.81.4 Fields....1203
17.3.82 GPCM Transfer Error Attributes register 0 (GPCM_ERATTR0)....1204
17.3.82.1 Offset....1204
17.3.82.2 Function....1204
17.3.82.3 Diagram....1204
17.3.82.4 Fields....1205
17.3.83 GPCM Transfer Error Attributes register 1 (GPCM_ERATTR1)....1205
17.3.83.1 Offset....1205
17.3.83.2 Function....1206
17.3.83.3 Diagram....1206
17.3.83.4 Fields....1206

17.3.84 GPCM Transfer Error Attributes register 2 (GPCM_ERATTR2)....1206
17.3.84.1 Offset....1206
17.3.84.2 Function....1207
17.3.84.3 Diagram....1207
17.3.84.4 Fields....1207
17.3.85 GPCM Status register (GPCM_STAT)....1208
17.3.85.1 Offset....1208
17.3.85.2 Function....1208
17.3.85.3 Diagram....1208
17.3.85.4 Fields....1209
17.4 IFC functional description....1209
17.4.1 General architecture....1211
17.4.1.1 Bank selection through address decoding....1211
17.4.1.2 Address Data Pin Multiplexing....1211
17.4.1.3 Mode-0 Pin Muxing (CSOR[ADM_SHFT_MODE] = 0)....1212
17.4.1.4 Mode-1 Pin Muxing (CSOR[ADM_SHFT_MODE] = 1)....1213
17.4.2 Programming model for flash interface timing....1215
17.4.3 Booting methods....1216
17.4.4 Software reset handling....1216
17.4.4.1 System bus handling....1217
17.4.4.2 Flash interface handling....1217
17.4.5 Data buffer control (BCTL)....1217
17.4.6 External transceiver enable (TE)....1219
17.4.6.1 Transceiver enable during boot....1219
17.4.6.2 Transceiver enable post-boot....1220
17.4.6.3 Transceiver enable example....1221
17.5 NAND flash control machine....1221
17.5.1 NAND Flash Connections....1221
17.5.2 NAND flash synchronous mode....1222

17.5.3 NAND flash asynchronous mode.... 1222
17.5.4 NV-DDR Mode.... 1223
17.5.5 Write protect.... 1223
17.5.6 SRAM buffer.... 1224
17.5.6.1 Guidelines for SRAM buffer for NAND access.... 1225
17.5.6.2 Buffer layout and page mapping for 512-byte page NAND flash.... 1226
17.5.6.3 Buffer layout and page mapping for 2-KB page NAND flash.... 1227
17.5.6.4 Buffer layout and page mapping for 4 KB page NAND flash.... 1228
17.5.6.5 Buffer layout and page mapping for 16 KB (large) page NAND flash.... 1229
17.5.6.6 SRAM buffer initialization requirement.... 1230
17.5.7 Use of ECC algorithms.... 1230
17.5.7.1 Generating Galois field elements.... 1231
17.5.7.2 BCH encoding.... 1231
17.5.8 Programming the NAND FCM.... 1236
17.5.8.1 FCM command instructions.... 1237
17.5.8.2 FCM address instructions.... 1238
17.5.8.3 FCM data read instructions.... 1238
17.5.8.4 FCM data write instructions.... 1239
17.5.9 Looping FIR sequences.... 1240
17.5.10 NAND DLL.... 1240
17.5.11 NAND asynchronous mode timings.... 1241
17.5.11.1 NAND asynchronous mode program data timing.... 1241
17.5.11.2 NAND asynchronous mode read data timing.... 1242
17.5.11.3 NAND asynchronous mode calculating read data window width.... 1243
17.5.11.4 NAND asynchronous mode read status timing.... 1244
17.5.12 NV-DDR mode timings.... 1245
17.5.12.1 NAND synchronous mode program operation.... 1245
17.5.12.2 NAND synchronous mode program with clock stopped.... 1245
17.5.12.3 NAND synchronous page read operation.... 1246

Section number	Title	Page

17.6 NOR flash control machine 1247

17.6.1 NOR boot 1247

17.6.2 Unmuxed (parallel) asynchronous NOR read timings 1247

17.6.3 Simple asynchronous NOR write timings 1248

17.6.4 Muxed (ADM) asynchronous NOR read timings 1249

17.6.5 Muxed (ADM) asynchronous NOR write timings 1249

17.7 Unmuxed (Parallel) Synchronous NOR Read Timings 1250

17.7.1 Continuous Mode Sync Burst Read- Unmuxed NOR 1250

17.7.2 Unmuxed (parallel) asynchronous NOR read timings 1251

17.8 Muxed (ADM) Synchronous NOR Read Timings 1252

17.8.1 Continuous Mode Sync Burst Read- ADM NOR 1252

17.8.2 Split Mode Sync Burst Mode- ADM NOR 1254

17.9 General purpose chip-select machine (GPCM) 1257

17.9.1 Normal GPCM mode of operation 1257

17.9.1.1 Normal GPCM program operation 1258

17.9.1.1.1 Normal GPCM internal counter-based program operation 1258

17.9.1.1.2 Normal GPCM external termination-based program operation 1261

17.9.1.2 Normal GPCM read operation 1262

17.9.1.2.1 Normal GPCM internal counter-based read operation 1263

17.9.1.2.2 Normal GPCM external termination-based read operation 1267

17.9.2 Generic ASIC mode of operation 1268

17.9.2.1 General ASIC program operation 1269

17.9.2.2 Generic ASIC read data sampling 1271

17.10 Clock generation module 1275

17.11 Initialization/Application information 1275

17.11.1 Unmuxed (Parallel) NOR with no loop back 1275

17.11.2 ADM NOR connection 1276

17.11.3 Switching Interfaces in ONFi NAND 1277

17.11.3.1 Activating the NVDDR Interface 1277

17.11.3.2 Switching to the asynchronous interface.... 1278
17.11.3.3 Switching timing modes when configured in NVDDR mode.... 1278
17.11.4 ONFI mode timing parameters.... 1279
17.11.5 NAND Volume Addressing Initialization sequence.... 1280
17.11.6 DLL configuration guideline (valid when IFC is in NVDDR Mode).... 1281
17.11.7 IFC flash connections.... 1281
17.11.8 Bus turnaround.... 1281
17.11.8.1 Address phase after previous read.... 1282
17.11.8.2 Read-data phase after address phase.... 1282
17.11.8.3 Read-modify-write cycle for parity protected memory banks.... 1282
17.11.9 Interfacing to different port sizes.... 1283
17.11.10 Command sequence examples for NAND flash EEPROM.... 1283
17.11.10.1 NAND flash soft reset command sequence example.... 1284
17.11.10.2 NAND flash read status command sequence example.... 1284
17.11.10.3 NAND flash read identification command sequence example.... 1285
17.11.10.4 NAND flash page read command sequence example.... 1286
17.11.10.5 NAND flash program command sequence example.... 1286
17.11.10.6 Read status command during busy period of program/erase operation.... 1287
17.11.10.7 Valid opcode transitions in the IFC.... 1288

Chapter 18
Inter-Integrated Circuit (I2C)

18.1 The I2C module as implemented on the chip.... 1291
18.2 Overview.... 1291
18.3 Introduction to I2C.... 1291
18.3.1 Definition: I2C module.... 1292
18.3.2 Module block diagram.... 1292
18.3.3 Features.... 1292
18.3.4 Modes of operation.... 1293
18.3.5 Definition: I2C conditions.... 1294

Section number	Title	Page

18.4 External signal descriptions.... 1295

18.4.1 Signal overview.... 1295

18.4.2 Detailed external signal descriptions.... 1295

18.5 I2C register descriptions.... 1296

18.5.1 I2C memory map.... 1296

18.5.2 I2C Bus Address Register (IBAD).... 1296

18.5.2.1 Offset.... 1296

18.5.2.2 Function.... 1297

18.5.2.3 Diagram.... 1297

18.5.2.4 Fields.... 1297

18.5.3 I2C Bus Frequency Divider Register (IBFD).... 1297

18.5.3.1 Offset.... 1297

18.5.3.2 Function.... 1298

18.5.3.3 Diagram.... 1298

18.5.3.4 Fields.... 1298

18.5.4 I2C Bus Control Register (IBCR).... 1298

18.5.4.1 Offset.... 1298

18.5.4.2 Function.... 1298

18.5.4.3 Diagram.... 1299

18.5.4.4 Fields.... 1299

18.5.5 I2C Bus Status Register (IBSR).... 1300

18.5.5.1 Offset.... 1300

18.5.5.2 Function.... 1300

18.5.5.3 Diagram.... 1300

18.5.5.4 Fields.... 1301

18.5.6 I2C Bus Data I/O Register (IBDR).... 1302

18.5.6.1 Offset.... 1302

18.5.6.2 Function.... 1302

18.5.6.3 Diagram.... 1303

18.5.6.4 Fields....1303
18.5.7 I2C Bus Interrupt Config Register (IBIC)....1304
18.5.7.1 Offset....1304
18.5.7.2 Function....1304
18.5.7.3 Diagram....1304
18.5.7.4 Fields....1304
18.5.8 I2C Bus Debug Register (IBDBG)....1305
18.5.8.1 Offset....1305
18.5.8.2 Function....1305
18.5.8.3 Diagram....1305
18.5.8.4 Fields....1305
18.6 Functional description....1306
18.6.1 Notes about module operation....1306
18.6.2 Transactions....1307
18.6.2.1 Protocol overview....1307
18.6.2.2 Transaction protocol definitions....1308
18.6.2.3 I2C calling address requirements....1308
18.6.2.4 High-level protocol steps....1309
18.6.2.5 START condition....1309
18.6.2.6 Slave address transmission....1309
18.6.2.7 Data transmission....1310
18.6.2.8 STOP condition....1310
18.6.2.9 Repeated START condition....1311
18.6.3 Arbitration procedure....1311
18.6.4 Clock behavior....1311
18.6.4.1 Clock synchronization....1312
18.6.4.2 Clock stretching....1312
18.6.4.3 Handshaking....1312
18.6.4.4 Clock rate and IBFD settings....1313

18.6.4.4.1 Timing definitions....1313
18.6.4.4.2 Divider and hold values....1313
18.6.5 Interrupts....1324
18.6.5.1 Interrupt vector....1325
18.6.5.2 Interrupt description....1325
18.6.6 STOP mode....1325
18.6.7 DEBUG mode....1326
18.7 Initialization/application information....1328
18.7.1 Recommended interrupt service flow....1328
18.7.2 General programming guidelines (for both master and slave mode)....1329
18.7.2.1 Initializing the I2C module....1329
18.7.2.2 Software response after a transfer....1330
18.7.3 Programming guidelines specific to Master mode....1331
18.7.3.1 Generating START....1331
18.7.3.2 Transmit/receive sequence....1332
18.7.3.3 Generating STOP....1334
18.7.3.4 Generating repeated START....1334
18.7.3.5 Loss of arbitration....1334
18.7.4 Programming guidelines specific to Slave mode....1335

Chapter 19
PCI Express Interface Controller

19.1 The PCI Express controller as implemented on the chip....1337
19.2 Introduction....1338
19.2.1 Overview....1338
19.2.1.1 Outbound Transactions....1340
19.2.1.2 Inbound Transactions....1340
19.2.2 Features....1341
19.2.3 Modes of Operation....1342
19.2.3.1 Root Complex/Endpoint Modes....1342

19.2.3.2 Link Width.... 1342
19.2.3.3 Link Speed.... 1342
19.2.3.4 EP SR-IOV.... 1342
19.2.3.5 MSI-X (SR-IOV-only).... 1343
19.3 External Signal Descriptions.... 1343
19.4 Memory map/register overview.... 1344
19.4.1 PCI Express configuration registers.... 1344
19.4.2 SR-IOV PF/VF configuration registers.... 1344
19.4.3 SR-IOV Instance (PEX1) register descriptions.... 1345
19.4.3.1 PCI_Express_Configuration_Registers memory map.... 1345
19.4.3.2 PCI Express Vendor ID Register (Vendor_ID_Register).... 1350
19.4.3.2.1 Offset.... 1351
19.4.3.2.2 Function.... 1351
19.4.3.2.3 Diagram.... 1351
19.4.3.2.4 Fields.... 1351
19.4.3.3 PCI Express Device ID Register (Device_ID_Register).... 1351
19.4.3.3.1 Offset.... 1351
19.4.3.3.2 Function.... 1351
19.4.3.3.3 Diagram.... 1352
19.4.3.3.4 Fields.... 1352
19.4.3.4 PCI Express Command Register (Command_Register).... 1352
19.4.3.4.1 Offset.... 1352
19.4.3.4.2 Function.... 1352
19.4.3.4.3 Diagram.... 1353
19.4.3.4.4 Fields.... 1353
19.4.3.5 PCI Express Status Register (Status_Register).... 1354
19.4.3.5.1 Offset.... 1354
19.4.3.5.2 Function.... 1354
19.4.3.5.3 Diagram.... 1354

19.4.3.5.4 Fields....1355
19.4.3.6 PCI Express Revision ID Register (Revision_ID_Register)....1356
19.4.3.6.1 Offset....1356
19.4.3.6.2 Function....1356
19.4.3.6.3 Diagram....1356
19.4.3.6.4 Fields....1356
19.4.3.7 PCI Express Class Code Register (programming interface) (Class_Code_Register_a)....1357
19.4.3.7.1 Offset....1357
19.4.3.7.2 Function....1357
19.4.3.7.3 Diagram....1357
19.4.3.7.4 Fields....1357
19.4.3.8 PCI Express Class Code Register (sub class) (Class_Code_Register_b)....1358
19.4.3.8.1 Offset....1358
19.4.3.8.2 Function....1358
19.4.3.8.3 Diagram....1358
19.4.3.8.4 Fields....1358
19.4.3.9 PCI Express Class Code Register (base class) (Class_Code_Register_c)....1358
19.4.3.9.1 Offset....1358
19.4.3.9.2 Function....1359
19.4.3.9.3 Diagram....1359
19.4.3.9.4 Fields....1359
19.4.3.10 PCI Express Cache Line Size Register (Cache_Line_Size_Register)....1359
19.4.3.10.1 Offset....1359
19.4.3.10.2 Function....1360
19.4.3.10.3 Diagram....1360
19.4.3.10.4 Fields....1360
19.4.3.11 PCI Express Latency Timer Register (Latency_Timer_Register)....1360
19.4.3.11.1 Offset....1360
19.4.3.11.2 Function....1360

19.4.3.11.3 Diagram....1360
19.4.3.11.4 Fields....1361
19.4.3.12 PCI Express Header Type Register (Header_Type_Register)....1361
19.4.3.12.1 Offset....1361
19.4.3.12.2 Function....1361
19.4.3.12.3 Diagram....1361
19.4.3.12.4 Fields....1362
19.4.3.13 PCI Express Base Address Register 0 (BAR0)....1362
19.4.3.13.1 Offset....1362
19.4.3.13.2 Function....1362
19.4.3.13.3 Diagram....1362
19.4.3.13.4 Fields....1363
19.4.3.14 PCI Express Base Address Register 1 (BAR1)....1363
19.4.3.14.1 Offset....1363
19.4.3.14.2 Function....1363
19.4.3.14.3 Diagram....1363
19.4.3.14.4 Fields....1364
19.4.3.15 PCI Express Base Address Register 2 (BAR2)....1364
19.4.3.15.1 Offset....1364
19.4.3.15.2 Function....1364
19.4.3.15.3 Diagram....1365
19.4.3.15.4 Fields....1365
19.4.3.16 PCI Express Primary Bus Number Register (Primary_Bus_Number_Register)....1365
19.4.3.16.1 Offset....1366
19.4.3.16.2 Function....1366
19.4.3.16.3 Diagram....1366
19.4.3.16.4 Fields....1366
19.4.3.17 PCI Express Secondary Bus Number Register (Secondary_Bus_Number_Register)....1366
19.4.3.17.1 Offset....1366

19.4.3.17.2 Function.... 1366
19.4.3.17.3 Diagram....1367
19.4.3.17.4 Fields....1367
19.4.3.18 PCI Express Subordinate Bus Number Register (Subordinate_Bus_Number_Register).... 1367
19.4.3.18.1 Offset....1367
19.4.3.18.2 Function.... 1367
19.4.3.18.3 Diagram....1367
19.4.3.18.4 Fields....1368
19.4.3.19 PCI Express Base Address Register 3 (BAR3)....1368
19.4.3.19.1 Offset....1368
19.4.3.19.2 Function.... 1368
19.4.3.19.3 Diagram....1368
19.4.3.19.4 Fields....1368
19.4.3.20 PCI Express I/O Base Register (IO_Base_Register).... 1369
19.4.3.20.1 Offset....1369
19.4.3.20.2 Function.... 1369
19.4.3.20.3 Diagram....1369
19.4.3.20.4 Fields....1369
19.4.3.21 PCI Express I/O Limit Register (IO_Limit_Register).... 1370
19.4.3.21.1 Offset....1370
19.4.3.21.2 Function.... 1370
19.4.3.21.3 Diagram....1370
19.4.3.21.4 Fields....1370
19.4.3.22 PCI Express Secondary Status Register (Secondary_Status_Register).... 1371
19.4.3.22.1 Offset....1371
19.4.3.22.2 Function.... 1371
19.4.3.22.3 Diagram....1371
19.4.3.22.4 Fields....1371
19.4.3.23 PCI Express Base Address Register 4 (BAR4)....1372

19.4.3.23.1 Offset....1372
19.4.3.23.2 Function.... 1372
19.4.3.23.3 Diagram....1372
19.4.3.23.4 Fields....1373
19.4.3.24 PCI Express Memory Base Register (Memory_Base_Register).... 1373
19.4.3.24.1 Offset....1373
19.4.3.24.2 Function.... 1373
19.4.3.24.3 Diagram....1374
19.4.3.24.4 Fields....1374
19.4.3.25 PCI Express Memory Limit Register (Memory_Limit_Register).... 1374
19.4.3.25.1 Offset....1374
19.4.3.25.2 Function.... 1374
19.4.3.25.3 Diagram....1374
19.4.3.25.4 Fields....1375
19.4.3.26 PCI Express Base Address Register 5 (BAR5)....1375
19.4.3.26.1 Offset....1375
19.4.3.26.2 Function.... 1375
19.4.3.26.3 Diagram....1375
19.4.3.26.4 Fields....1376
19.4.3.27 PCI Express Prefetchable Memory Base Register (Prefetchable_Memory_Base_Register).. 1376
19.4.3.27.1 Offset....1376
19.4.3.27.2 Function.... 1376
19.4.3.27.3 Diagram....1376
19.4.3.27.4 Fields....1377
19.4.3.28 PCI Express Prefetchable Memory Limit Register (Prefetchable_Memory_Limit_Register) 1377
19.4.3.28.1 Offset....1377
19.4.3.28.2 Function.... 1377
19.4.3.28.3 Diagram....1377
19.4.3.28.4 Fields....1378

19.4.3.29 PCI Express Prefetchable Base Upper 32 Bits Register (Prefetchable_Base_Upper_32_Bits_Register)..... 1378

19.4.3.29.1 Offset.....1378

19.4.3.29.2 Function..... 1378

19.4.3.29.3 Diagram.....1378

19.4.3.29.4 Fields.....1379

19.4.3.30 PCI Express Prefetchable Limit Upper 32 Bits Register (Prefetchable_Limit_Upper_32_Bits_Register)..... 1379

19.4.3.30.1 Offset.....1379

19.4.3.30.2 Function..... 1379

19.4.3.30.3 Diagram.....1379

19.4.3.30.4 Fields.....1380

19.4.3.31 PCI Express Subsystem Vendor ID Register (Subsystem_Vendor_ID_Register)..... 1380

19.4.3.31.1 Offset.....1380

19.4.3.31.2 Function..... 1380

19.4.3.31.3 Diagram.....1380

19.4.3.31.4 Fields.....1381

19.4.3.32 PCI Express Subsystem ID Register (Subsystem_ID_Register)..... 1381

19.4.3.32.1 Offset.....1381

19.4.3.32.2 Function..... 1381

19.4.3.32.3 Diagram.....1381

19.4.3.32.4 Fields.....1381

19.4.3.33 PCI Express Expansion ROM Base Address Register (EP-Mode) (Expansion_ROM_BAR_Type0).....1382

19.4.3.33.1 Offset.....1382

19.4.3.33.2 Function..... 1382

19.4.3.33.3 Diagram.....1382

19.4.3.33.4 Fields.....1382

19.4.3.34 PCI Express I/O Base Upper 16 Bits Register (IO_Base_Upper_16_Bits_Register)..... 1383

19.4.3.34.1 Offset.....1383

19.4.3.34.2 Function.... 1383
19.4.3.34.3 Diagram....1383
19.4.3.34.4 Fields....1383
19.4.3.35 PCI Express I/O Limit Upper 16 Bits Register (IO_Limit_Upper_16_Bits_Register).... 1384
19.4.3.35.1 Offset....1384
19.4.3.35.2 Function.... 1384
19.4.3.35.3 Diagram....1384
19.4.3.35.4 Fields....1384
19.4.3.36 Capabilities Pointer Register (Capabilities_Pointer_Register).... 1384
19.4.3.36.1 Offset....1385
19.4.3.36.2 Function.... 1385
19.4.3.36.3 Diagram....1385
19.4.3.36.4 Fields....1385
19.4.3.37 PCI Express Expansion ROM Base Address Register (RC-Mode) (Expansion_ROM_BAR_Type1)....1385
19.4.3.37.1 Offset....1385
19.4.3.37.2 Function.... 1386
19.4.3.37.3 Diagram....1386
19.4.3.37.4 Fields....1386
19.4.3.38 PCI Express Interrupt Line Register (Interrupt_Line_Register)....1386
19.4.3.38.1 Offset....1386
19.4.3.38.2 Function.... 1387
19.4.3.38.3 Diagram....1387
19.4.3.38.4 Fields....1387
19.4.3.39 PCI Express Interrupt Pin Register (Interrupt_Pin_Register)....1387
19.4.3.39.1 Offset....1387
19.4.3.39.2 Function.... 1387
19.4.3.39.3 Diagram....1388
19.4.3.39.4 Fields....1388

Section number	Title	Page

19.4.3.40 PCI Express Bridge Control Register (Bridge_Control_Register)..... 1388

19.4.3.40.1 Offset.....1388

19.4.3.40.2 Function..... 1388

19.4.3.40.3 Diagram.....1388

19.4.3.40.4 Fields.....1389

19.4.3.41 PCI Express Minimum Grant Register (Minimum_Grant_Register)..... 1389

19.4.3.41.1 Offset.....1389

19.4.3.41.2 Function..... 1389

19.4.3.41.3 Diagram.....1389

19.4.3.41.4 Fields.....1390

19.4.3.42 PCI Express Maximum Latency Register (Maximum_Latency_Register)..... 1390

19.4.3.42.1 Offset.....1390

19.4.3.42.2 Function..... 1390

19.4.3.42.3 Diagram.....1390

19.4.3.42.4 Fields.....1390

19.4.3.43 PCI Express Power Management Capability ID Register (Power_Management_Capability_ID_Register).....1391

19.4.3.43.1 Offset.....1391

19.4.3.43.2 Diagram.....1391

19.4.3.43.3 Fields.....1391

19.4.3.44 PCI Express Power Management Capabilities Register (Power_Management_Capabilities_Register)..... 1391

19.4.3.44.1 Offset.....1391

19.4.3.44.2 Diagram.....1392

19.4.3.44.3 Fields.....1392

19.4.3.45 PCI Express Power Management Status and Control Register (Power_Management_Status_and_Control_Register).....1392

19.4.3.45.1 Offset.....1393

19.4.3.45.2 Diagram.....1393

19.4.3.45.3 Fields.....1393

19.4.3.46 PCI Express Power Management Data Register (Power_Management_Data_Register)..... 1394
19.4.3.46.1 Offset.....1394
19.4.3.46.2 Diagram.....1394
19.4.3.46.3 Fields.....1394
19.4.3.47 PCI Express MSI Message Capability ID Register (MSI_Message_Capability_ID_Register).....1394
19.4.3.47.1 Offset.....1394
19.4.3.47.2 Function..... 1395
19.4.3.47.3 Diagram.....1395
19.4.3.47.4 Fields.....1395
19.4.3.48 PCI Express MSI Message Control Register (MSI_Message_Control_Register).....1395
19.4.3.48.1 Offset.....1395
19.4.3.48.2 Function..... 1395
19.4.3.48.3 Diagram.....1396
19.4.3.48.4 Fields.....1396
19.4.3.49 PCI Express MSI Message Address Register (MSI_Message_Address_Register)..... 1396
19.4.3.49.1 Offset.....1396
19.4.3.49.2 Function..... 1396
19.4.3.49.3 Diagram.....1397
19.4.3.49.4 Fields.....1397
19.4.3.50 PCI Express MSI Message Upper Address Register (MSI_Message_Upper_Address_Register)..... 1397
19.4.3.50.1 Offset.....1397
19.4.3.50.2 Function..... 1397
19.4.3.50.3 Diagram.....1398
19.4.3.50.4 Fields.....1398
19.4.3.51 PCI Express MSI Message Data Register (MSI_Message_Data_Register)..... 1398
19.4.3.51.1 Offset.....1398
19.4.3.51.2 Function..... 1398

19.4.3.51.3 Diagram....1398
19.4.3.51.4 Fields....1399
19.4.3.52 PCI Express Capability ID Register (Capability_ID_Register)....1399
19.4.3.52.1 Offset....1399
19.4.3.52.2 Diagram....1399
19.4.3.52.3 Fields....1399
19.4.3.53 PCI Express Capabilities Register (Capabilities_Register)....1400
19.4.3.53.1 Offset....1400
19.4.3.53.2 Diagram....1400
19.4.3.53.3 Fields....1400
19.4.3.54 PCI Express Device Capabilities Register (Device_Capabilities_Register)....1401
19.4.3.54.1 Offset....1401
19.4.3.54.2 Diagram....1401
19.4.3.54.3 Fields....1402
19.4.3.55 PCI Express Device Control Register (Device_Control_Register)....1402
19.4.3.55.1 Offset....1402
19.4.3.55.2 Diagram....1403
19.4.3.55.3 Fields....1403
19.4.3.56 PCI Express Device Status Register (Device_Status_Register)....1404
19.4.3.56.1 Offset....1404
19.4.3.56.2 Diagram....1404
19.4.3.56.3 Fields....1404
19.4.3.57 PCI Express Link Capabilities Register (Link_Capabilities_Register)....1405
19.4.3.57.1 Offset....1405
19.4.3.57.2 Diagram....1405
19.4.3.57.3 Fields....1405
19.4.3.58 PCI Express Link Control Register (Link_Control_Register)....1406
19.4.3.58.1 Offset....1406
19.4.3.58.2 Diagram....1406

19.4.3.58.3 Fields....1407
19.4.3.59 PCI Express Link Status Register (Link_Status_Register)....1408
19.4.3.59.1 Offset....1408
19.4.3.59.2 Diagram....1408
19.4.3.59.3 Fields....1408
19.4.3.60 PCI Express Slot Capabilities Register (Slot_Capabilities_Register)....1409
19.4.3.60.1 Offset....1409
19.4.3.60.2 Function....1409
19.4.3.60.3 Diagram....1409
19.4.3.60.4 Fields....1410
19.4.3.61 PCI Express Slot Control Register (Slot_Control_Register)....1410
19.4.3.61.1 Offset....1410
19.4.3.61.2 Function....1411
19.4.3.61.3 Diagram....1411
19.4.3.61.4 Fields....1411
19.4.3.62 PCI Express Slot Status Register (Slot_Status_Register)....1412
19.4.3.62.1 Offset....1412
19.4.3.62.2 Function....1412
19.4.3.62.3 Diagram....1412
19.4.3.62.4 Fields....1412
19.4.3.63 PCI Express Root Control Register (Root_Control_Register)....1413
19.4.3.63.1 Offset....1413
19.4.3.63.2 Function....1413
19.4.3.63.3 Diagram....1413
19.4.3.63.4 Fields....1414
19.4.3.64 PCI Express Root Capabilities Register (Root_Capabilities_Register)....1414
19.4.3.64.1 Offset....1414
19.4.3.64.2 Function....1414
19.4.3.64.3 Diagram....1414

19.4.3.64.4 Fields....1415
19.4.3.65 PCI Express Root Status Register (Root_Status_Register).... 1415
19.4.3.65.1 Offset....1415
19.4.3.65.2 Function.... 1415
19.4.3.65.3 Diagram....1415
19.4.3.65.4 Fields....1416
19.4.3.66 PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)....1416
19.4.3.66.1 Offset....1416
19.4.3.66.2 Diagram....1416
19.4.3.66.3 Fields....1417
19.4.3.67 PCI Express Device Control 2 Register (Device_Control_2_Register).... 1417
19.4.3.67.1 Offset....1417
19.4.3.67.2 Diagram....1417
19.4.3.67.3 Fields....1418
19.4.3.68 PCI Express Link Capabilities 2 Register (Link_Capabilities_2_Register).... 1418
19.4.3.68.1 Offset....1418
19.4.3.68.2 Function.... 1418
19.4.3.68.3 Diagram....1419
19.4.3.68.4 Fields....1419
19.4.3.69 PCI Express Link Control 2 Register (Link_Control_2_Register).... 1420
19.4.3.69.1 Offset....1420
19.4.3.69.2 Diagram....1420
19.4.3.69.3 Fields....1420
19.4.3.70 PCI Express Link Status 2 Register (Link_Status_2_Register)....1421
19.4.3.70.1 Offset....1421
19.4.3.70.2 Diagram....1421
19.4.3.70.3 Fields....1421
19.4.3.71 PCI Express MSI-X Message Capability ID Register (MSI_X_Message_Capability_ID_Register)....1421

19.4.3.71.1 Offset....1422
19.4.3.71.2 Function.... 1422
19.4.3.71.3 Diagram....1422
19.4.3.71.4 Fields....1422
19.4.3.72 PCI Express MSI-X Message Control Register (MSI_X_Message_Control_Register)....1422
19.4.3.72.1 Offset....1423
19.4.3.72.2 Function.... 1423
19.4.3.72.3 Diagram....1423
19.4.3.72.4 Fields....1423
19.4.3.73 PCI Express MSI-X Table Offset/BIR Register (MSI_X_Table_Offset_BIR_Register).... 1423
19.4.3.73.1 Offset....1424
19.4.3.73.2 Function.... 1424
19.4.3.73.3 Diagram....1424
19.4.3.73.4 Fields....1424
19.4.3.74 PCI Express MSI-X PBA Offset/BIR Register (MSI_X_PBA_Offset_BIR_Register).... 1425
19.4.3.74.1 Offset....1425
19.4.3.74.2 Function.... 1425
19.4.3.74.3 Diagram....1425
19.4.3.74.4 Fields....1425
19.4.3.75 PCI Express Advanced Error Reporting Capability ID Register (Advanced_Error_Reporting_Capability_ID_Register).... 1426
19.4.3.75.1 Offset....1426
19.4.3.75.2 Diagram....1426
19.4.3.75.3 Fields....1426
19.4.3.76 PCI Express Uncorrectable Error Status Register (Uncorrectable_Error_Status_Register)....1426
19.4.3.76.1 Offset....1427
19.4.3.76.2 Diagram....1427
19.4.3.76.3 Fields....1427
19.4.3.77 PCI Express Uncorrectable Error Mask Register (Uncorrectable_Error_Mask_Register)....1428

19.4.3.77.1 Offset....1428
19.4.3.77.2 Diagram....1428
19.4.3.77.3 Fields....1429
19.4.3.78 PCI Express Uncorrectable Error Severity Register (Uncorrectable_Error_Severity_Register).... 1430
19.4.3.78.1 Offset....1430
19.4.3.78.2 Diagram....1430
19.4.3.78.3 Fields....1430
19.4.3.79 PCI Express Correctable Error Status Register (Correctable_Error_Status_Register)....1431
19.4.3.79.1 Offset....1431
19.4.3.79.2 Diagram....1431
19.4.3.79.3 Fields....1432
19.4.3.80 PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register)....1432
19.4.3.80.1 Offset....1433
19.4.3.80.2 Diagram....1433
19.4.3.80.3 Fields....1433
19.4.3.81 PCI Express Advanced Error Capabilities and Control Register (Advanced_Error_Capabilities_and_Control_Register).... 1434
19.4.3.81.1 Offset....1434
19.4.3.81.2 Diagram....1434
19.4.3.81.3 Fields....1434
19.4.3.82 PCI Express Header Log Register 1 (Header_Log_Register_DWORD1).... 1435
19.4.3.82.1 Offset....1435
19.4.3.82.2 Function.... 1435
19.4.3.82.3 Diagram....1435
19.4.3.82.4 Fields....1436
19.4.3.83 PCI Express Header Log Register 2 (Header_Log_Register_DWORD2).... 1436
19.4.3.83.1 Offset....1436
19.4.3.83.2 Function.... 1436

19.4.3.83.3 Diagram....1436
19.4.3.83.4 Fields....1437
19.4.3.84 PCI Express Header Log Register 3 (Header_Log_Register_DWORD3).... 1437
19.4.3.84.1 Offset....1437
19.4.3.84.2 Function.... 1437
19.4.3.84.3 Diagram....1437
19.4.3.84.4 Fields....1438
19.4.3.85 PCI Express Header Log Register 4 (Header_Log_Register_DWORD4).... 1438
19.4.3.85.1 Offset....1438
19.4.3.85.2 Function.... 1438
19.4.3.85.3 Diagram....1438
19.4.3.85.4 Fields....1439
19.4.3.86 PCI Express Root Error Command Register (Root_Error_Command_Register)....1439
19.4.3.86.1 Offset....1439
19.4.3.86.2 Function.... 1439
19.4.3.86.3 Diagram....1439
19.4.3.86.4 Fields....1440
19.4.3.87 PCI Express Root Error Status Register (Root_Error_Status_Register).... 1440
19.4.3.87.1 Offset....1440
19.4.3.87.2 Function.... 1440
19.4.3.87.3 Diagram....1441
19.4.3.87.4 Fields....1441
19.4.3.88 PCI Express Correctable Error Source ID Register (Correctable_Error_Source_ID_Register).... 1441
19.4.3.88.1 Offset....1442
19.4.3.88.2 Diagram....1442
19.4.3.88.3 Fields....1442
19.4.3.89 PCI Express Error Source ID Register (Error_Source_ID_Register).... 1442
19.4.3.89.1 Offset....1442

19.4.3.89.2 Diagram....1442
19.4.3.89.3 Fields....1443
19.4.3.90 ARI Capability Header (ARI_Capability_Header)....1443
19.4.3.90.1 Offset....1443
19.4.3.90.2 Function....1443
19.4.3.90.3 Diagram....1443
19.4.3.90.4 Fields....1444
19.4.3.91 ARI Capability Register (ARI_Capability_Register)....1444
19.4.3.91.1 Offset....1444
19.4.3.91.2 Function....1444
19.4.3.91.3 Diagram....1444
19.4.3.91.4 Fields....1444
19.4.3.92 ARI Control Register (ARI_Control_Register)....1445
19.4.3.92.1 Offset....1445
19.4.3.92.2 Function....1445
19.4.3.92.3 Diagram....1445
19.4.3.92.4 Fields....1445
19.4.3.93 Secondary PCI Express Extended Capability Header (SPCIE_CAP_HEADER_REG)....1446
19.4.3.93.1 Offset....1446
19.4.3.93.2 Diagram....1446
19.4.3.93.3 Fields....1446
19.4.3.94 Link Control 3 Register (LINK_CONTROL3_REG)....1447
19.4.3.94.1 Offset....1447
19.4.3.94.2 Diagram....1447
19.4.3.94.3 Fields....1447
19.4.3.95 Lane Error Status Register (LANE_ERR_STATUS_REG)....1448
19.4.3.95.1 Offset....1448
19.4.3.95.2 Diagram....1448
19.4.3.95.3 Fields....1448

19.4.3.96 Lane Equalization Control Register (LANE0_EQUALIZATION_CONTROL - LANE3_EQUALIZATION_CONTROL)..... 1449

19.4.3.96.1 Offset.....1449

19.4.3.96.2 Diagram.....1449

19.4.3.96.3 Fields.....1449

19.4.3.97 SR-IOV Extended Capability Header (SR_IOV_Extended_Capability_ID_Register)..... 1450

19.4.3.97.1 Offset.....1450

19.4.3.97.2 Function..... 1450

19.4.3.97.3 Diagram.....1450

19.4.3.97.4 Fields.....1451

19.4.3.98 PCI Express SR-IOV Capabilities Register (SR_IOV_Capabilities_Register)..... 1451

19.4.3.98.1 Offset.....1451

19.4.3.98.2 Function..... 1452

19.4.3.98.3 Diagram.....1452

19.4.3.98.4 Fields.....1452

19.4.3.99 PCI Express SR-IOV Control Register (SR_IOV_Control_Register).....1452

19.4.3.99.1 Offset.....1452

19.4.3.99.2 Function..... 1453

19.4.3.99.3 Diagram.....1453

19.4.3.99.4 Fields.....1453

19.4.3.100 PCI Express SR-IOV Status Register (SR_IOV_Status_Register).....1454

19.4.3.100.1 Offset.....1454

19.4.3.100.2 Function..... 1454

19.4.3.100.3 Diagram.....1454

19.4.3.100.4 Fields.....1454

19.4.3.101 PCI Express SR-IOV Initial VF Register (SR_IOV_Initial_VF_Register).....1454

19.4.3.101.1 Offset.....1454

19.4.3.101.2 Function..... 1455

19.4.3.101.3 Diagram.....1455

19.4.3.101.4 Fields....1455

19.4.3.102 PCI Express SR-IOV Total VF Register (SR_IOV_Total_VF_Register)....1455

19.4.3.102.1 Offset....1455

19.4.3.102.2 Function....1455

19.4.3.102.3 Diagram....1456

19.4.3.102.4 Fields....1456

19.4.3.103 PCI Express SR-IOV Number VF Register (SR_IOV_Number_VF_Register)....1456

19.4.3.103.1 Offset....1456

19.4.3.103.2 Function....1456

19.4.3.103.3 Diagram....1456

19.4.3.103.4 Fields....1456

19.4.3.104 PCI Express SR-IOV Function Dependency Link Register (SR_IOV_Function_Dependency_Link_Register)....1457

19.4.3.104.1 Offset....1457

19.4.3.104.2 Function....1457

19.4.3.104.3 Diagram....1457

19.4.3.104.4 Fields....1457

19.4.3.105 PCI Express SR-IOV First VF Offset Register (SR_IOV_First_VF_Offset_Register)....1457

19.4.3.105.1 Offset....1458

19.4.3.105.2 Function....1458

19.4.3.105.3 Diagram....1458

19.4.3.105.4 Fields....1458

19.4.3.106 PCI Express SR-IOV VF Stride Register (SR_IOV_VF_Stride_Register)....1458

19.4.3.106.1 Offset....1458

19.4.3.106.2 Function....1459

19.4.3.106.3 Diagram....1459

19.4.3.106.4 Fields....1459

19.4.3.107 PCI Express SR-IOV Device ID Register (SR_IOV_Device_ID_Register)....1459

19.4.3.107.1 Offset....1459

19.4.3.107.2 Function.... 1459
19.4.3.107.3 Diagram....1459
19.4.3.107.4 Fields....1460
19.4.3.108 PCI Express SR-IOV Supported Page Size Register (SR_IOV_Supported_Page_Size_Register)....1460
19.4.3.108.1 Offset....1460
19.4.3.108.2 Function.... 1460
19.4.3.108.3 Diagram....1460
19.4.3.108.4 Fields....1460
19.4.3.109 PCI Express SR-IOV System Page Size Register (SR_IOV_System_Page_Size_Register).. 1461
19.4.3.109.1 Offset....1461
19.4.3.109.2 Function.... 1461
19.4.3.109.3 Diagram....1461
19.4.3.109.4 Fields....1461
19.4.3.110 PCI Express SR-IOV VF Base Address Register 0 (VFBAR0).... 1462
19.4.3.110.1 Offset....1462
19.4.3.110.2 Function.... 1462
19.4.3.110.3 Diagram....1462
19.4.3.110.4 Fields....1462
19.4.3.111 PCI Express SR-IOV VF Base Address Register 1 (VFBAR1).... 1463
19.4.3.111.1 Offset....1463
19.4.3.111.2 Function.... 1463
19.4.3.111.3 Diagram....1463
19.4.3.111.4 Fields....1464
19.4.3.112 PCI Express SR-IOV VF Base Address Register 2 (VFBAR2).... 1464
19.4.3.112.1 Offset....1464
19.4.3.112.2 Function.... 1464
19.4.3.112.3 Diagram....1465
19.4.3.112.4 Fields....1465

19.4.3.113 PCI Express SR-IOV VF Base Address Register 3 (VFBAR3)..... 1465
19.4.3.113.1 Offset.....1466
19.4.3.113.2 Function..... 1466
19.4.3.113.3 Diagram.....1466
19.4.3.113.4 Fields.....1466
19.4.3.114 PCI Express SR-IOV VF Base Address Register 4 (VFBAR4)..... 1466
19.4.3.114.1 Offset.....1467
19.4.3.114.2 Function..... 1467
19.4.3.114.3 Diagram.....1467
19.4.3.114.4 Fields.....1467
19.4.3.115 PCI Express SR-IOV VF Base Address Register 5 (VFBAR5)..... 1468
19.4.3.115.1 Offset.....1468
19.4.3.115.2 Function..... 1468
19.4.3.115.3 Diagram.....1468
19.4.3.115.4 Fields.....1469
19.4.3.116 PCI Express SR-IOV Migration State Array Offset Register (SR_IOV_Migration_State_Array_Offset_Register)..... 1469
19.4.3.116.1 Offset.....1469
19.4.3.116.2 Function..... 1469
19.4.3.116.3 Diagram.....1469
19.4.3.116.4 Fields.....1469
19.4.3.117 Symbol Timer Register and Filter Mask 1 Register (SYMBOL_TIMER_FILTER_1_OFF). 1470
19.4.3.117.1 Offset.....1470
19.4.3.117.2 Function..... 1470
19.4.3.117.3 Diagram.....1470
19.4.3.117.4 Fields.....1470
19.4.3.118 Gen3 Control Register (GEN3_RELATED_OFF)..... 1473
19.4.3.118.1 Offset.....1473
19.4.3.118.2 Diagram.....1473

19.4.3.118.3 Fields....1473
19.4.3.119 DBI Read-only Write Enable Register (MISC_CONTROL_1_OFF)....1474
19.4.3.119.1 Offset....1474
19.4.3.119.2 Diagram....1474
19.4.3.119.3 Fields....1475
19.4.3.120 Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)....1475
19.4.3.120.1 Offset....1475
19.4.3.120.2 Diagram....1475
19.4.3.120.3 Fields....1476
19.4.3.121 Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)....1476
19.4.3.121.1 Offset....1477
19.4.3.121.2 Diagram....1477
19.4.3.121.3 Fields....1477
19.4.3.122 Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)....1477
19.4.3.122.1 Offset....1477
19.4.3.122.2 Diagram....1478
19.4.3.122.3 Fields....1478
19.4.3.123 iATU Index Register (IATU_VIEWPORT_OFF)....1478
19.4.3.123.1 Offset....1478
19.4.3.123.2 Function....1478
19.4.3.123.3 Diagram....1478
19.4.3.123.4 Fields....1479
19.4.3.124 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_INBOUND_0)....1479
19.4.3.124.1 Offset....1479
19.4.3.124.2 Diagram....1479
19.4.3.124.3 Fields....1480
19.4.3.125 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)....1481
19.4.3.125.1 Offset....1481
19.4.3.125.2 Diagram....1481

19.4.3.125.3 Fields....1481

19.4.3.126 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)....1482

19.4.3.126.1 Offset....1482

19.4.3.126.2 Diagram....1482

19.4.3.126.3 Fields....1483

19.4.3.127 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)....1485

19.4.3.127.1 Offset....1485

19.4.3.127.2 Diagram....1485

19.4.3.127.3 Fields....1486

19.4.3.128 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)....1487

19.4.3.128.1 Offset....1487

19.4.3.128.2 Diagram....1487

19.4.3.128.3 Fields....1487

19.4.3.129 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0) 1487

19.4.3.129.1 Offset....1487

19.4.3.129.2 Diagram....1488

19.4.3.129.3 Fields....1488

19.4.3.130 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0). 1488

19.4.3.130.1 Offset....1488

19.4.3.130.2 Diagram....1489

19.4.3.130.3 Fields....1489

19.4.3.131 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)....1489

19.4.3.131.1 Offset....1489

19.4.3.131.2 Diagram....1489

19.4.3.131.3 Fields....1490

19.4.3.132 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_INBOUND_0)....1490

19.4.3.132.1 Offset....1490

19.4.3.132.2 Diagram....1490

19.4.3.132.3 Fields....1491

19.4.3.133 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)....1491

19.4.3.133.1 Offset....1491

19.4.3.133.2 Diagram....1491

19.4.3.133.3 Fields....1492

19.4.3.134 iATU Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)....1492

19.4.3.134.1 Offset....1492

19.4.3.134.2 Diagram....1492

19.4.3.134.3 Fields....1493

19.4.3.135 iATU Outbound Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)....1493

19.4.3.135.1 Offset....1493

19.4.3.135.2 Diagram....1494

19.4.3.135.3 Fields....1494

19.4.3.136 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0)....1494

19.4.3.136.1 Offset....1494

19.4.3.136.2 Diagram....1494

19.4.3.136.3 Fields....1495

19.4.3.137 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)....1495

19.4.3.137.1 Offset....1495

19.4.3.137.2 Diagram....1495

19.4.3.137.3 Fields....1496

19.4.3.138 Base Address Register 0 Mask (BAR0_MASK)....1496

19.4.3.138.1 Offset....1496

19.4.3.138.2 Function....1496

19.4.3.138.3 Diagram....1496

19.4.3.138.4 Fields....1497

Section number	Title	Page

19.4.3.139 Base Address Register 1 Mask (BAR1_MASK) 1497

19.4.3.139.1 Offset 1497

19.4.3.139.2 Function 1497

19.4.3.139.3 Diagram 1497

19.4.3.139.4 Fields 1498

19.4.3.140 Base Address Register 2 Mask (BAR2_MASK) 1498

19.4.3.140.1 Offset 1498

19.4.3.140.2 Function 1498

19.4.3.140.3 Diagram 1499

19.4.3.140.4 Fields 1499

19.4.3.141 Base Address Register 3 Mask (BAR3_MASK) 1499

19.4.3.141.1 Offset 1499

19.4.3.141.2 Function 1499

19.4.3.141.3 Diagram 1500

19.4.3.141.4 Fields 1500

19.4.3.142 Base Address Register 4 Mask (BAR4_MASK) 1500

19.4.3.142.1 Offset 1500

19.4.3.142.2 Function 1501

19.4.3.142.3 Diagram 1501

19.4.3.142.4 Fields 1501

19.4.3.143 Base Address Register 5 Mask (BAR5_MASK) 1501

19.4.3.143.1 Offset 1501

19.4.3.143.2 Function 1502

19.4.3.143.3 Diagram 1502

19.4.3.143.4 Fields 1502

19.4.3.144 Expansion ROM Base Address Register Mask (EP mode) (EXP_ROM_BAR_MASK_EP) 1502

19.4.3.144.1 Offset 1503

19.4.3.144.2 Function 1503

19.4.3.144.3 Diagram 1503

19.4.3.144.4 Fields....1503

19.4.3.145 Expansion ROM Base Address Register Mask (RC mode) (EXP_ROM_BAR_MASK_RC)....1504

19.4.3.145.1 Offset....1504

19.4.3.145.2 Function....1504

19.4.3.145.3 Diagram....1504

19.4.3.145.4 Fields....1505

19.4.4 Non-SR-IOV Instances (PEX2 and PEX3) register descriptions....1505

19.4.4.1 PCI_Express_Configuration_Registers memory map....1505

19.4.4.2 PCI Express Vendor ID Register (Vendor_ID_Register)....1509

19.4.4.2.1 Offset....1509

19.4.4.2.2 Function....1510

19.4.4.2.3 Diagram....1510

19.4.4.2.4 Fields....1510

19.4.4.3 PCI Express Device ID Register (Device_ID_Register)....1510

19.4.4.3.1 Offset....1510

19.4.4.3.2 Function....1510

19.4.4.3.3 Diagram....1511

19.4.4.3.4 Fields....1511

19.4.4.4 PCI Express Command Register (Command_Register)....1511

19.4.4.4.1 Offset....1511

19.4.4.4.2 Function....1511

19.4.4.4.3 Diagram....1511

19.4.4.4.4 Fields....1512

19.4.4.5 PCI Express Status Register (Status_Register)....1513

19.4.4.5.1 Offset....1513

19.4.4.5.2 Function....1513

19.4.4.5.3 Diagram....1513

19.4.4.5.4 Fields....1514

19.4.4.6 PCI Express Revision ID Register (Revision_ID_Register).....1515
19.4.4.6.1 Offset.....1515
19.4.4.6.2 Function..... 1515
19.4.4.6.3 Diagram.....1515
19.4.4.6.4 Fields.....1515
19.4.4.7 PCI Express Class Code Register (programming interface) (Class_Code_Register_a)..... 1516
19.4.4.7.1 Offset.....1516
19.4.4.7.2 Function..... 1516
19.4.4.7.3 Diagram.....1516
19.4.4.7.4 Fields.....1516
19.4.4.8 PCI Express Class Code Register (sub class) (Class_Code_Register_b)..... 1517
19.4.4.8.1 Offset.....1517
19.4.4.8.2 Function..... 1517
19.4.4.8.3 Diagram.....1517
19.4.4.8.4 Fields.....1517
19.4.4.9 PCI Express Class Code Register (base class) (Class_Code_Register_c)..... 1517
19.4.4.9.1 Offset.....1517
19.4.4.9.2 Function..... 1518
19.4.4.9.3 Diagram.....1518
19.4.4.9.4 Fields.....1518
19.4.4.10 PCI Express Cache Line Size Register (Cache_Line_Size_Register)..... 1518
19.4.4.10.1 Offset.....1518
19.4.4.10.2 Function..... 1519
19.4.4.10.3 Diagram.....1519
19.4.4.10.4 Fields.....1519
19.4.4.11 PCI Express Latency Timer Register (Latency_Timer_Register)..... 1519
19.4.4.11.1 Offset.....1519
19.4.4.11.2 Function..... 1519
19.4.4.11.3 Diagram.....1519

19.4.4.11.4 Fields....1520
19.4.4.12 PCI Express Header Type Register (Header_Type_Register)....1520
19.4.4.12.1 Offset....1520
19.4.4.12.2 Function....1520
19.4.4.12.3 Diagram....1520
19.4.4.12.4 Fields....1521
19.4.4.13 PCI Express Base Address Register 0 (BAR0)....1521
19.4.4.13.1 Offset....1521
19.4.4.13.2 Function....1521
19.4.4.13.3 Diagram....1521
19.4.4.13.4 Fields....1522
19.4.4.14 PCI Express Base Address Register 1 (BAR1)....1522
19.4.4.14.1 Offset....1522
19.4.4.14.2 Function....1522
19.4.4.14.3 Diagram....1522
19.4.4.14.4 Fields....1523
19.4.4.15 PCI Express Base Address Register 2 (BAR2)....1523
19.4.4.15.1 Offset....1523
19.4.4.15.2 Function....1523
19.4.4.15.3 Diagram....1524
19.4.4.15.4 Fields....1524
19.4.4.16 PCI Express Primary Bus Number Register (Primary_Bus_Number_Register)....1524
19.4.4.16.1 Offset....1525
19.4.4.16.2 Function....1525
19.4.4.16.3 Diagram....1525
19.4.4.16.4 Fields....1525
19.4.4.17 PCI Express Secondary Bus Number Register (Secondary_Bus_Number_Register)....1525
19.4.4.17.1 Offset....1525
19.4.4.17.2 Function....1525

19.4.4.17.3 Diagram....1526
19.4.4.17.4 Fields....1526
19.4.4.18 PCI Express Subordinate Bus Number Register (Subordinate_Bus_Number_Register)....1526
19.4.4.18.1 Offset....1526
19.4.4.18.2 Function....1526
19.4.4.18.3 Diagram....1526
19.4.4.18.4 Fields....1527
19.4.4.19 PCI Express Base Address Register 3 (BAR3)....1527
19.4.4.19.1 Offset....1527
19.4.4.19.2 Function....1527
19.4.4.19.3 Diagram....1527
19.4.4.19.4 Fields....1527
19.4.4.20 PCI Express I/O Base Register (IO_Base_Register)....1528
19.4.4.20.1 Offset....1528
19.4.4.20.2 Function....1528
19.4.4.20.3 Diagram....1528
19.4.4.20.4 Fields....1528
19.4.4.21 PCI Express I/O Limit Register (IO_Limit_Register)....1529
19.4.4.21.1 Offset....1529
19.4.4.21.2 Function....1529
19.4.4.21.3 Diagram....1529
19.4.4.21.4 Fields....1529
19.4.4.22 PCI Express Secondary Status Register (Secondary_Status_Register)....1530
19.4.4.22.1 Offset....1530
19.4.4.22.2 Function....1530
19.4.4.22.3 Diagram....1530
19.4.4.22.4 Fields....1530
19.4.4.23 PCI Express Base Address Register 4 (BAR4)....1531
19.4.4.23.1 Offset....1531

19.4.4.23.2 Function.... 1531
19.4.4.23.3 Diagram....1531
19.4.4.23.4 Fields....1532
19.4.4.24 PCI Express Memory Base Register (Memory_Base_Register).... 1532
19.4.4.24.1 Offset....1532
19.4.4.24.2 Function.... 1532
19.4.4.24.3 Diagram....1533
19.4.4.24.4 Fields....1533
19.4.4.25 PCI Express Memory Limit Register (Memory_Limit_Register).... 1533
19.4.4.25.1 Offset....1533
19.4.4.25.2 Function.... 1533
19.4.4.25.3 Diagram....1533
19.4.4.25.4 Fields....1534
19.4.4.26 PCI Express Base Address Register 5 (BAR5)....1534
19.4.4.26.1 Offset....1534
19.4.4.26.2 Function.... 1534
19.4.4.26.3 Diagram....1534
19.4.4.26.4 Fields....1535
19.4.4.27 PCI Express Prefetchable Memory Base Register (Prefetchable_Memory_Base_Register).. 1535
19.4.4.27.1 Offset....1535
19.4.4.27.2 Function.... 1535
19.4.4.27.3 Diagram....1535
19.4.4.27.4 Fields....1536
19.4.4.28 PCI Express Prefetchable Memory Limit Register (Prefetchable_Memory_Limit_Register) 1536
19.4.4.28.1 Offset....1536
19.4.4.28.2 Function.... 1536
19.4.4.28.3 Diagram....1536
19.4.4.28.4 Fields....1537

19.4.4.29 PCI Express Prefetchable Base Upper 32 Bits Register (Prefetchable_Base_Upper_32_Bits_Register)..... 1537

19.4.4.29.1 Offset.....1537

19.4.4.29.2 Function..... 1537

19.4.4.29.3 Diagram.....1537

19.4.4.29.4 Fields.....1538

19.4.4.30 PCI Express Prefetchable Limit Upper 32 Bits Register (Prefetchable_Limit_Upper_32_Bits_Register)..... 1538

19.4.4.30.1 Offset.....1538

19.4.4.30.2 Function..... 1538

19.4.4.30.3 Diagram.....1538

19.4.4.30.4 Fields.....1539

19.4.4.31 PCI Express Subsystem Vendor ID Register (Subsystem_Vendor_ID_Register)..... 1539

19.4.4.31.1 Offset.....1539

19.4.4.31.2 Function..... 1539

19.4.4.31.3 Diagram.....1539

19.4.4.31.4 Fields.....1540

19.4.4.32 PCI Express Subsystem ID Register (Subsystem_ID_Register)..... 1540

19.4.4.32.1 Offset.....1540

19.4.4.32.2 Function..... 1540

19.4.4.32.3 Diagram.....1540

19.4.4.32.4 Fields.....1540

19.4.4.33 PCI Express Expansion ROM Base Address Register (EP-Mode) (Expansion_ROM_BAR_Type0).....1541

19.4.4.33.1 Offset.....1541

19.4.4.33.2 Function..... 1541

19.4.4.33.3 Diagram.....1541

19.4.4.33.4 Fields.....1541

19.4.4.34 PCI Express I/O Base Upper 16 Bits Register (IO_Base_Upper_16_Bits_Register)..... 1542

19.4.4.34.1 Offset.....1542

19.4.4.34.2 Function.... 1542
19.4.4.34.3 Diagram....1542
19.4.4.34.4 Fields....1542
19.4.4.35 PCI Express I/O Limit Upper 16 Bits Register (IO_Limit_Upper_16_Bits_Register).... 1543
19.4.4.35.1 Offset....1543
19.4.4.35.2 Function.... 1543
19.4.4.35.3 Diagram....1543
19.4.4.35.4 Fields....1543
19.4.4.36 Capabilities Pointer Register (Capabilities_Pointer_Register).... 1543
19.4.4.36.1 Offset....1544
19.4.4.36.2 Function.... 1544
19.4.4.36.3 Diagram....1544
19.4.4.36.4 Fields....1544
19.4.4.37 PCI Express Expansion ROM Base Address Register (RC-Mode) (Expansion_ROM_BAR_Type1)....1544
19.4.4.37.1 Offset....1544
19.4.4.37.2 Function.... 1545
19.4.4.37.3 Diagram....1545
19.4.4.37.4 Fields....1545
19.4.4.38 PCI Express Interrupt Line Register (Interrupt_Line_Register)....1545
19.4.4.38.1 Offset....1545
19.4.4.38.2 Function.... 1546
19.4.4.38.3 Diagram....1546
19.4.4.38.4 Fields....1546
19.4.4.39 PCI Express Interrupt Pin Register (Interrupt_Pin_Register)....1546
19.4.4.39.1 Offset....1546
19.4.4.39.2 Function.... 1546
19.4.4.39.3 Diagram....1547
19.4.4.39.4 Fields....1547

Section number	Title	Page

19.4.4.40 PCI Express Bridge Control Register (Bridge_Control_Register)..... 1547

19.4.4.40.1 Offset.....1547

19.4.4.40.2 Function..... 1547

19.4.4.40.3 Diagram.....1547

19.4.4.40.4 Fields.....1548

19.4.4.41 PCI Express Minimum Grant Register (Minimum_Grant_Register)..... 1548

19.4.4.41.1 Offset.....1548

19.4.4.41.2 Function..... 1548

19.4.4.41.3 Diagram.....1548

19.4.4.41.4 Fields.....1549

19.4.4.42 PCI Express Maximum Latency Register (Maximum_Latency_Register)..... 1549

19.4.4.42.1 Offset.....1549

19.4.4.42.2 Function..... 1549

19.4.4.42.3 Diagram.....1549

19.4.4.42.4 Fields.....1549

19.4.4.43 PCI Express Power Management Capability ID Register (Power_Management_Capability_ID_Register).....1550

19.4.4.43.1 Offset.....1550

19.4.4.43.2 Diagram.....1550

19.4.4.43.3 Fields.....1550

19.4.4.44 PCI Express Power Management Capabilities Register (Power_Management_Capabilities_Register)..... 1550

19.4.4.44.1 Offset.....1550

19.4.4.44.2 Diagram.....1551

19.4.4.44.3 Fields.....1551

19.4.4.45 PCI Express Power Management Status and Control Register (Power_Management_Status_and_Control_Register).....1551

19.4.4.45.1 Offset.....1552

19.4.4.45.2 Diagram.....1552

19.4.4.45.3 Fields.....1552

Section number	Title	Page

19.4.4.46 PCI Express Power Management Data Register (Power_Management_Data_Register)..... 1553

19.4.4.46.1 Offset.....1553

19.4.4.46.2 Diagram.....1553

19.4.4.46.3 Fields.....1553

19.4.4.47 PCI Express MSI Message Capability ID Register (MSI_Message_Capability_ID_Register).....1553

19.4.4.47.1 Offset.....1553

19.4.4.47.2 Function..... 1554

19.4.4.47.3 Diagram.....1554

19.4.4.47.4 Fields.....1554

19.4.4.48 PCI Express MSI Message Control Register (MSI_Message_Control_Register).....1554

19.4.4.48.1 Offset.....1554

19.4.4.48.2 Function..... 1554

19.4.4.48.3 Diagram.....1555

19.4.4.48.4 Fields.....1555

19.4.4.49 PCI Express MSI Message Address Register (MSI_Message_Address_Register)..... 1555

19.4.4.49.1 Offset.....1555

19.4.4.49.2 Function..... 1555

19.4.4.49.3 Diagram.....1556

19.4.4.49.4 Fields.....1556

19.4.4.50 PCI Express MSI Message Upper Address Register (MSI_Message_Upper_Address_Register)..... 1556

19.4.4.50.1 Offset.....1556

19.4.4.50.2 Function..... 1556

19.4.4.50.3 Diagram.....1557

19.4.4.50.4 Fields.....1557

19.4.4.51 PCI Express MSI Message Data Register (MSI_Message_Data_Register)..... 1557

19.4.4.51.1 Offset.....1557

19.4.4.51.2 Function..... 1557

19.4.4.51.3 Diagram....1557
19.4.4.51.4 Fields....1558
19.4.4.52 PCI Express Capability ID Register (Capability_ID_Register)....1558
19.4.4.52.1 Offset....1558
19.4.4.52.2 Diagram....1558
19.4.4.52.3 Fields....1558
19.4.4.53 PCI Express Capabilities Register (Capabilities_Register)....1559
19.4.4.53.1 Offset....1559
19.4.4.53.2 Diagram....1559
19.4.4.53.3 Fields....1559
19.4.4.54 PCI Express Device Capabilities Register (Device_Capabilities_Register)....1560
19.4.4.54.1 Offset....1560
19.4.4.54.2 Diagram....1560
19.4.4.54.3 Fields....1561
19.4.4.55 PCI Express Device Control Register (Device_Control_Register)....1561
19.4.4.55.1 Offset....1561
19.4.4.55.2 Diagram....1562
19.4.4.55.3 Fields....1562
19.4.4.56 PCI Express Device Status Register (Device_Status_Register)....1563
19.4.4.56.1 Offset....1563
19.4.4.56.2 Diagram....1563
19.4.4.56.3 Fields....1563
19.4.4.57 PCI Express Link Capabilities Register (Link_Capabilities_Register)....1564
19.4.4.57.1 Offset....1564
19.4.4.57.2 Diagram....1564
19.4.4.57.3 Fields....1564
19.4.4.58 PCI Express Link Control Register (Link_Control_Register)....1565
19.4.4.58.1 Offset....1565
19.4.4.58.2 Diagram....1565

19.4.4.58.3 Fields....1566
19.4.4.59 PCI Express Link Status Register (Link_Status_Register)....1567
19.4.4.59.1 Offset....1567
19.4.4.59.2 Diagram....1567
19.4.4.59.3 Fields....1567
19.4.4.60 PCI Express Slot Capabilities Register (Slot_Capabilities_Register)....1568
19.4.4.60.1 Offset....1568
19.4.4.60.2 Function....1568
19.4.4.60.3 Diagram....1568
19.4.4.60.4 Fields....1568
19.4.4.61 PCI Express Slot Control Register (Slot_Control_Register)....1569
19.4.4.61.1 Offset....1569
19.4.4.61.2 Function....1569
19.4.4.61.3 Diagram....1570
19.4.4.61.4 Fields....1570
19.4.4.62 PCI Express Slot Status Register (Slot_Status_Register)....1571
19.4.4.62.1 Offset....1571
19.4.4.62.2 Function....1571
19.4.4.62.3 Diagram....1571
19.4.4.62.4 Fields....1571
19.4.4.63 PCI Express Root Control Register (Root_Control_Register)....1572
19.4.4.63.1 Offset....1572
19.4.4.63.2 Function....1572
19.4.4.63.3 Diagram....1572
19.4.4.63.4 Fields....1573
19.4.4.64 PCI Express Root Capabilities Register (Root_Capabilities_Register)....1573
19.4.4.64.1 Offset....1573
19.4.4.64.2 Function....1573
19.4.4.64.3 Diagram....1573

19.4.4.64.4 Fields....1574

19.4.4.65 PCI Express Root Status Register (Root_Status_Register).... 1574

19.4.4.65.1 Offset....1574

19.4.4.65.2 Function.... 1574

19.4.4.65.3 Diagram....1574

19.4.4.65.4 Fields....1574

19.4.4.66 PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)....1575

19.4.4.66.1 Offset....1575

19.4.4.66.2 Diagram....1575

19.4.4.66.3 Fields....1575

19.4.4.67 PCI Express Device Control 2 Register (Device_Control_2_Register).... 1576

19.4.4.67.1 Offset....1576

19.4.4.67.2 Diagram....1576

19.4.4.67.3 Fields....1576

19.4.4.68 PCI Express Link Capabilities 2 Register (Link_Capabilities_2_Register).... 1577

19.4.4.68.1 Offset....1577

19.4.4.68.2 Function.... 1577

19.4.4.68.3 Diagram....1577

19.4.4.68.4 Fields....1578

19.4.4.69 PCI Express Link Control 2 Register (Link_Control_2_Register).... 1578

19.4.4.69.1 Offset....1579

19.4.4.69.2 Diagram....1579

19.4.4.69.3 Fields....1579

19.4.4.70 PCI Express Link Status 2 Register (Link_Status_2_Register)....1579

19.4.4.70.1 Offset....1580

19.4.4.70.2 Diagram....1580

19.4.4.70.3 Fields....1580

19.4.4.71 PCI Express Advanced Error Reporting Capability ID Register (Advanced_Error_Reporting_Capability_ID_Register).... 1580

19.4.4.71.1 Offset....1581
19.4.4.71.2 Diagram....1581
19.4.4.71.3 Fields....1581
19.4.4.72 PCI Express Uncorrectable Error Status Register (Uncorrectable_Error_Status_Register)....1581
19.4.4.72.1 Offset....1581
19.4.4.72.2 Diagram....1581
19.4.4.72.3 Fields....1582
19.4.4.73 PCI Express Uncorrectable Error Mask Register (Uncorrectable_Error_Mask_Register)....1583
19.4.4.73.1 Offset....1583
19.4.4.73.2 Diagram....1583
19.4.4.73.3 Fields....1583
19.4.4.74 PCI Express Uncorrectable Error Severity Register (Uncorrectable_Error_Severity_Register)....1584
19.4.4.74.1 Offset....1584
19.4.4.74.2 Diagram....1584
19.4.4.74.3 Fields....1585
19.4.4.75 PCI Express Correctable Error Status Register (Correctable_Error_Status_Register)....1586
19.4.4.75.1 Offset....1586
19.4.4.75.2 Diagram....1586
19.4.4.75.3 Fields....1586
19.4.4.76 PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register)....1587
19.4.4.76.1 Offset....1587
19.4.4.76.2 Diagram....1587
19.4.4.76.3 Fields....1588
19.4.4.77 PCI Express Advanced Error Capabilities and Control Register (Advanced_Error_Capabilities_and_Control_Register)....1588
19.4.4.77.1 Offset....1588
19.4.4.77.2 Diagram....1588
19.4.4.77.3 Fields....1589

Section number	Title	Page

19.4.4.78 PCI Express Header Log Register 1 (Header_Log_Register_DWORD1)..... 1589

19.4.4.78.1 Offset.....1589

19.4.4.78.2 Function..... 1590

19.4.4.78.3 Diagram.....1590

19.4.4.78.4 Fields.....1590

19.4.4.79 PCI Express Header Log Register 2 (Header_Log_Register_DWORD2)..... 1590

19.4.4.79.1 Offset.....1590

19.4.4.79.2 Function..... 1591

19.4.4.79.3 Diagram.....1591

19.4.4.79.4 Fields.....1591

19.4.4.80 PCI Express Header Log Register 3 (Header_Log_Register_DWORD3)..... 1591

19.4.4.80.1 Offset.....1591

19.4.4.80.2 Function..... 1592

19.4.4.80.3 Diagram.....1592

19.4.4.80.4 Fields.....1592

19.4.4.81 PCI Express Header Log Register 4 (Header_Log_Register_DWORD4)..... 1592

19.4.4.81.1 Offset.....1592

19.4.4.81.2 Function..... 1593

19.4.4.81.3 Diagram.....1593

19.4.4.81.4 Fields.....1593

19.4.4.82 PCI Express Root Error Command Register (Root_Error_Command_Register).....1593

19.4.4.82.1 Offset.....1593

19.4.4.82.2 Function..... 1594

19.4.4.82.3 Diagram.....1594

19.4.4.82.4 Fields.....1594

19.4.4.83 PCI Express Root Error Status Register (Root_Error_Status_Register)..... 1594

19.4.4.83.1 Offset.....1595

19.4.4.83.2 Function..... 1595

19.4.4.83.3 Diagram.....1595

19.4.4.83.4 Fields....1595

19.4.4.84 PCI Express Correctable Error Source ID Register (Correctable_Error_Source_ID_Register)....1596

19.4.4.84.1 Offset....1596

19.4.4.84.2 Diagram....1596

19.4.4.84.3 Fields....1596

19.4.4.85 PCI Express Error Source ID Register (Error_Source_ID_Register)....1597

19.4.4.85.1 Offset....1597

19.4.4.85.2 Diagram....1597

19.4.4.85.3 Fields....1597

19.4.4.86 Secondary PCI Express Extended Capability Header (SPCIE_CAP_HEADER_REG)....1597

19.4.4.86.1 Offset....1597

19.4.4.86.2 Diagram....1597

19.4.4.86.3 Fields....1598

19.4.4.87 Link Control 3 Register (LINK_CONTROL3_REG)....1598

19.4.4.87.1 Offset....1598

19.4.4.87.2 Diagram....1598

19.4.4.87.3 Fields....1599

19.4.4.88 Lane Error Status Register (LANE_ERR_STATUS_REG)....1599

19.4.4.88.1 Offset....1599

19.4.4.88.2 Diagram....1599

19.4.4.88.3 Fields....1600

19.4.4.89 Lane Equalization Control Register (LANE0_EQUALIZATION_CONTROL - LANE1_EQUALIZATION_CONTROL)....1600

19.4.4.89.1 Offset....1601

19.4.4.89.2 Diagram....1602

19.4.4.89.3 Fields....1602

19.4.4.90 Symbol Timer Register and Filter Mask 1 Register (SYMBOL_TIMER_FILTER_1_OFF)....1602

19.4.4.90.1 Offset....1602

19.4.4.90.2 Function.... 1603
19.4.4.90.3 Diagram....1603
19.4.4.90.4 Fields....1603
19.4.4.91 Gen3 Control Register (GEN3_RELATED_OFF).... 1605
19.4.4.91.1 Offset....1605
19.4.4.91.2 Diagram....1605
19.4.4.91.3 Fields....1606
19.4.4.92 DBI Read-only Write Enable Register (MISC_CONTROL_1_OFF).... 1607
19.4.4.92.1 Offset....1607
19.4.4.92.2 Diagram....1607
19.4.4.92.3 Fields....1607
19.4.4.93 Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF).... 1608
19.4.4.93.1 Offset....1608
19.4.4.93.2 Diagram....1608
19.4.4.93.3 Fields....1608
19.4.4.94 Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF).... 1609
19.4.4.94.1 Offset....1609
19.4.4.94.2 Diagram....1609
19.4.4.94.3 Fields....1609
19.4.4.95 Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF).... 1610
19.4.4.95.1 Offset....1610
19.4.4.95.2 Diagram....1610
19.4.4.95.3 Fields....1610
19.4.4.96 iATU Index Register (IATU_VIEWPORT_OFF)....1610
19.4.4.96.1 Offset....1611
19.4.4.96.2 Function.... 1611
19.4.4.96.3 Diagram....1611
19.4.4.96.4 Fields....1611
19.4.4.97 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_INBOUND_0).... 1612

19.4.4.97.1 Offset....1612
19.4.4.97.2 Diagram....1612
19.4.4.97.3 Fields....1612
19.4.4.98 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)....1613
19.4.4.98.1 Offset....1613
19.4.4.98.2 Diagram....1613
19.4.4.98.3 Fields....1614
19.4.4.99 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)....1614
19.4.4.99.1 Offset....1615
19.4.4.99.2 Diagram....1615
19.4.4.99.3 Fields....1615
19.4.4.100 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)....1617
19.4.4.100.1 Offset....1617
19.4.4.100.2 Diagram....1617
19.4.4.100.3 Fields....1618
19.4.4.101 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)....1619
19.4.4.101.1 Offset....1619
19.4.4.101.2 Diagram....1619
19.4.4.101.3 Fields....1619
19.4.4.102 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0) 1619
19.4.4.102.1 Offset....1619
19.4.4.102.2 Diagram....1620
19.4.4.102.3 Fields....1620
19.4.4.103 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0). 1620
19.4.4.103.1 Offset....1620
19.4.4.103.2 Diagram....1621
19.4.4.103.3 Fields....1621
19.4.4.104 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)....1621

19.4.4.104.1 Offset....1621
19.4.4.104.2 Diagram....1621
19.4.4.104.3 Fields....1622
19.4.4.105 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_INBOUND_0)....1622
19.4.4.105.1 Offset....1622
19.4.4.105.2 Diagram....1622
19.4.4.105.3 Fields....1623
19.4.4.106 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)....1623
19.4.4.106.1 Offset....1623
19.4.4.106.2 Diagram....1623
19.4.4.106.3 Fields....1624
19.4.4.107 iATU Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)....1624
19.4.4.107.1 Offset....1624
19.4.4.107.2 Diagram....1624
19.4.4.107.3 Fields....1625
19.4.4.108 iATU Outbound Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)....1625
19.4.4.108.1 Offset....1625
19.4.4.108.2 Diagram....1626
19.4.4.108.3 Fields....1626
19.4.4.109 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0)....1626
19.4.4.109.1 Offset....1626
19.4.4.109.2 Diagram....1626
19.4.4.109.3 Fields....1627
19.4.4.110 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)....1627
19.4.4.110.1 Offset....1627
19.4.4.110.2 Diagram....1627

19.4.4.110.3 Fields....1628
19.4.4.111 Base Address Register 0 Mask (BAR0_MASK).... 1628
19.4.4.111.1 Offset....1628
19.4.4.111.2 Function.... 1628
19.4.4.111.3 Diagram....1628
19.4.4.111.4 Fields....1629
19.4.4.112 Base Address Register 1 Mask (BAR1_MASK).... 1629
19.4.4.112.1 Offset....1629
19.4.4.112.2 Function.... 1629
19.4.4.112.3 Diagram....1629
19.4.4.112.4 Fields....1630
19.4.4.113 Base Address Register 2 Mask (BAR2_MASK).... 1630
19.4.4.113.1 Offset....1630
19.4.4.113.2 Function.... 1630
19.4.4.113.3 Diagram....1631
19.4.4.113.4 Fields....1631
19.4.4.114 Base Address Register 3 Mask (BAR3_MASK).... 1631
19.4.4.114.1 Offset....1631
19.4.4.114.2 Function.... 1631
19.4.4.114.3 Diagram....1632
19.4.4.114.4 Fields....1632
19.4.4.115 Base Address Register 4 Mask (BAR4_MASK).... 1632
19.4.4.115.1 Offset....1632
19.4.4.115.2 Function.... 1633
19.4.4.115.3 Diagram....1633
19.4.4.115.4 Fields....1633
19.4.4.116 Base Address Register 5 Mask (BAR5_MASK).... 1633
19.4.4.116.1 Offset....1633
19.4.4.116.2 Function.... 1634

19.4.4.116.3 Diagram....1634
19.4.4.116.4 Fields....1634
19.4.4.117 Expansion ROM Base Address Register Mask (EP mode) (EXP_ROM_BAR_MASK_EP).1634
19.4.4.117.1 Offset....1635
19.4.4.117.2 Function....1635
19.4.4.117.3 Diagram....1635
19.4.4.117.4 Fields....1635
19.4.4.118 Expansion ROM Base Address Register Mask (RC mode) (EXP_ROM_BAR_MASK_RC)....1636
19.4.4.118.1 Offset....1636
19.4.4.118.2 Function....1636
19.4.4.118.3 Diagram....1636
19.4.4.118.4 Fields....1637
19.5 PEX_LUT memory map/registers overview....1637
19.5.1 SR-IOV PEX Look Up Table (PEX1_LUT) register descriptions....1637
19.5.1.1 PEX_LUT memory map....1637
19.5.1.2 PEX LUT Status Register (PEXLSR)....1640
19.5.1.2.1 Offset....1640
19.5.1.2.2 Function....1640
19.5.1.2.3 Diagram....1640
19.5.1.2.4 Fields....1641
19.5.1.3 PEX LUT Control Register (PEXLCR)....1641
19.5.1.3.1 Offset....1641
19.5.1.3.2 Function....1641
19.5.1.3.3 Diagram....1642
19.5.1.3.4 Fields....1642
19.5.1.4 PEX LUT Entry a Upper Data Register (PEXL0UDR)....1642
19.5.1.4.1 Offset....1642
19.5.1.4.2 Function....1643

19.5.1.4.3 Diagram....1643
19.5.1.4.4 Fields....1643
19.5.1.5 PEX LUT Entry a Lower Data Register (PEXL0LDR)....1643
19.5.1.5.1 Offset....1643
19.5.1.5.2 Function....1643
19.5.1.5.3 Diagram....1644
19.5.1.5.4 Fields....1644
19.5.1.6 PEX LUT Entry a Upper Data Register (PEXL1UDR - PEXL31UDR)....1644
19.5.1.6.1 Offset....1644
19.5.1.6.2 Function....1645
19.5.1.6.3 Diagram....1645
19.5.1.6.4 Fields....1645
19.5.1.7 PEX LUT Entry a Lower Data Register (PEXL1LDR - PEXL31LDR)....1645
19.5.1.7.1 Offset....1645
19.5.1.7.2 Function....1646
19.5.1.7.3 Diagram....1646
19.5.1.7.4 Fields....1646
19.5.1.8 PEX PFa Config Register (PEX_PF0_CONFIG - PEX_PF1_CONFIG)....1647
19.5.1.8.1 Offset....1647
19.5.1.8.2 Function....1647
19.5.1.8.3 Diagram....1647
19.5.1.8.4 Fields....1647
19.5.1.9 PEX PFa Interrupt status Register (PEX_PF0_INT_STAT - PEX_PF1_INT_STAT)....1648
19.5.1.9.1 Offset....1648
19.5.1.9.2 Function....1648
19.5.1.9.3 Diagram....1648
19.5.1.9.4 Fields....1648
19.5.1.10 PEX PFa PCIE pme and message detect register (PEX_PF0_PME_MES_DR - PEX_PF1_PME_MES_DR)....1649

19.5.1.10.1 Offset....1649
19.5.1.10.2 Function.... 1649
19.5.1.10.3 Diagram....1649
19.5.1.10.4 Fields....1649
19.5.1.11 PEX PFa PCIE pme and message disable register (PEX_PF0_PME_MES_DISR - PEX_PF1_PME_MES_DISR)....1650
19.5.1.11.1 Offset....1650
19.5.1.11.2 Function.... 1650
19.5.1.11.3 Diagram....1650
19.5.1.11.4 Fields....1651
19.5.1.12 PEX PFa PCIE pme and message interrupt enable register (PEX_PF0_PME_MES_IER - PEX_PF1_PME_MES_IER).... 1651
19.5.1.12.1 Offset....1652
19.5.1.12.2 Function.... 1652
19.5.1.12.3 Diagram....1652
19.5.1.12.4 Fields....1652
19.5.1.13 PEX PFa PCIE message command register (PEX_PF0_MCR - PEX_PF1_MCR).... 1653
19.5.1.13.1 Offset....1653
19.5.1.13.2 Function.... 1653
19.5.1.13.3 Diagram....1653
19.5.1.13.4 Fields....1654
19.5.1.14 PEX PFa Route By Port Address Upper register (PEX_PF0_RBP_ADDR_U - PEX_PF1_RBP_ADDR_U)....1654
19.5.1.14.1 Offset....1654
19.5.1.14.2 Function.... 1655
19.5.1.14.3 Diagram....1655
19.5.1.14.4 Fields....1655
19.5.1.15 PEX PFa PCIE error detect register (PEX_PF0_ERR_DR - PEX_PF1_ERR_DR).... 1655
19.5.1.15.1 Offset....1655
19.5.1.15.2 Function.... 1656

19.5.1.15.3 Diagram....1656
19.5.1.15.4 Fields....1656
19.5.1.16 PEX PFa PCIE error interrupt enable register (PEX_PF0_ERR_EN - PEX_PF1_ERR_EN) 1657
19.5.1.16.1 Offset....1657
19.5.1.16.2 Function.... 1657
19.5.1.16.3 Diagram....1657
19.5.1.16.4 Fields....1658
19.5.1.17 PEX PFa PCIE error disable register (PEX_PF0_ERR_DISR - PEX_PF1_ERR_DISR)..... 1658
19.5.1.17.1 Offset....1659
19.5.1.17.2 Function.... 1659
19.5.1.17.3 Diagram....1659
19.5.1.17.4 Fields....1659
19.5.1.18 PEX PFa VF control register (PEX_PF0_VF - PEX_PF1_VF).... 1660
19.5.1.18.1 Offset....1660
19.5.1.18.2 Function.... 1660
19.5.1.18.3 Diagram....1660
19.5.1.18.4 Fields....1661
19.5.1.19 PEX PF0 Debug register (PEX_PF0_DBG).... 1661
19.5.1.19.1 Offset....1661
19.5.1.19.2 Function.... 1661
19.5.1.19.3 Diagram....1662
19.5.1.19.4 Fields....1662
19.5.2 Non-SR-IOV PEX Look Up Table (PEX2_LUT and PEX3_LUT) register descriptions....1663
19.5.2.1 PEX_LUT memory map.... 1663
19.5.2.2 PEX LUT Status Register (PEXLSR)....1666
19.5.2.2.1 Offset....1666
19.5.2.2.2 Function.... 1666
19.5.2.2.3 Diagram....1666
19.5.2.2.4 Fields....1666

Section number	Title	Page

19.5.2.3 PEX LUT Control Register (PEXLCR) 1667

19.5.2.3.1 Offset 1667

19.5.2.3.2 Function 1667

19.5.2.3.3 Diagram 1667

19.5.2.3.4 Fields 1667

19.5.2.4 PEX LUT Entry a Upper Data Register (PEXL0UDR) 1668

19.5.2.4.1 Offset 1668

19.5.2.4.2 Function 1668

19.5.2.4.3 Diagram 1668

19.5.2.4.4 Fields 1668

19.5.2.5 PEX LUT Entry a Lower Data Register (PEXL0LDR) 1669

19.5.2.5.1 Offset 1669

19.5.2.5.2 Function 1669

19.5.2.5.3 Diagram 1669

19.5.2.5.4 Fields 1669

19.5.2.6 PEX LUT Entry a Upper Data Register (PEXL1UDR - PEXL31UDR) 1670

19.5.2.6.1 Offset 1670

19.5.2.6.2 Function 1670

19.5.2.6.3 Diagram 1670

19.5.2.6.4 Fields 1671

19.5.2.7 PEX LUT Entry a Lower Data Register (PEXL1LDR - PEXL31LDR) 1671

19.5.2.7.1 Offset 1671

19.5.2.7.2 Function 1671

19.5.2.7.3 Diagram 1671

19.5.2.7.4 Fields 1672

19.5.2.8 PEX PFa Config Register (PEX_PF0_CONFIG) 1672

19.5.2.8.1 Offset 1672

19.5.2.8.2 Function 1672

19.5.2.8.3 Diagram 1673

19.5.2.8.4 Fields....1673
19.5.2.9 PEX PFa Interrupt status Register (PEX_PF0_INT_STAT).... 1673
19.5.2.9.1 Offset....1673
19.5.2.9.2 Function.... 1673
19.5.2.9.3 Diagram....1674
19.5.2.9.4 Fields....1674
19.5.2.10 PEX PFa PCIE pme and message detect register (PEX_PF0_PME_MES_DR)....1674
19.5.2.10.1 Offset....1674
19.5.2.10.2 Function.... 1674
19.5.2.10.3 Diagram....1675
19.5.2.10.4 Fields....1675
19.5.2.11 PEX PFa PCIE pme and message disable register (PEX_PF0_PME_MES_DISR).... 1676
19.5.2.11.1 Offset....1676
19.5.2.11.2 Function.... 1676
19.5.2.11.3 Diagram....1676
19.5.2.11.4 Fields....1676
19.5.2.12 PEX PFa PCIE pme and message interrupt enable register (PEX_PF0_PME_MES_IER).... 1677
19.5.2.12.1 Offset....1677
19.5.2.12.2 Function.... 1677
19.5.2.12.3 Diagram....1677
19.5.2.12.4 Fields....1678
19.5.2.13 PEX PFa PCIE message command register (PEX_PF0_MCR).... 1678
19.5.2.13.1 Offset....1679
19.5.2.13.2 Function.... 1679
19.5.2.13.3 Diagram....1679
19.5.2.13.4 Fields....1679
19.5.2.14 PEX PFa Route By Port Address Upper register (PEX_PF0_RBP_ADDR_U).... 1680
19.5.2.14.1 Offset....1680
19.5.2.14.2 Function.... 1680

19.5.2.14.3 Diagram....1680
19.5.2.14.4 Fields....1680
19.5.2.15 PEX PFa PCIE error detect register (PEX_PF0_ERR_DR)....1681
19.5.2.15.1 Offset....1681
19.5.2.15.2 Function....1681
19.5.2.15.3 Diagram....1681
19.5.2.15.4 Fields....1681
19.5.2.16 PEX PFa PCIE error interrupt enable register (PEX_PF0_ERR_EN)....1682
19.5.2.16.1 Offset....1682
19.5.2.16.2 Function....1682
19.5.2.16.3 Diagram....1682
19.5.2.16.4 Fields....1683
19.5.2.17 PEX PFa PCIE error disable register (PEX_PF0_ERR_DISR)....1683
19.5.2.17.1 Offset....1683
19.5.2.17.2 Function....1684
19.5.2.17.3 Diagram....1684
19.5.2.17.4 Fields....1684
19.5.2.18 PEX PF0 Debug register (PEX_PF0_DBG)....1685
19.5.2.18.1 Offset....1685
19.5.2.18.2 Function....1685
19.5.2.18.3 Diagram....1685
19.5.2.18.4 Fields....1685
19.6 Functional Description....1687
19.6.1 Architecture....1688
19.6.1.1 PCI Express Transactions....1688
19.6.1.2 Lane Reversal....1689
19.6.1.3 Transaction ordering rules....1689
19.6.1.4 Internal Address Translation Unit....1691
19.6.1.4.1 iATU Overview....1691

19.6.1.4.2 Programming the iATU ... 1692
19.6.1.4.3 Outbound iATU Operation ... 1693
19.6.1.4.3.1 Overview (Address Match Mode) ... 1693
19.6.1.4.3.2 RID BDF Number Replacement ... 1695
19.6.1.4.3.3 iATU Outbound MSG Handling ... 1695
19.6.1.4.3.4 CFG Handling ... 1695
19.6.1.4.3.5 CFG Shift Feature ... 1695
19.6.1.4.3.6 FMT Translation ... 1696
19.6.1.4.3.7 No Address Match Result ... 1696
19.6.1.4.3.8 Writing to a MRdLk Region ... 1696
19.6.1.4.3.9 Outbound Programming Example ... 1696
19.6.1.4.4 Inbound iATU Operation ... 1697
19.6.1.4.4.1 Overview ... 1697
19.6.1.4.4.2 MEM Match Modes ... 1698
19.6.1.4.4.3 CFG Handling (Upstream Port) ... 1700
19.6.1.4.4.4 FMT Translation ... 1701
19.6.1.4.4.5 Inbound Programming Example ... 1701
19.6.1.5 Memory Space Addressing ... 1703
19.6.1.6 I/O Space Addressing ... 1703
19.6.1.7 Configuration Space Addressing ... 1704
19.6.1.8 Messages ... 1704
19.6.1.8.1 Outbound Message Generation ... 1704
19.6.1.8.2 Inbound Messages ... 1704
19.6.1.9 Error Handling ... 1705
19.6.1.9.1 PCI Express Error Logging and Signaling ... 1706
19.6.2 Interrupts ... 1707
19.6.3 Initial Credit Advertisement ... 1708
19.6.4 Power Management ... 1708
19.6.4.1 L2/L3 Ready Link State ... 1709

19.6.5 Hot Reset....1709
19.6.6 Alternative Routing ID (ARI, SR-IOV only)....1710
19.6.7 Functional Level Reset (FLR, SR-IOV only)....1710
19.7 Initialization/Application Information....1710
19.7.1 Gen 3 Link Equalization....1710
19.7.2 Configuring the chip for inbound CCSR accesses....1710
19.7.3 Poisoned TLP handling....1711

Chapter 20
Power Management

20.1 The PMU registers as implemented on the chip....1713
20.2 Overview....1713
20.3 Arm GPP Cores and Cluster Level Power Management....1714
20.3.1 Central SoC Power Management Unit Features....1714
20.3.2 Arm GPP Core Power Management State Summary- Core and Core Cluster level....1714
20.3.3 Arm Power Management....1715
20.3.3.1 Arm Overview....1715
20.3.3.2 Arm GPP Core PM State: Entry and Exit....1716
20.3.3.3 Arm GPP Cluster PM State: Entry and Exit....1718
20.3.3.4 Steps for preparing GPP Arm core for PCL10 or LPM20....1718
20.3.3.4.1 LPM20 Operation....1719
20.3.3.4.1.1 LPM20 Entry Sequence....1719
20.3.3.4.1.2 LPM20 Exit Sequence....1720
20.3.3.4.2 STANDBYWFIL2 (PCL10) Entry and Exit Sequences....1721
20.3.3.4.2.1 PCL10 Entry Sequence....1721
20.3.3.4.2.2 PCL10 Exit Requirements for Arm GPP Clusters....1721
20.3.3.5 Memory Map and Register Definition....1722
20.3.3.5.1 PMU_Arm register descriptions....1722
20.3.3.5.1.1 PMU_Arm CCSR memory map....1722
20.3.3.5.1.2 Core Wait Status Register (TWAITSR0)....1723

Section number	Title	Page

20.3.3.5.1.3 Core Watchdog Detect Status Register (TWDTDSR0)..... 1725

20.3.3.5.1.4 Core Time Base Halt Control Register n (TTBHLTCR0)..... 1726

20.3.3.5.1.5 Physical Core WFE Status Register (PCWFESR)..... 1728

20.3.3.5.1.6 Physical Core PW20 Status Register (PCPW20SR)..... 1729

20.3.3.5.1.7 Physical Core PH30 Status Register (PCPH30SR)..... 1730

20.3.3.5.1.8 Physical Core Time Base Enable Register (PCTBENR)..... 1732

20.3.3.5.1.9 Physical Core Time Base Clock Select Register (PCTBCKSELR)..... 1733

20.3.3.5.1.10 Cluster PCL10 (Stop) Status Register (CLPCL10SR)..... 1734

20.3.3.5.1.11 Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)..... 1735

20.3.3.5.1.12 Cluster Wakeup Event Delay Register (CLWAKEEVENTDLYR)..... 1736

20.3.3.5.1.13 Cluster ACP AINACTS Set Control Register (CLAINACTSETR)..... 1738

20.3.3.5.1.14 Cluster ACP AINACTS Clear Control Register (CLAINACTCLRR)..... 1739

20.3.3.5.1.15 Cluster Skyros SINACT Set Control Register (CLSINACTSETR)..... 1740

20.3.3.5.1.16 Cluster Skyros SINACT Clear Control Register (CLSINACTCLRR)..... 1741

20.3.3.5.1.17 Cluster L2FLUSH Set Control Register (CLL2FLUSHSETR)..... 1742

20.3.3.5.1.18 Cluster L2FLUSH Clear Control Register (CLSL2FLUSHCLRR)..... 1743

20.3.3.5.1.19 Cluster L2 FLUSH Done Status Register (CLL2FLUSHSR).. 1744

20.3.3.5.1.20 Cluster Time Base Enable Register (CLTBENR)..... 1745

20.3.3.5.1.21 IP Revision 1 Register (IPREV1R)..... 1746

20.3.3.5.1.22 IP Revision 2 Register (IPREV2R)..... 1747

20.3.4 Cluster L2 Clock Gating..... 1748

20.4 Device Power Management..... 1748

Section number	Title	Page

20.4.1 Central SoC Power Management Unit Features - Device and Domain 1748

20.4.2 States of Operation..... 1749

20.4.2.1 Arm GPP Core Power Management State Summary- Core and Core Cluster level..... 1749

20.4.3 LPM20 Operation Sequence..... 1749

20.4.4 Memory Map and Register Definition..... 1751

20.4.4.1 PMU_SOC register descriptions..... 1751

20.4.4.1.1 PMU_SOC CCSR memory map..... 1751

20.4.4.1.2 Core Wait Status Register (TWAITSR0)..... 1752

20.4.4.1.2.1 Offset..... 1752

20.4.4.1.2.2 Function..... 1752

20.4.4.1.2.3 Diagram..... 1752

20.4.4.1.2.4 Fields..... 1753

20.4.4.1.3 Core Time Base Halt Control Register n (TTBHLTCR0)..... 1754

20.4.4.1.3.1 Offset..... 1754

20.4.4.1.3.2 Function..... 1754

20.4.4.1.3.3 Diagram..... 1754

20.4.4.1.3.4 Fields..... 1754

20.4.4.1.4 Physical Core PH20 Set Control Register (PCPH20SETR)..... 1755

20.4.4.1.4.1 Offset..... 1755

20.4.4.1.4.2 Function..... 1755

20.4.4.1.4.3 Diagram..... 1755

20.4.4.1.4.4 Fields..... 1756

20.4.4.1.5 Physical Core PH20 Clear Control Register (PCPH20CLRR)..... 1757

20.4.4.1.5.1 Offset..... 1757

20.4.4.1.5.2 Function..... 1757

20.4.4.1.5.3 Diagram..... 1757

20.4.4.1.5.4 Fields..... 1757

20.4.4.1.6 Physical Core PH30 Status Register (PCPH30SR)..... 1758

20.4.4.1.6.1 Offset..... 1758

20.4.4.1.6.2 Function.....1758
20.4.4.1.6.3 Diagram.....1758
20.4.4.1.6.4 Fields.....1759
20.4.4.1.7 Physical Core Time Base Enable Register (PCTBENR).....1759
20.4.4.1.7.1 Offset.....1760
20.4.4.1.7.2 Function.....1760
20.4.4.1.7.3 Diagram.....1760
20.4.4.1.7.4 Fields.....1760
20.4.4.1.8 Time Base Clock Divider Register (TBCLKDIVR).....1761
20.4.4.1.8.1 Offset.....1761
20.4.4.1.8.2 Function.....1761
20.4.4.1.8.3 Diagram.....1761
20.4.4.1.8.4 Fields.....1761
20.4.4.1.9 Cluster PCL10 (Stop) Status Register (CLPCL10SR).....1762
20.4.4.1.9.1 Offset.....1762
20.4.4.1.9.2 Function.....1762
20.4.4.1.9.3 Diagram.....1762
20.4.4.1.9.4 Fields.....1762
20.4.4.1.10 Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR).....1763
20.4.4.1.10.1 Offset.....1763
20.4.4.1.10.2 Function.....1763
20.4.4.1.10.3 Diagram.....1763
20.4.4.1.10.4 Fields.....1764
20.4.4.1.11 Power Management Control and Status Register (POWMGTCSR).....1764
20.4.4.1.11.1 Offset.....1764
20.4.4.1.11.2 Function.....1764
20.4.4.1.11.3 Diagram.....1764
20.4.4.1.11.4 Fields.....1765
20.4.4.1.12 IP Powerdown Exception Control Register 0 (IPPDEXPCR0).....1766

20.4.4.1.12.1 Offset ... 1766
20.4.4.1.12.2 Function ... 1766
20.4.4.1.12.3 Diagram ... 1767
20.4.4.1.12.4 Fields ... 1767
20.4.4.1.13 IP Powerdown Exception Control Register 1 (IPPDEXPCR1) ... 1768
20.4.4.1.13.1 Offset ... 1768
20.4.4.1.13.2 Function ... 1768
20.4.4.1.13.3 Diagram ... 1768
20.4.4.1.13.4 Fields ... 1769
20.4.4.1.14 IP Powerdown Exception Control Register 2 (IPPDEXPCR2) ... 1769
20.4.4.1.14.1 Offset ... 1769
20.4.4.1.14.2 Function ... 1769
20.4.4.1.14.3 Diagram ... 1770
20.4.4.1.14.4 Fields ... 1770
20.4.4.1.15 IP Powerdown Exception Control Register 3 (IPPDEXPCR3) ... 1771
20.4.4.1.15.1 Offset ... 1771
20.4.4.1.15.2 Function ... 1771
20.4.4.1.15.3 Diagram ... 1771
20.4.4.1.15.4 Fields ... 1772
20.4.4.1.16 IP Powerdown Exception Control Register 4 (IPPDEXPCR4) ... 1772
20.4.4.1.16.1 Offset ... 1772
20.4.4.1.16.2 Function ... 1773
20.4.4.1.16.3 Diagram ... 1773
20.4.4.1.16.4 Fields ... 1773
20.4.4.1.17 IP Powerdown Exception Control Register 5 (IPPDEXPCR5) ... 1774
20.4.4.1.17.1 Offset ... 1774
20.4.4.1.17.2 Function ... 1774
20.4.4.1.17.3 Diagram ... 1775
20.4.4.1.17.4 Fields ... 1775

Section number	Title	Page

20.5 Reset Modes....1775

20.5.1 Power-On Reset and Hard Reset State....1775

20.5.2 Power Management States....1776

20.5.2.1 PH10 State....1776

20.5.2.2 PH15 State....1776

20.5.2.3 PH20 State....1776

20.5.2.4 PH30 State....1776

20.5.2.5 PCL10 State....1777

20.6 External Signal Description....1777

20.7 Satellite DPAA Power Management Unit Features....1777

20.7.1 Power PC Overview....1779

20.7.2 DPAA2 Core Power Management State Summary- Core, Cluster, Domain and Device Levels....1779

20.7.3 Power Management States Entry and Exit Conditions....1782

20.7.4 Memory Map and Register Definition....1784

20.7.4.1 DP_PMU_PPC register descriptions....1784

20.7.4.1.1 DP_PMU_PPC CCSR memory map....1784

20.7.4.1.2 Core PH10 Status Register (TPH10SR0)....1785

20.7.4.1.2.1 Offset....1785

20.7.4.1.2.2 Function....1785

20.7.4.1.2.3 Diagram....1786

20.7.4.1.2.4 Fields....1786

20.7.4.1.3 Core PH10 Set Control Register (TPH10SETR0)....1787

20.7.4.1.3.1 Offset....1787

20.7.4.1.3.2 Function....1787

20.7.4.1.3.3 Diagram....1788

20.7.4.1.3.4 Fields....1788

20.7.4.1.4 Core PH10 Clear Control Register (TPH10CLRR)....1790

20.7.4.1.4.1 Offset....1790

20.7.4.1.4.2 Function....1790

20.7.4.1.4.3 Diagram..... 1790
20.7.4.1.4.4 Fields..... 1791
20.7.4.1.5 Core PH10 Previous Status Register (TPH10PSR0)..... 1793
20.7.4.1.5.1 Offset..... 1793
20.7.4.1.5.2 Function..... 1793
20.7.4.1.5.3 Diagram..... 1793
20.7.4.1.5.4 Fields..... 1794
20.7.4.1.6 Core Wait Status Register (TWAITSR0)..... 1796
20.7.4.1.6.1 Offset..... 1796
20.7.4.1.6.2 Function..... 1796
20.7.4.1.6.3 Diagram..... 1796
20.7.4.1.6.4 Fields..... 1797
20.7.4.1.7 Core Time Base Halt Control Register n (TTBHLTCR0)..... 1798
20.7.4.1.7.1 Offset..... 1798
20.7.4.1.7.2 Function..... 1798
20.7.4.1.7.3 Diagram..... 1798
20.7.4.1.7.4 Fields..... 1799
20.7.4.1.8 Physical Core PH15 Status Register (PCPH15SR)..... 1800
20.7.4.1.8.1 Offset..... 1800
20.7.4.1.8.2 Function..... 1800
20.7.4.1.8.3 Diagram..... 1801
20.7.4.1.8.4 Fields..... 1801
20.7.4.1.9 Physical Core PH15 Set Control Register (PCPH15SETR)..... 1802
20.7.4.1.9.1 Offset..... 1802
20.7.4.1.9.2 Function..... 1803
20.7.4.1.9.3 Diagram..... 1803
20.7.4.1.9.4 Fields..... 1803
20.7.4.1.10 Physical Core PH15 Clear Control Register (PCPH15CLRR)..... 1805
20.7.4.1.10.1 Offset..... 1805

20.7.4.1.10.2 Function ... 1805
20.7.4.1.10.3 Diagram ... 1805
20.7.4.1.10.4 Fields ... 1806
20.7.4.1.11 Physical Core PH15 Previous Status Register (PCPH15PSR) ... 1808
20.7.4.1.11.1 Offset ... 1808
20.7.4.1.11.2 Function ... 1808
20.7.4.1.11.3 Diagram ... 1808
20.7.4.1.11.4 Fields ... 1809
20.7.4.1.12 Physical Core PH20 Status Register (PCPH20SR) ... 1811
20.7.4.1.12.1 Offset ... 1811
20.7.4.1.12.2 Function ... 1811
20.7.4.1.12.3 Diagram ... 1811
20.7.4.1.12.4 Fields ... 1811
20.7.4.1.13 Physical Core PH20 Previous Status Register (PCPH20PSR) ... 1813
20.7.4.1.13.1 Offset ... 1813
20.7.4.1.13.2 Function ... 1813
20.7.4.1.13.3 Diagram ... 1813
20.7.4.1.13.4 Fields ... 1814
20.7.4.1.14 Physical Core PW20 State Retention Configuration Register (PCRETCR) ... 1815
20.7.4.1.14.1 Offset ... 1815
20.7.4.1.14.2 Function ... 1815
20.7.4.1.14.3 Diagram ... 1816
20.7.4.1.14.4 Fields ... 1816
20.7.4.1.15 Physical Core PH30 Status Register (PCPH30SR) ... 1817
20.7.4.1.15.1 Offset ... 1817
20.7.4.1.15.2 Function ... 1818
20.7.4.1.15.3 Diagram ... 1818
20.7.4.1.15.4 Fields ... 1818
20.7.4.1.16 Physical Core Time Base Enable Register (PCTBENR) ... 1819

20.7.4.1.16.1 Offset ... 1820
20.7.4.1.16.2 Function ... 1820
20.7.4.1.16.3 Diagram ... 1820
20.7.4.1.16.4 Fields ... 1820
20.7.4.1.17 Cluster PCL10 (Stop) Status Register (CLPCL10SR) ... 1822
20.7.4.1.17.1 Offset ... 1822
20.7.4.1.17.2 Function ... 1822
20.7.4.1.17.3 Diagram ... 1822
20.7.4.1.17.4 Fields ... 1822
20.7.4.1.18 Cluster PCL10 (Stop) Set Control Register (CLPCL10SETR) ... 1823
20.7.4.1.18.1 Offset ... 1823
20.7.4.1.18.2 Function ... 1823
20.7.4.1.18.3 Diagram ... 1823
20.7.4.1.18.4 Fields ... 1824
20.7.4.1.19 Cluster PCL10 (Stop) Clear Control Register (CLPCL10CLRR) ... 1824
20.7.4.1.19.1 Offset ... 1824
20.7.4.1.19.2 Function ... 1824
20.7.4.1.19.3 Diagram ... 1824
20.7.4.1.19.4 Fields ... 1825
20.7.4.1.20 Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR) ... 1826
20.7.4.1.20.1 Offset ... 1826
20.7.4.1.20.2 Function ... 1826
20.7.4.1.20.3 Diagram ... 1826
20.7.4.1.20.4 Fields ... 1826
20.7.4.1.21 Cluster PCL20 (Retain) Status Register (CLPCL20SR) ... 1827
20.7.4.1.21.1 Offset ... 1827
20.7.4.1.21.2 Function ... 1827
20.7.4.1.21.3 Diagram ... 1827
20.7.4.1.21.4 Fields ... 1828

Section number	Title	Page

20.7.4.1.22 Cluster PCL20 (Retain) Set Control Register (CLPCL20SETR)..... 1828

20.7.4.1.22.1 Offset..... 1828

20.7.4.1.22.2 Function..... 1828

20.7.4.1.22.3 Diagram..... 1828

20.7.4.1.22.4 Fields..... 1829

20.7.4.1.23 Cluster PCL20 (Retain) Clear Control Register (CLPCL20CLRR)..... 1829

20.7.4.1.23.1 Offset..... 1829

20.7.4.1.23.2 Function..... 1830

20.7.4.1.23.3 Diagram..... 1830

20.7.4.1.23.4 Fields..... 1830

20.7.4.1.24 Cluster PCL20 (Retain) Previous Status Register (CLPCL20PSR)..... 1831

20.7.4.1.24.1 Offset..... 1831

20.7.4.1.24.2 Function..... 1831

20.7.4.1.24.3 Diagram..... 1831

20.7.4.1.24.4 Fields..... 1832

20.7.4.1.25 Cluster Domain DM10 (Halt) Status Register (DMPDM10SR)..... 1832

20.7.4.1.25.1 Offset..... 1832

20.7.4.1.25.2 Function..... 1833

20.7.4.1.25.3 Diagram..... 1833

20.7.4.1.25.4 Fields..... 1833

20.7.4.1.26 Cluster Domain DM10 (Halt) Set Control Register (DMPDM10SETR)..... 1834

20.7.4.1.26.1 Offset..... 1834

20.7.4.1.26.2 Function..... 1834

20.7.4.1.26.3 Diagram..... 1834

20.7.4.1.26.4 Fields..... 1834

20.7.4.1.27 Cluster Domain DM10 (Halt) Clear Control Register (DMPDM10CLRR)..... 1835

20.7.4.1.27.1 Offset..... 1835

20.7.4.1.27.2 Function..... 1835

20.7.4.1.27.3 Diagram..... 1835

Section number	Title	Page

20.7.4.1.27.4 Fields ... 1836

20.7.4.1.28 Cluster Domain DM10 (Halt) Previous Status Register (DMPDM10PSR) ... 1837

20.7.4.1.28.1 Offset ... 1837

20.7.4.1.28.2 Function ... 1837

20.7.4.1.28.3 Diagram ... 1837

20.7.4.1.28.4 Fields ... 1837

20.7.4.1.29 Cluster Domain DM15 (Stop) Status Register (DMPDM15SR) ... 1838

20.7.4.1.29.1 Offset ... 1838

20.7.4.1.29.2 Function ... 1838

20.7.4.1.29.3 Diagram ... 1838

20.7.4.1.29.4 Fields ... 1839

20.7.4.1.30 Cluster Domain DM15 (Stop) Set Control Register (DMPDM15SETR) ... 1839

20.7.4.1.30.1 Offset ... 1839

20.7.4.1.30.2 Function ... 1839

20.7.4.1.30.3 Diagram ... 1839

20.7.4.1.30.4 Fields ... 1840

20.7.4.1.31 Cluster Domain DM15 (Stop) Clear Control Register (DMPDM15CLRR) ... 1840

20.7.4.1.31.1 Offset ... 1841

20.7.4.1.31.2 Function ... 1841

20.7.4.1.31.3 Diagram ... 1841

20.7.4.1.31.4 Fields ... 1841

20.7.4.1.32 Cluster Domain DM15 (Stop) Previous Status Register (DMPDM15PSR) ... 1842

20.7.4.1.32.1 Offset ... 1842

20.7.4.1.32.2 Function ... 1842

20.7.4.1.32.3 Diagram ... 1842

20.7.4.1.32.4 Fields ... 1843

20.7.4.1.33 Cluster Domain DM20 (Retain) Status Register (DMPDM20SR) ... 1843

20.7.4.1.33.1 Offset ... 1843

20.7.4.1.33.2 Function ... 1844

20.7.4.1.33.3 Diagram..... 1844
20.7.4.1.33.4 Fields..... 1844
20.7.4.1.34 Cluster Domain DM20 (Retain) Set Control Register (DMPDM20SETR)..... 1845
20.7.4.1.34.1 Offset..... 1845
20.7.4.1.34.2 Function..... 1845
20.7.4.1.34.3 Diagram..... 1845
20.7.4.1.34.4 Fields..... 1845
20.7.4.1.35 Cluster Domain DM20 (Retain) Clear Control Register (DMPDM20CLRR). 1846
20.7.4.1.35.1 Offset..... 1846
20.7.4.1.35.2 Function..... 1846
20.7.4.1.35.3 Diagram..... 1846
20.7.4.1.35.4 Fields..... 1847
20.7.4.1.36 Cluster Domain DM20 (Retain) Previous Status Register (DMPDM20PSR)..1847
20.7.4.1.36.1 Offset..... 1848
20.7.4.1.36.2 Function..... 1848
20.7.4.1.36.3 Diagram..... 1848
20.7.4.1.36.4 Fields..... 1848
20.7.4.1.37 Cluster Domain Warm Reset Control Register (RSTCR)..... 1849
20.7.4.1.37.1 Offset..... 1849
20.7.4.1.37.2 Function..... 1849
20.7.4.1.37.3 Diagram..... 1849
20.7.4.1.37.4 Fields..... 1850
20.7.4.1.38 Cluster Domain Warm Reset Disable Control Register (RSTEXPCR).....1850
20.7.4.1.38.1 Offset..... 1850
20.7.4.1.38.2 Function..... 1850
20.7.4.1.38.3 Diagram..... 1850
20.7.4.1.38.4 Fields..... 1850
20.8 Initialization Information..... 1851
20.8.1 Low Power Modes Registers Initialization..... 1851

Section number	Title	Page

Chapter 21
Quad Serial Peripheral Interface (QuadSPI)

21.1 QuadSPI as implemented on the chip 1853

21.2 Introduction 1855

21.2.1 Features 1855

21.2.2 Block Diagram 1856

21.2.3 QuadSPI Modes of Operation 1857

21.2.4 Acronyms and Abbreviations 1857

21.2.5 Glossary for QuadSPI module 1858

21.3 External Signal Description 1859

21.3.1 Driving External Signals 1860

21.4 Memory Map and Register Definition 1862

21.4.1 Register Write Access 1862

21.4.2 Peripheral Bus Register Descriptions 1863

21.4.2.1 Module Configuration Register (QuadSPI_MCR) 1869

21.4.2.2 IP Configuration Register (QuadSPI_IPCR) 1872

21.4.2.3 Flash Configuration Register (QuadSPI_FLSHCR) 1873

21.4.2.4 Buffer0 Configuration Register (QuadSPI_BUF0CR) 1874

21.4.2.5 Buffer1 Configuration Register (QuadSPI_BUF1CR) 1875

21.4.2.6 Buffer2 Configuration Register (QuadSPI_BUF2CR) 1876

21.4.2.7 Buffer3 Configuration Register (QuadSPI_BUF3CR) 1877

21.4.2.8 Buffer Generic Configuration Register (QuadSPI_BFGENCR) 1878

21.4.2.9 SOC Configuration Register (QuadSPI_SOCCR) 1879

21.4.2.10 Buffer0 Top Index Register (QuadSPI_BUF0IND) 1879

21.4.2.11 Buffer1 Top Index Register (QuadSPI_BUF1IND) 1880

21.4.2.12 Buffer2 Top Index Register (QuadSPI_BUF2IND) 1881

21.4.2.13 Serial Flash Address Register (QuadSPI_SFAR) 1881

21.4.2.14 Sampling Register (QuadSPI_SMPR) 1882

21.4.2.15 RX Buffer Status Register (QuadSPI_RBSR) 1884

21.4.2.16 RX Buffer Control Register (QuadSPI_RBCT)....1884
21.4.2.17 TX Buffer Status Register (QuadSPI_TBSR)....1886
21.4.2.18 TX Buffer Data Register (QuadSPI_TBDR).... 1886
21.4.2.19 Status Register (QuadSPI_SR)....1888
21.4.2.20 Flag Register (QuadSPI_FR).... 1891
21.4.2.21 Interrupt and DMA Request Select and Enable Register (QuadSPI_RSER)....1894
21.4.2.22 Sequence Suspend Status Register (QuadSPI_SPNDST)....1898
21.4.2.23 Sequence Pointer Clear Register (QuadSPI_SPTRCLR).... 1900
21.4.2.24 Serial Flash A1 Top Address (QuadSPI_SFA1AD).... 1901
21.4.2.25 Serial Flash A2 Top Address (QuadSPI_SFA2AD).... 1901
21.4.2.26 Serial Flash B1 Top Address (QuadSPI_SFB1AD).... 1902
21.4.2.27 Serial Flash B2 Top Address (QuadSPI_SFB2AD).... 1902
21.4.2.28 Data Learn Pattern Register (QuadSPI_DLPR)....1903
21.4.2.29 RX Buffer Data Register (QuadSPI_RBDR*n*)....1903
21.4.2.30 LUT Key Register (QuadSPI_LUTKEY)....1904
21.4.2.31 LUT Lock Configuration Register (QuadSPI_LCKCR).... 1905
21.4.2.32 Look-up Table register (QuadSPI_LUT*n*)....1906
21.4.3 Serial Flash Address Assignment.... 1907
21.5 Flash memory mapped AMBA bus.... 1908
21.5.1 AHB Bus Access Considerations....1909
21.5.2 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A....1910
21.5.3 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B.... 1910
21.5.4 Parallel Flash Mode.... 1911
21.5.5 AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31).... 1912
21.5.5.1 AHB RX Data Buffer register (ARDB*n*)....1913
21.6 Interrupt Signals....1915
21.7 Functional Description....1916
21.7.1 Serial Flash Access Schemes.... 1916
21.7.2 Normal Mode.... 1917

21.7.2.1 Programmable Sequence Engine.... 1917
21.7.2.2 Flexible AHB buffers.... 1919
21.7.2.3 Suspend-Abort Mechanism.... 1921
21.7.2.4 Look-up Table.... 1921
21.7.2.5 Issuing Serial Flash Memory (SFM) Commands.... 1923
21.7.2.6 Flash Programming.... 1924
21.7.2.7 Flash Read.... 1925
21.7.2.8 Byte Ordering of Serial Flash Read Data.... 1929
21.7.2.9 Normal Mode Interrupt and DMA Requests.... 1932
21.7.2.10 TX Buffer Operation.... 1934
21.7.2.11 Address scheme.... 1934
21.8 Initialization/Application Information.... 1935
21.8.1 Power Up and Reset.... 1935
21.8.2 Available Status/Flag Information.... 1936
21.8.2.1 IP Commands.... 1936
21.8.2.2 AHB Commands.... 1936
21.8.2.3 Overview of Error Flags.... 1937
21.8.2.4 IP Bus and AHB Access Command Collisions.... 1938
21.8.3 Exclusive Access to Serial Flash for AHB Commands.... 1938
21.8.3.1 RX Buffer Read via QSPI_ARDB Registers.... 1939
21.8.3.2 RX Buffer Read via QSPI_RBDR Registers.... 1939
21.8.4 Command Arbitration.... 1939
21.8.5 Flash Device Selection.... 1940
21.8.6 DMA Usage.... 1941
21.8.6.1 DMA Usage in Normal Mode.... 1941
21.8.6.1.1 Bandwidth considerations.... 1941
21.8.7 Parallel mode.... 1943
21.9 Byte Ordering - Endianness.... 1945
21.9.1 Programming Flash Data.... 1946

21.9.2 Reading Flash Data into the RX Buffer.....1947
21.9.2.1 Readout of the RX Buffer via QSPI_RBDRn.....1947
21.9.2.2 Readout of the RX Buffer via ARDBn.....1947
21.9.3 Reading Flash Data into the AHB Buffer.....1948
21.9.3.1 Readout of the AHB Buffer via Memory Mapped Read.....1948
21.10 Driving Flash Control Signals in Single and Dual Mode.....1948
21.11 Serial Flash Devices.....1949
21.11.1 Example Sequences.....1949
21.11.1.1 Read Command (Spansion Hyperflash).....1949
21.11.1.2 Read Status Register(Spansion Hyperflash).....1950
21.11.1.3 Word Program (Spansion Hyperflash).....1950
21.11.1.4 Fast Read Sequence (Macronix/Numonyx/Spansion/Winbond).....1952
21.11.1.5 Fast Dual I/O DT Read Sequence (Macronix).....1952
21.11.1.6 Fast Read Quad Output (Winbond).....1953
21.11.1.7 4 x I/O Read Enhance Performance Mode (XIP) (Macronix).....1953
21.11.1.8 Dual Command Page Program (Numonyx).....1953
21.11.1.9 Sector Erase (Macronix/Spansion/Numonyx).....1954
21.11.1.10 Read Status Register (Macronix/Spansion/Numonyx/Winbond).....1954
21.11.2 Dual Die Flashes.....1954
21.11.3 Boot initialization sequence.....1955
21.11.4 Serial Flash Clock Frequency Limitations.....1956
21.12 Sampling of Serial Flash Input Data.....1956
21.12.1 Basic Description.....1957
21.12.2 Supported read modes.....1958
21.12.2.1 SDR mode.....1958
21.12.2.1.1 Internal sampling.....1958
21.12.2.1.2 DQS sampling method.....1960
21.12.2.2 DDR Mode.....1960
21.12.2.2.1 Internal sampling (4x sampling method).....1960

21.12.2.2.2 DQS sampling method....1961

21.12.3 Data Strobe (DQS) sampling method....1961

21.12.3.1 Basic Description....1961

21.12.3.2 Internally generated DQS....1963

21.12.3.3 External DQS....1963

21.12.4 Data Learning....1964

21.12.4.1 Basic description....1964

21.12.4.2 4x sampling method....1964

21.12.4.3 Data Strobe (DQS) sampling method....1966

21.12.4.4 Example: programming the data learning pattern in flash memory....1968

21.13 Data Input Hold Requirement of Flash....1968

Chapter 22
SATA 3.0

22.1 The SATA module as implemented on the chip....1971

22.2 Advanced host controller interface overview....1971

22.3 SATA features summary....1972

22.4 SATA AHCI register descriptions....1972

22.4.1 SATA memory map....1973

22.4.2 HBA capabilities register (CAP)....1974

22.4.2.1 Offset....1974

22.4.2.2 Function....1974

22.4.2.3 Diagram....1974

22.4.2.4 Fields....1975

22.4.3 Global HBA control register (GHC)....1977

22.4.3.1 Offset....1977

22.4.3.2 Function....1977

22.4.3.3 Diagram....1977

22.4.3.4 Fields....1978

22.4.4 AHCI version register (VS)....1979

22.4.4.1 Offset....1979
22.4.4.2 Function....1979
22.4.4.3 Diagram....1979
22.4.4.4 Fields....1979
22.4.5 Command completion coalescing control register (CCC_CTL)....1980
22.4.5.1 Offset....1980
22.4.5.2 Function....1980
22.4.5.3 Diagram....1980
22.4.5.4 Fields....1980
22.4.6 HBA capabilities extended register (CAP2)....1981
22.4.6.1 Offset....1981
22.4.6.2 Function....1981
22.4.6.3 Diagram....1982
22.4.6.4 Fields....1982
22.4.7 Port config register (PCFG)....1982
22.4.7.1 Offset....1982
22.4.7.2 Function....1983
22.4.7.3 Diagram....1983
22.4.7.4 Fields....1983
22.4.8 Port Phy1Cfg register (PPCFG)....1984
22.4.8.1 Offset....1984
22.4.8.2 Function....1984
22.4.8.3 Diagram....1984
22.4.8.4 Fields....1984
22.4.9 Port Phy2Cfg register (PP2C)....1986
22.4.9.1 Offset....1986
22.4.9.2 Function....1986
22.4.9.3 Diagram....1986
22.4.9.4 Fields....1987

Section number	Title	Page

22.4.10 Port Phy3Cfg register (PP3C) 1987

22.4.10.1 Offset 1987

22.4.10.2 Function 1987

22.4.10.3 Diagram 1988

22.4.10.4 Fields 1988

22.4.11 Port Phy4Cfg register (PP4C) 1989

22.4.11.1 Offset 1989

22.4.11.2 Function 1989

22.4.11.3 Diagram 1989

22.4.11.4 Fields 1990

22.4.12 Port Phy5Cfg register (PP5C) 1990

22.4.12.1 Offset 1990

22.4.12.2 Function 1990

22.4.12.3 Diagram 1991

22.4.12.4 Fields 1991

22.4.13 AXI cache control register (AXICC) 1991

22.4.13.1 Offset 1992

22.4.13.2 Function 1992

22.4.13.3 Diagram 1993

22.4.13.4 Fields 1993

22.4.14 Port AXICfg register (PAXIC) 1994

22.4.14.1 Offset 1994

22.4.14.2 Function 1994

22.4.14.3 Diagram 1995

22.4.14.4 Fields 1995

22.4.15 AXI PROT control register (AXIPC) 1996

22.4.15.1 Offset 1996

22.4.15.2 Function 1996

22.4.15.3 Diagram 1997

22.4.15.4 Fields.....1997
22.4.16 Port TransCfg register (PTC).....1998
22.4.16.1 Offset.....1998
22.4.16.2 Function.....1998
22.4.16.3 Diagram.....1999
22.4.16.4 Fields.....1999
22.4.17 Port LinkCfg register (PLC).....2000
22.4.17.1 Offset.....2000
22.4.17.2 Function.....2000
22.4.17.3 Diagram.....2000
22.4.17.4 Fields.....2000
22.4.18 Port LinkCfg1 register (PLC1).....2001
22.4.18.1 Offset.....2001
22.4.18.2 Function.....2002
22.4.18.3 Diagram.....2002
22.4.18.4 Fields.....2002
22.4.19 Port LinkCfg2 register (PLC2).....2003
22.4.19.1 Offset.....2003
22.4.19.2 Function.....2003
22.4.19.3 Diagram.....2003
22.4.19.4 Fields.....2003
22.4.20 Port LinkStatus1 register (PLS1).....2003
22.4.20.1 Offset.....2003
22.4.20.2 Function.....2004
22.4.20.3 Diagram.....2004
22.4.20.4 Fields.....2004
22.4.21 Port CmdConfig register (PCMDC).....2005
22.4.21.1 Offset.....2005
22.4.21.2 Function.....2005

22.4.21.3 Diagram....2005
22.4.21.4 Fields....2005
22.4.22 Port PhyControl status register (PPCS)....2006
22.4.22.1 Offset....2006
22.4.22.2 Function....2006
22.4.22.3 Diagram....2006
22.4.22.4 Fields....2007
22.4.23 Timer control register (TCR)....2007
22.4.23.1 Offset....2007
22.4.23.2 Function....2008
22.4.23.3 Diagram....2008
22.4.23.4 Fields....2008
22.4.24 Port x command list base address register (PxCLB)....2008
22.4.24.1 Offset....2008
22.4.24.2 Function....2009
22.4.24.3 Diagram....2009
22.4.24.4 Fields....2009
22.4.25 Port x command list base address upper 32-bit register (PxCLBU)....2009
22.4.25.1 Offset....2009
22.4.25.2 Function....2010
22.4.25.3 Diagram....2010
22.4.25.4 Fields....2010
22.4.26 Port x FIS base address register (PxFB)....2010
22.4.26.1 Offset....2010
22.4.26.2 Function....2011
22.4.26.3 Diagram....2011
22.4.26.4 Fields....2011
22.4.27 Port x FIS base address upper 32-bit register (PxFBU)....2011
22.4.27.1 Offset....2012

22.4.27.2 Function 2012
22.4.27.3 Diagram 2012
22.4.27.4 Fields 2012
22.4.28 Port x interrupt status register (PxIS) 2012
22.4.28.1 Offset 2012
22.4.28.2 Function 2013
22.4.28.3 Diagram 2013
22.4.28.4 Fields 2013
22.4.29 Port x command and status register (PxCMD) 2015
22.4.29.1 Offset 2015
22.4.29.2 Function 2015
22.4.29.3 Diagram 2015
22.4.29.4 Fields 2016
22.4.30 Port x SATA status register (PxSSTS) 2019
22.4.30.1 Offset 2019
22.4.30.2 Function 2019
22.4.30.3 Diagram 2019
22.4.30.4 Fields 2019
22.4.31 Port x SATA control register (PxSCTL) 2020
22.4.31.1 Offset 2020
22.4.31.2 Function 2020
22.4.31.3 Diagram 2020
22.4.31.4 Fields 2021
22.4.32 Port x SATA error register (PxSERR) 2022
22.4.32.1 Offset 2022
22.4.32.2 Function 2022
22.4.32.3 Diagram 2022
22.4.32.4 Fields 2022
22.4.33 Port x command issue register (PxCI) 2024

22.4.33.1 Offset....2024
22.4.33.2 Function....2025
22.4.33.3 Diagram....2025
22.4.33.4 Fields....2025
22.4.34 Port x FIS-based switching control register (PxFBS)....2025
22.4.34.1 Offset....2025
22.4.34.2 Function....2026
22.4.34.3 Diagram....2026
22.4.34.4 Fields....2026
22.4.35 Port 0 BIST error register (PBERR)....2027
22.4.35.1 Offset....2027
22.4.35.2 Function....2027
22.4.35.3 Diagram....2027
22.4.35.4 Fields....2028
22.5 Command layer....2028
22.5.1 Local port context management....2028
22.5.2 Vendor-specific BIST operation....2029
22.5.2.1 Command list structure....2029
22.5.2.2 Vendor BIST command descriptor command FIS....2030
22.5.2.3 Receive FIS area reg D2H RFIS structure....2031
22.6 PhyControl BIST modes....2031
22.7 PHY control configuration register OOB timing setup....2033
22.8 Modifications to the AHCI standard PRDT entry....2034
22.9 Transport layer architectural overview....2035
22.10 Link layer overview....2036
22.10.1 General operation....2036
22.10.2 List of functions....2037
22.10.3 Link layer state machines....2037
22.10.3.1 Link idle state machine....2037

22.10.3.2 Transmit state machine.... 2038
22.10.3.3 Receive state machine.... 2038
22.10.3.4 Power mode change state machine.... 2039
22.10.4 Frame content scrambler and descrambler.... 2040
22.10.5 CRC generator and checker.... 2041
22.10.6 8B/10B encode and decode.... 2041
22.10.7 CONT primitive processing.... 2041
22.10.8 ALIGN insertion.... 2042
22.10.9 Debug functionality.... 2042
22.10.10 BIST support.... 2043
22.11 PHY control layer overview.... 2044
22.12 Reset.... 2044
22.12.1 Software reset.... 2045
22.12.2 Port reset.... 2048
22.12.3 HBA reset.... 2048

Chapter 23
SerDes Module

23.1 SerDes Module as Implemented on LS1088A.... 2051
23.1.1 SerDes 1 and SerDes 2 Protocol Notation.... 2051
23.1.2 SerDes options.... 2052
23.1.3 Networking SerDes Protocol Lane Configuration.... 2054
23.1.3.1 Configuration of Networking SerDes.... 2054
23.1.3.2 Rules for disabling a networking SerDes module.... 2054
23.1.3.3 Rules for Selecting Valid Networking SerDes Options.... 2055
23.1.3.4 Rules for using PCIe1 on SerDes1.... 2055
23.1.3.5 RGMII Interfaces.... 2055
23.1.3.6 Rules for disabling networking SerDes PLLs when not all lanes are needed.... 2056
23.1.4 Non Networking SerDes Protocol Lane Configuration.... 2057
23.1.4.1 Configuration of Non-Networking SerDes.... 2057

23.1.4.2 Rules for Selecting Valid Non Networking SerDes Options....2057
23.1.5 Disabling Unused SerDes Modules....2058
23.1.6 Powering Down Unused SerDes Lanes....2058
23.1.7 Valid Reference Clocks and PLL Configurations for SerDes Protocols....2059
23.1.8 Ethernet MAC Capabilities....2059
23.1.9 Protocol Converter to MAC and Interface Mapping....2060
23.1.10 Non Networking Protocol to Interface Mapping....2060
23.2 Overview....2061
23.2.1 Features....2061
23.3 Modes of Operation....2061
23.4 External Signals Description....2061
23.5 SerDes register descriptions....2062
23.5.1 SerDes memory map....2063
23.5.2 SerDes PLLa Reset Control Register (PLL1RSTCTL - PLL2RSTCTL)....2066
23.5.2.1 Offset....2066
23.5.2.2 Function....2066
23.5.2.3 Diagram....2066
23.5.2.4 Fields....2066
23.5.3 SerDes PLLa Control Register 0 (PLL1CR0 - PLL2CR0)....2068
23.5.3.1 Offset....2068
23.5.3.2 Function....2068
23.5.3.3 Diagram....2068
23.5.3.4 Fields....2069
23.5.4 SerDes PLLa Control Register 1 (PLL1CR1 - PLL2CR1)....2070
23.5.4.1 Offset....2070
23.5.4.2 Function....2070
23.5.4.3 Diagram....2070
23.5.4.4 Fields....2071
23.5.5 SerDes PLLa Control Register 5 (PLL1CR5 - PLL2CR5)....2071

23.5.5.1 Offset....2071
23.5.5.2 Function....2071
23.5.5.3 Diagram....2071
23.5.5.4 Fields....2072
23.5.6 SerDes Transmit Calibration Control Register (TCALCR)....2072
23.5.6.1 Offset....2072
23.5.6.2 Function....2073
23.5.6.3 Diagram....2073
23.5.6.4 Fields....2073
23.5.7 SerDes Transmit Calibration Control Register 1 (TCALCR1)....2073
23.5.7.1 Offset....2074
23.5.7.2 Function....2074
23.5.7.3 Diagram....2074
23.5.7.4 Fields....2074
23.5.8 Receive Calibration Control Register (RCALCR)....2075
23.5.8.1 Offset....2075
23.5.8.2 Function....2075
23.5.8.3 Diagram....2075
23.5.8.4 Fields....2076
23.5.9 SerDes Receive Calibration Control Register 1 (RCALCR1)....2076
23.5.9.1 Offset....2076
23.5.9.2 Function....2076
23.5.9.3 Diagram....2076
23.5.9.4 Fields....2077
23.5.10 General Control Register 0 (GR0)....2078
23.5.10.1 Offset....2078
23.5.10.2 Function....2078
23.5.10.3 Diagram....2078
23.5.10.4 Fields....2078

Section number	Title	Page

23.5.11 Lane a Protocol Select Status Register 0 (LNAPSSR0 - LNDPSSR0)..... 2079

23.5.11.1 Offset.....2079

23.5.11.2 Function..... 2079

23.5.11.3 Diagram.....2079

23.5.11.4 Fields.....2080

23.5.12 Protocol Configuration Register 0 (PCCR0)..... 2081

23.5.12.1 Offset.....2081

23.5.12.2 Function..... 2081

23.5.12.3 Diagram.....2081

23.5.12.4 Fields.....2082

23.5.13 Protocol Configuration Register 2 (PCCR2)..... 2083

23.5.13.1 Offset.....2083

23.5.13.2 Function..... 2083

23.5.13.3 Diagram.....2083

23.5.13.4 Fields.....2084

23.5.14 Protocol Configuration Register 8 (PCCR8)..... 2085

23.5.14.1 Offset.....2085

23.5.14.2 Function..... 2085

23.5.14.3 Diagram.....2085

23.5.14.4 Fields.....2085

23.5.15 Protocol Configuration Register 9 (PCCR9)..... 2087

23.5.15.1 Offset.....2087

23.5.15.2 Function..... 2087

23.5.15.3 Diagram.....2087

23.5.15.4 Fields.....2088

23.5.16 Protocol Configuration Register B (PCCRB)..... 2089

23.5.16.1 Offset.....2089

23.5.16.2 Function..... 2089

23.5.16.3 Diagram.....2089

23.5.16.4 Fields....2089
23.5.17 PCIe Equalization Configuration Register (PEXEQCR)....2091
23.5.17.1 Offset....2091
23.5.17.2 Function....2091
23.5.17.3 Diagram....2091
23.5.17.4 Fields....2091
23.5.18 PCIe Equalization Preset 0 Register (PEXEQP0CR)....2092
23.5.18.1 Offset....2092
23.5.18.2 Function....2092
23.5.18.3 Diagram....2092
23.5.18.4 Fields....2092
23.5.19 PCIe Equalization Preset 1 Register (PEXEQP1CR)....2093
23.5.19.1 Offset....2093
23.5.19.2 Function....2093
23.5.19.3 Diagram....2094
23.5.19.4 Fields....2094
23.5.20 PCIe Equalization Preset 2 Register (PEXEQP2CR)....2095
23.5.20.1 Offset....2095
23.5.20.2 Function....2095
23.5.20.3 Diagram....2095
23.5.20.4 Fields....2095
23.5.21 PCIe Equalization Preset 3 Register (PEXEQP3CR)....2096
23.5.21.1 Offset....2096
23.5.21.2 Function....2096
23.5.21.3 Diagram....2097
23.5.21.4 Fields....2097
23.5.22 PCIe Equalization Preset 4 Register (PEXEQP4CR)....2098
23.5.22.1 Offset....2098
23.5.22.2 Function....2098

23.5.22.3 Diagram....2098
23.5.22.4 Fields....2098
23.5.23 PCIe Equalization Preset 5 Register (PEXEQP5CR)....2099
23.5.23.1 Offset....2099
23.5.23.2 Function....2099
23.5.23.3 Diagram....2100
23.5.23.4 Fields....2100
23.5.24 PCIe Equalization Preset 6 Register (PEXEQP6CR)....2101
23.5.24.1 Offset....2101
23.5.24.2 Function....2101
23.5.24.3 Diagram....2101
23.5.24.4 Fields....2101
23.5.25 PCIe Equalization Preset 7 Register (PEXEQP7CR)....2102
23.5.25.1 Offset....2102
23.5.25.2 Function....2102
23.5.25.3 Diagram....2103
23.5.25.4 Fields....2103
23.5.26 PCIe Equalization Preset 8 Register (PEXEQP8CR)....2104
23.5.26.1 Offset....2104
23.5.26.2 Function....2104
23.5.26.3 Diagram....2104
23.5.26.4 Fields....2104
23.5.27 PCIe Equalization Preset 9 Register (PEXEQP9CR)....2105
23.5.27.1 Offset....2105
23.5.27.2 Function....2105
23.5.27.3 Diagram....2106
23.5.27.4 Fields....2106
23.5.28 PCIe Equalization Preset 10 Register (PEXEQP10CR)....2107
23.5.28.1 Offset....2107

23.5.28.2 Function 2107
23.5.28.3 Diagram 2107
23.5.28.4 Fields 2107
23.5.29 General Control Register 0 - Lane a (LNAGCR0 - LNDGCR0) 2108
23.5.29.1 Offset 2108
23.5.29.2 Function 2109
23.5.29.3 Diagram 2109
23.5.29.4 Fields 2109
23.5.30 General Control Register 1 - Lane a (LNAGCR1 - LNDGCR1) 2112
23.5.30.1 Offset 2112
23.5.30.2 Function 2112
23.5.30.3 Diagram 2112
23.5.30.4 Fields 2113
23.5.31 Speed Switch Control Register 0 - Lane a (LNASSCR0 - LNDSSCR0) 2115
23.5.31.1 Offset 2115
23.5.31.2 Function 2116
23.5.31.3 Diagram 2116
23.5.31.4 Fields 2116
23.5.32 Receive Equalization Control Register 0 - Lane a (LNARECR0 - LNDRECR0) 2120
23.5.32.1 Offset 2120
23.5.32.2 Function 2120
23.5.32.3 Diagram 2120
23.5.32.4 Fields 2120
23.5.33 Receive Equalization Control Register 1- Lane a (LNARECR1 - LNDRECR1) 2123
23.5.33.1 Offset 2123
23.5.33.2 Function 2123
23.5.33.3 Diagram 2123
23.5.33.4 Fields 2123
23.5.34 Transmit Equalization Control Register 0 - Lane a (LNATECR0 - LNDTECR0) 2124

23.5.34.1 Offset....2124
23.5.34.2 Function.... 2124
23.5.34.3 Diagram....2124
23.5.34.4 Fields....2125
23.5.35 Speed Switch Control Register 1- Lane 0 (LNASSCR1 - LNDSSCR1)....2127
23.5.35.1 Offset....2127
23.5.35.2 Function.... 2127
23.5.35.3 Diagram....2128
23.5.35.4 Fields....2128
23.5.36 TTL Control Register 0 - Lane a (LNATTLCR0 - LNDTTLCR0)....2131
23.5.36.1 Offset....2131
23.5.36.2 Function.... 2132
23.5.36.3 Diagram....2132
23.5.36.4 Fields....2132
23.5.37 Test Control/Status Register 3 - Lane a (LNATCSR3 - LNDTCSR3)....2133
23.5.37.1 Offset....2133
23.5.37.2 Function.... 2133
23.5.37.3 Diagram....2133
23.5.37.4 Fields....2133
23.5.38 PEXa Protocol Control Register 0 (PEXACR0 - PEXCCR0)....2134
23.5.38.1 Offset....2134
23.5.38.2 Function.... 2134
23.5.38.3 Diagram....2134
23.5.38.4 Fields....2135
23.5.39 SGMIIa Protocol Control Register 1 (SGMIIACR1 - SGMIIDCR1).... 2135
23.5.39.1 Offset....2135
23.5.39.2 Function.... 2136
23.5.39.3 Diagram....2136
23.5.39.4 Fields....2136

23.5.40 SGMIIa Protocol Control Register 3 (SGMIIACR3 - SGMIIDCR3)..... 2137
23.5.40.1 Offset.....2137
23.5.40.2 Function..... 2137
23.5.40.3 Diagram.....2137
23.5.40.4 Fields.....2137
23.5.41 QSGMIIa Protocol Control Register 1 (QSGMIIACR1 - QSGMIIBCR1).....2138
23.5.41.1 Offset.....2138
23.5.41.2 Function..... 2138
23.5.41.3 Diagram.....2138
23.5.41.4 Fields.....2139
23.5.42 QSGMIIa Protocol Control Register 3 (QSGMIIACR3 - QSGMIIBCR3).....2139
23.5.42.1 Offset.....2139
23.5.42.2 Function..... 2139
23.5.42.3 Diagram.....2140
23.5.42.4 Fields.....2140
23.5.43 XFIa Protocol Control Register 1 (XFIACR1 - XFIBCR1)..... 2141
23.5.43.1 Offset.....2141
23.5.43.2 Function..... 2141
23.5.43.3 Diagram.....2142
23.5.43.4 Fields.....2142
23.5.44 XFIa Protocol Control Register 3 (XFIACR3 - XFIBCR3)..... 2142
23.5.44.1 Offset.....2142
23.5.44.2 Function..... 2142
23.5.44.3 Diagram.....2143
23.5.44.4 Fields.....2143
23.6 MDIO register spaces..... 2144
23.6.1 MDIO_XFI_PMD register descriptions.....2145
23.6.1.1 MDIO_XFI_PMD memory map.....2145
23.6.1.2 XFI PMD Control 1 (XFI_PMD_CR1)..... 2146

23.6.1.2.1 Offset....2146
23.6.1.2.2 Function.... 2146
23.6.1.2.3 Diagram....2146
23.6.1.2.4 Fields....2146
23.6.1.3 XFI 10GBASE-KR PMD Control Register (XFI_10GKR_PMD_CR).... 2147
23.6.1.3.1 Offset....2147
23.6.1.3.2 Function.... 2147
23.6.1.3.3 Diagram....2147
23.6.1.3.4 Fields....2147
23.6.1.4 XFI 10GBASE-KR PMD Status Register (XFI_10GKR_PMD_SR).... 2148
23.6.1.4.1 Offset....2148
23.6.1.4.2 Function.... 2148
23.6.1.4.3 Diagram....2148
23.6.1.4.4 Fields....2148
23.6.1.5 XFI 10GBASE-KR LP Coefficient Update Register (XFI_10GKR_LP_COEF_UPDATE). 2149
23.6.1.5.1 Offset....2149
23.6.1.5.2 Function.... 2149
23.6.1.5.3 Diagram....2150
23.6.1.5.4 Fields....2150
23.6.1.6 XFI 10GBASE-KR LP Status Report Register (XFI_10GKR_LP_SRR)....2151
23.6.1.6.1 Offset....2151
23.6.1.6.2 Function.... 2151
23.6.1.6.3 Diagram....2151
23.6.1.6.4 Fields....2151
23.6.1.7 XFI 10GBASE-KR LD Coefficient Update Register (XFI_10GKR_LD_COEF_UPDATE) 2152
23.6.1.7.1 Offset....2152
23.6.1.7.2 Function.... 2152
23.6.1.7.3 Diagram....2152
23.6.1.7.4 Fields....2153

23.6.1.8 XFI 10GBASE-KR LD Status Report Register (XFI_10GKR_LD_SRR)..... 2154
23.6.1.8.1 Offset.....2154
23.6.1.8.2 Function..... 2154
23.6.1.8.3 Diagram.....2154
23.6.1.8.4 Fields.....2154
23.6.1.9 XFI 10GBASE-R FEC Ability (XFI_10GR_FEC_ABIL)..... 2155
23.6.1.9.1 Offset.....2155
23.6.1.9.2 Function..... 2155
23.6.1.9.3 Diagram.....2155
23.6.1.9.4 Fields.....2155
23.6.1.10 XFI 10GBASE-R FEC Control (XFI_10GR_FEC_CTL)..... 2156
23.6.1.10.1 Offset.....2156
23.6.1.10.2 Function..... 2156
23.6.1.10.3 Diagram.....2156
23.6.1.10.4 Fields.....2156
23.6.1.11 XFI 10GBASE-R FEC Corrected Blocks Counter Lower (XFI_10GR_FEC_COR_BLK_CNT_L)..... 2157
23.6.1.11.1 Offset.....2157
23.6.1.11.2 Function..... 2157
23.6.1.11.3 Diagram.....2157
23.6.1.11.4 Fields.....2157
23.6.1.12 XFI 10GBASE-R FEC Corrected Blocks Counter Upper (XFI_10GR_FEC_COR_BLK_CNT_U).....2158
23.6.1.12.1 Offset.....2158
23.6.1.12.2 Function..... 2158
23.6.1.12.3 Diagram.....2158
23.6.1.12.4 Fields.....2158
23.6.1.13 XFI 10GBASE-R FEC Uncorrected Blocks Counter Lower (XFI_10GR_FEC_UNCOR_BLK_CNT_L)..... 2159
23.6.1.13.1 Offset.....2159

23.6.1.13.2 Function.... 2159
23.6.1.13.3 Diagram....2159
23.6.1.13.4 Fields....2159
23.6.1.14 XFI 10GBASE-R FEC Uncorrected Blocks Counter Upper (XFI_10GR_FEC_UNCOR_BLK_CNT_U)....2159
23.6.1.14.1 Offset....2159
23.6.1.14.2 Function.... 2160
23.6.1.14.3 Diagram....2160
23.6.1.14.4 Fields....2160
23.6.1.15 XFI 10GBASE-R Vendor-specific PMA Status (XFI_VEND_SPEC_PMA_STAT).... 2160
23.6.1.15.1 Offset....2160
23.6.1.15.2 Function.... 2160
23.6.1.15.3 Diagram....2161
23.6.1.15.4 Fields....2161
23.6.1.16 XFI Number of PRBS Sequence Errors Lower (XFI_10GKR_NPRBS_ERR_CNT_LO).....2161
23.6.1.16.1 Offset....2161
23.6.1.16.2 Function.... 2161
23.6.1.16.3 Diagram....2162
23.6.1.16.4 Fields....2162
23.6.1.17 XFI Number of PRBS Sequence Errors Upper (XFI_10GKR_NPRBS_ERR_CNT_HI)..... 2162
23.6.1.17.1 Offset....2162
23.6.1.17.2 Function.... 2162
23.6.1.17.3 Diagram....2162
23.6.1.17.4 Fields....2163
23.6.2 MDIO_XFI_PCS register descriptions.... 2163
23.6.2.1 MDIO_XFI_PCS memory map.... 2163
23.6.2.2 XFI PCS Control 1 (XFI_PCS_CR1).... 2164
23.6.2.2.1 Offset....2164
23.6.2.2.2 Function.... 2164

23.6.2.2.3 Diagram....2164
23.6.2.2.4 Fields....2164
23.6.2.3 XFI PCS Status 1 (XFI_PCS_SR1).... 2165
23.6.2.3.1 Offset....2165
23.6.2.3.2 Function.... 2166
23.6.2.3.3 Diagram....2166
23.6.2.3.4 Fields....2166
23.6.2.4 XFI PCS Device Identifier Upper (XFI_PCS_DEV_ID_H).... 2167
23.6.2.4.1 Offset....2167
23.6.2.4.2 Function.... 2167
23.6.2.4.3 Diagram....2168
23.6.2.4.4 Fields....2168
23.6.2.5 XFI PCS Device Identifier Lower (XFI_PCS_DEV_ID_L).... 2168
23.6.2.5.1 Offset....2168
23.6.2.5.2 Function.... 2168
23.6.2.5.3 Diagram....2168
23.6.2.5.4 Fields....2168
23.6.2.6 XFI PCS Speed Ability (XFI_PCS_SPEED_ABIL).... 2169
23.6.2.6.1 Offset....2169
23.6.2.6.2 Function.... 2169
23.6.2.6.3 Diagram....2169
23.6.2.6.4 Fields....2169
23.6.2.7 XFI PCS Devices In Package 0 (XFI_PCS_DEV_PRES0)....2170
23.6.2.7.1 Offset....2170
23.6.2.7.2 Function.... 2170
23.6.2.7.3 Diagram....2170
23.6.2.7.4 Fields....2170
23.6.2.8 XFI PCS Devices in Package 1 (XFI_PCS_DEV_PRES1)....2171
23.6.2.8.1 Offset....2171

23.6.2.8.2 Function.... 2171
23.6.2.8.3 Diagram....2172
23.6.2.8.4 Fields....2172
23.6.2.9 XFI 10G PCS Control 2 (XFI_PCS_CR2).... 2172
23.6.2.9.1 Offset....2172
23.6.2.9.2 Function.... 2173
23.6.2.9.3 Diagram....2173
23.6.2.9.4 Fields....2173
23.6.2.10 XFI 10G PCS Status 2 (XFI_PCS_SR2).... 2173
23.6.2.10.1 Offset....2173
23.6.2.10.2 Function.... 2173
23.6.2.10.3 Diagram....2173
23.6.2.10.4 Fields....2174
23.6.2.11 XFI PCS Package Identifier Upper (XFI_PCS_PKG_ID_H)....2175
23.6.2.11.1 Offset....2175
23.6.2.11.2 Function.... 2175
23.6.2.11.3 Diagram....2175
23.6.2.11.4 Fields....2175
23.6.2.12 XFI PCS Package Identifier Lower (XFI_PCS_PKG_ID_L)....2175
23.6.2.12.1 Offset....2175
23.6.2.12.2 Function.... 2176
23.6.2.12.3 Diagram....2176
23.6.2.12.4 Fields....2176
23.6.2.13 XFI 10GBASE-R PCS Status 1 (XFI_PCS_10GR_SR1)....2176
23.6.2.13.1 Offset....2176
23.6.2.13.2 Function.... 2176
23.6.2.13.3 Diagram....2176
23.6.2.13.4 Fields....2177
23.6.2.14 XFI 10GBASE-R PCS Status 2 (XFI_PCS_10GR_SR2)....2177

23.6.2.14.1 Offset....2177
23.6.2.14.2 Function.... 2178
23.6.2.14.3 Diagram....2178
23.6.2.14.4 Fields....2178
23.6.2.15 XFI 10GBASE-R PCS Test Pattern Seed A 0 (XFI_PCS_TP_SEED_A0).... 2179
23.6.2.15.1 Offset....2179
23.6.2.15.2 Function.... 2179
23.6.2.15.3 Diagram....2179
23.6.2.15.4 Fields....2179
23.6.2.16 XFI 10GBASE-R PCS Test Pattern Seed A 1 (XFI_PCS_TP_SEED_A1).... 2179
23.6.2.16.1 Offset....2179
23.6.2.16.2 Function.... 2180
23.6.2.16.3 Diagram....2180
23.6.2.16.4 Fields....2180
23.6.2.17 XFI 10GBASE-R PCS Test Pattern Seed A 2 (XFI_PCS_TP_SEED_A2).... 2180
23.6.2.17.1 Offset....2180
23.6.2.17.2 Function.... 2180
23.6.2.17.3 Diagram....2180
23.6.2.17.4 Fields....2181
23.6.2.18 XFI 10GBASE-R PCS Test Pattern Seed A 3 (XFI_PCS_TP_SEED_A3).... 2181
23.6.2.18.1 Offset....2181
23.6.2.18.2 Function.... 2181
23.6.2.18.3 Diagram....2181
23.6.2.18.4 Fields....2181
23.6.2.19 XFI 10GBASE-R PCS Test Pattern Seed B 0 (XFI_PCS_TP_SEED_B0)....2182
23.6.2.19.1 Offset....2182
23.6.2.19.2 Function.... 2182
23.6.2.19.3 Diagram....2182
23.6.2.19.4 Fields....2182

Section number	Title	Page

23.6.2.20 XFI 10GBASE-R PCS Test Pattern Seed B 1 (XFI_PCS_TP_SEED_B1).....2183

23.6.2.20.1 Offset.....2183

23.6.2.20.2 Function..... 2183

23.6.2.20.3 Diagram.....2183

23.6.2.20.4 Fields.....2183

23.6.2.21 XFI 10GBASE-R PCS Test Pattern Seed B 2 (XFI_PCS_TP_SEED_B2).....2183

23.6.2.21.1 Offset.....2183

23.6.2.21.2 Function..... 2184

23.6.2.21.3 Diagram.....2184

23.6.2.21.4 Fields.....2184

23.6.2.22 XFI 10GBASE-R PCS Test Pattern Seed B 3 (XFI_PCS_TP_SEED_B3).....2184

23.6.2.22.1 Offset.....2184

23.6.2.22.2 Function..... 2184

23.6.2.22.3 Diagram.....2184

23.6.2.22.4 Fields.....2185

23.6.2.23 XFI 10GBASE-R PCS Test Pattern Control (XFI_PCS_TP_CR)..... 2185

23.6.2.23.1 Offset.....2185

23.6.2.23.2 Function..... 2185

23.6.2.23.3 Diagram.....2185

23.6.2.23.4 Fields.....2186

23.6.2.24 XFI 10GBASE-R PCS Test Pattern Error Counter (XFI_PCS_TP_ERR_CNT).....2187

23.6.2.24.1 Offset.....2187

23.6.2.24.2 Function..... 2187

23.6.2.24.3 Diagram.....2187

23.6.2.24.4 Fields.....2187

23.6.2.25 Vendor Specific PCS Status (XFI_PCS_VENDOR_SR).....2187

23.6.2.25.1 Offset.....2188

23.6.2.25.2 Function..... 2188

23.6.2.25.3 Diagram.....2188

23.6.2.25.4 Fields....2188

23.6.3 MDIO_XFI_AN register descriptions.... 2189

23.6.3.1 MDIO_XFI_AN memory map....2189

23.6.3.2 XFI AN Control (XF_AN_CR).... 2189

23.6.3.2.1 Offset....2189

23.6.3.2.2 Function.... 2190

23.6.3.2.3 Diagram....2190

23.6.3.2.4 Fields....2190

23.6.3.3 XFI AN Status (XF_AN_SR).... 2191

23.6.3.3.1 Offset....2191

23.6.3.3.2 Function.... 2191

23.6.3.3.3 Diagram....2191

23.6.3.3.4 Fields....2191

23.6.3.4 XFI AN Device Identifier Upper (XFI_AN_DEV_ID_H).... 2192

23.6.3.4.1 Offset....2193

23.6.3.4.2 Function.... 2193

23.6.3.4.3 Diagram....2193

23.6.3.4.4 Fields....2193

23.6.3.5 XFI AN Device Identifier Lower (XFI_DEV_ID_L)....2193

23.6.3.5.1 Offset....2193

23.6.3.5.2 Function.... 2193

23.6.3.5.3 Diagram....2194

23.6.3.5.4 Fields....2194

23.6.3.6 XFI AN Devices In Package 0 (XFI_AN_DEV_PRES0).... 2194

23.6.3.6.1 Offset....2194

23.6.3.6.2 Function.... 2194

23.6.3.6.3 Diagram....2194

23.6.3.6.4 Fields....2195

23.6.3.7 XFI AN Devices In Package 1 (XFI_AN_DEV_PRES1).... 2196

23.6.3.7.1 Offset....2196
23.6.3.7.2 Function.... 2196
23.6.3.7.3 Diagram....2196
23.6.3.7.4 Fields....2196
23.6.3.8 XFI AN Package Identifier Upper (XF_AN_PKG_ID_H)....2197
23.6.3.8.1 Offset....2197
23.6.3.8.2 Function.... 2197
23.6.3.8.3 Diagram....2197
23.6.3.8.4 Fields....2197
23.6.3.9 XFI AN Package Identifier Lower (XFI_AN_PKG_ID_L).... 2197
23.6.3.9.1 Offset....2197
23.6.3.9.2 Function.... 2198
23.6.3.9.3 Diagram....2198
23.6.3.9.4 Fields....2198
23.6.3.10 XFI AN Advertisement 0 (XFI_AN_ADVERT0)....2198
23.6.3.10.1 Offset....2198
23.6.3.10.2 Function.... 2198
23.6.3.10.3 Diagram....2199
23.6.3.10.4 Fields....2199
23.6.3.11 XFI AN Advertisement 1 (XFI_AN_ADVERT1)....2200
23.6.3.11.1 Offset....2200
23.6.3.11.2 Function.... 2200
23.6.3.11.3 Diagram....2200
23.6.3.11.4 Fields....2200
23.6.3.12 XFI AN Advertisement 2 (XFI_AN_ADVERT2)....2201
23.6.3.12.1 Offset....2201
23.6.3.12.2 Function.... 2201
23.6.3.12.3 Diagram....2201
23.6.3.12.4 Fields....2201

23.6.3.13 XFI AN LP Base Page Ability 0 (XFI_AN_LP_BASE_PG_ABIL0).....2202
23.6.3.13.1 Offset.....2202
23.6.3.13.2 Function..... 2202
23.6.3.13.3 Diagram.....2202
23.6.3.13.4 Fields.....2202
23.6.3.14 XFI AN LP Base Page Ability 1 (XFI_AN_LP_BASE_PG_ABIL1).....2203
23.6.3.14.1 Offset.....2203
23.6.3.14.2 Function..... 2204
23.6.3.14.3 Diagram.....2204
23.6.3.14.4 Fields.....2204
23.6.3.15 XFI AN LP Base Page Ability 2 (XFI_AN_LP_BASE_PG_ABIL2).....2204
23.6.3.15.1 Offset.....2204
23.6.3.15.2 Function..... 2204
23.6.3.15.3 Diagram.....2205
23.6.3.15.4 Fields.....2205
23.6.3.16 XFI AN Extended Next Page Transmit 0 (XFI_AN_XNP_TX0)..... 2205
23.6.3.16.1 Offset.....2205
23.6.3.16.2 Function..... 2205
23.6.3.16.3 Diagram.....2206
23.6.3.16.4 Fields.....2206
23.6.3.17 XFI AN Extended Next Page Transmit 1 (XFI_AN_XNP_TX1)..... 2207
23.6.3.17.1 Offset.....2207
23.6.3.17.2 Function..... 2207
23.6.3.17.3 Diagram.....2207
23.6.3.17.4 Fields.....2207
23.6.3.18 XFI AN Extended Next Page Transmit 2 (XFI_AN_XNP_TX2)..... 2207
23.6.3.18.1 Offset.....2207
23.6.3.18.2 Function..... 2207
23.6.3.18.3 Diagram.....2208

23.6.3.18.4 Fields....2208
23.6.3.19 XFI AN LP Extended Next Page Ability 0 (XFI_AN_LP_XNP_ABIL0)....2208
23.6.3.19.1 Offset....2208
23.6.3.19.2 Function....2208
23.6.3.19.3 Diagram....2208
23.6.3.19.4 Fields....2209
23.6.3.20 XFI AN LP Extended Next Page Ability 1 (XFI_AN_LP_XNP_ABIL1)....2209
23.6.3.20.1 Offset....2210
23.6.3.20.2 Function....2210
23.6.3.20.3 Diagram....2210
23.6.3.20.4 Fields....2210
23.6.3.21 XFI AN LP Extended Next Page Ability 2 (XFI_AN_LP_XNP_ABIL2)....2210
23.6.3.21.1 Offset....2210
23.6.3.21.2 Function....2210
23.6.3.21.3 Diagram....2211
23.6.3.21.4 Fields....2211
23.6.3.22 XFI Backplane Ethernet Status (XFI_BP_STAT)....2211
23.6.3.22.1 Offset....2211
23.6.3.22.2 Function....2211
23.6.3.22.3 Diagram....2211
23.6.3.22.4 Fields....2212
23.6.4 MDIO_XFI_VENDOR_SPEC register descriptions....2212
23.6.4.1 MDIO_XFI_VENDOR_SPEC memory map....2213
23.6.4.2 XFI Revision (XFI_VND_REV)....2213
23.6.4.2.1 Offset....2213
23.6.4.2.2 Function....2213
23.6.4.2.3 Diagram....2213
23.6.4.2.4 Fields....2214
23.6.4.3 XFI Scratch (XFI_VND_SCRATCH)....2214

23.6.4.3.1 Offset....2214
23.6.4.3.2 Function.... 2214
23.6.4.3.3 Diagram....2214
23.6.4.3.4 Fields....2214
23.6.4.4 XFI PCS Interrupt Event (XFI_VND_PCS_INT).... 2215
23.6.4.4.1 Offset....2215
23.6.4.4.2 Function.... 2215
23.6.4.4.3 Diagram....2215
23.6.4.4.4 Fields....2215
23.6.4.5 XFI PCS Interrupt Mask (XFI_VND_PCS_INT_MSK).... 2217
23.6.4.5.1 Offset....2217
23.6.4.5.2 Function.... 2217
23.6.4.5.3 Diagram....2217
23.6.4.5.4 Fields....2217
23.6.4.6 XFI Auto-Negotiation Interrupt Event (XFI_VND_AN_INT).... 2218
23.6.4.6.1 Offset....2219
23.6.4.6.2 Function.... 2219
23.6.4.6.3 Diagram....2219
23.6.4.6.4 Fields....2219
23.6.4.7 XFI Auto-Negotiation Interrupt Mask (XFI_VND_AN_INT_MSK).... 2220
23.6.4.7.1 Offset....2220
23.6.4.7.2 Function.... 2220
23.6.4.7.3 Diagram....2220
23.6.4.7.4 Fields....2221
23.6.4.8 XFI Link Training Interrupt Event (XFI_VND_LT_INT).... 2221
23.6.4.8.1 Offset....2221
23.6.4.8.2 Function.... 2222
23.6.4.8.3 Diagram....2222
23.6.4.8.4 Fields....2222

23.6.4.9 XFI Link Training Interrupt Mask (XFI_VND_LT_INT_MSK)....2223
23.6.4.9.1 Offset....2223
23.6.4.9.2 Function....2223
23.6.4.9.3 Diagram....2223
23.6.4.9.4 Fields....2223
23.6.5 MDIO_KX_PCS register descriptions....2224
23.6.5.1 MDIO_KX_PCS memory map....2224
23.6.5.2 KX PCS Control (KX_PCS_CR)....2225
23.6.5.2.1 Offset....2225
23.6.5.2.2 Function....2225
23.6.5.2.3 Diagram....2226
23.6.5.2.4 Fields....2226
23.6.5.3 KX PCS Status (KX_PCS_SR)....2226
23.6.5.3.1 Offset....2227
23.6.5.3.2 Function....2227
23.6.5.3.3 Diagram....2227
23.6.5.3.4 Fields....2227
23.6.5.4 KX PCS Device Identifier Upper (KX_PCS_DEV_ID)....2227
23.6.5.4.1 Offset....2227
23.6.5.4.2 Function....2228
23.6.5.4.3 Diagram....2228
23.6.5.4.4 Fields....2228
23.6.5.5 KX PCS Device Identifier Lower (KX_PCS_DEV_ID_L)....2228
23.6.5.5.1 Offset....2228
23.6.5.5.2 Function....2228
23.6.5.5.3 Diagram....2229
23.6.5.5.4 Fields....2229
23.6.5.6 KX PCS Devices In Package 0 (KX_PCS_DEV_PRES0)....2229
23.6.5.6.1 Offset....2229

23.6.5.6.2 Function.... 2229
23.6.5.6.3 Diagram....2229
23.6.5.6.4 Fields....2230
23.6.5.7 KX PCS Devices In Package 1 (KX_PCS_DEV_PRES1)....2230
23.6.5.7.1 Offset....2231
23.6.5.7.2 Function.... 2231
23.6.5.7.3 Diagram....2231
23.6.5.7.4 Fields....2231
23.6.5.8 KX PCS Package Identifier Upper (KX_PCS_PKG_ID_U)....2232
23.6.5.8.1 Offset....2232
23.6.5.8.2 Function.... 2232
23.6.5.8.3 Diagram....2232
23.6.5.8.4 Fields....2232
23.6.5.9 KX PCS Package Identifier Lower (KX_PCS_PKG_ID_L)....2232
23.6.5.9.1 Offset....2232
23.6.5.9.2 Function.... 2232
23.6.5.9.3 Diagram....2233
23.6.5.9.4 Fields....2233
23.6.5.10 SGMII Control (C45_SGMII_CR).... 2233
23.6.5.10.1 Offset....2233
23.6.5.10.2 Function.... 2233
23.6.5.10.3 Diagram....2233
23.6.5.10.4 Fields....2234
23.6.5.11 SGMII Status (C45_SGMII_SR)....2235
23.6.5.11.1 Offset....2235
23.6.5.11.2 Function.... 2235
23.6.5.11.3 Diagram....2235
23.6.5.11.4 Fields....2236
23.6.5.12 SGMII PHY Identifier Upper (C45_SGMII_PHY_ID_H)....2237

23.6.5.12.1 Offset....2237
23.6.5.12.2 Function.... 2237
23.6.5.12.3 Diagram....2237
23.6.5.12.4 Fields....2237
23.6.5.13 SGMII PHY Identifier Lower (C45_SGMII_PHY_ID_L)....2237
23.6.5.13.1 Offset....2237
23.6.5.13.2 Function.... 2237
23.6.5.13.3 Diagram....2238
23.6.5.13.4 Fields....2238
23.6.5.14 SGMII Device Ability for 1000Base-X (C45_SGMII_DEV_ABIL_1KBX).... 2238
23.6.5.14.1 Offset....2238
23.6.5.14.2 Function.... 2238
23.6.5.14.3 Diagram....2238
23.6.5.14.4 Fields....2239
23.6.5.15 SGMII Device Ability for SGMII (C45_SGMII_DEV_ABIL_SGMII).... 2240
23.6.5.15.1 Offset....2240
23.6.5.15.2 Function.... 2240
23.6.5.15.3 Diagram....2240
23.6.5.15.4 Fields....2240
23.6.5.16 SGMII Partner Ability for 1000Base-X (C45_SGMII_LP_DEV_ABIL_1KBX).... 2241
23.6.5.16.1 Offset....2241
23.6.5.16.2 Function.... 2241
23.6.5.16.3 Diagram....2241
23.6.5.16.4 Fields....2241
23.6.5.17 SGMII Partner Ability for SGMII (C45_SGMII_LP_DEV_ABIL_SGMII).... 2242
23.6.5.17.1 Offset....2242
23.6.5.17.2 Function.... 2242
23.6.5.17.3 Diagram....2243
23.6.5.17.4 Fields....2243

Section number	Title	Page

23.6.5.18 SGMII AN Expansion (C45_SGMII_AN_EXP)....2244

23.6.5.18.1 Offset....2244

23.6.5.18.2 Function....2244

23.6.5.18.3 Diagram....2244

23.6.5.18.4 Fields....2244

23.6.5.19 SGMII Next Page Transmit (C45_SGMII_NP_TX)....2245

23.6.5.19.1 Offset....2245

23.6.5.19.2 Function....2245

23.6.5.19.3 Diagram....2245

23.6.5.19.4 Fields....2245

23.6.5.20 SGMII LP Next Page Receive (C45_SGMII_NP_RX)....2246

23.6.5.20.1 Offset....2246

23.6.5.20.2 Function....2246

23.6.5.20.3 Diagram....2246

23.6.5.20.4 Fields....2246

23.6.5.21 SGMII Extended Status (C45_SGMII_XTND_STAT)....2247

23.6.5.21.1 Offset....2247

23.6.5.21.2 Function....2247

23.6.5.21.3 Diagram....2247

23.6.5.21.4 Fields....2248

23.6.5.22 SGMII Scratch (C45_SGMII_SCRATCH)....2248

23.6.5.22.1 Offset....2248

23.6.5.22.2 Function....2248

23.6.5.22.3 Diagram....2248

23.6.5.22.4 Fields....2248

23.6.5.23 SGMII Design Revision (C45_SGMII_REV)....2248

23.6.5.23.1 Offset....2248

23.6.5.23.2 Function....2249

23.6.5.23.3 Diagram....2249

23.6.5.23.4 Fields....2249
23.6.5.24 SGMII Link Timer Lower (C45_SGMII_LINK_TMR_L).... 2249
23.6.5.24.1 Offset....2249
23.6.5.24.2 Function.... 2249
23.6.5.24.3 Diagram....2250
23.6.5.24.4 Fields....2250
23.6.5.25 SGMII Link Timer Upper (C45_SGMII_LINK_TMR_H).... 2250
23.6.5.25.1 Offset....2250
23.6.5.25.2 Function.... 2250
23.6.5.25.3 Diagram....2250
23.6.5.25.4 Fields....2251
23.6.5.26 SGMII IF Mode (C45_SGMII_IF_MODE).... 2251
23.6.5.26.1 Offset....2251
23.6.5.26.2 Function.... 2251
23.6.5.26.3 Diagram....2251
23.6.5.26.4 Fields....2251
23.6.6 MDIO_KX_AN register descriptions.... 2252
23.6.6.1 MDIO_KX_AN memory map.... 2252
23.6.6.2 KX AN Control (KX_AN_CR).... 2253
23.6.6.2.1 Offset....2253
23.6.6.2.2 Function.... 2253
23.6.6.2.3 Diagram....2253
23.6.6.2.4 Fields....2254
23.6.6.3 KX AN Status (KX_AN_SR).... 2255
23.6.6.3.1 Offset....2255
23.6.6.3.2 Function.... 2255
23.6.6.3.3 Diagram....2255
23.6.6.3.4 Fields....2255
23.6.6.4 KX AN Device Identifier Upper (KX_AN_DEV_ID_H).... 2256

23.6.6.4.1 Offset....2256
23.6.6.4.2 Function.... 2257
23.6.6.4.3 Diagram....2257
23.6.6.4.4 Fields....2257
23.6.6.5 KX AN Device Identifier Lower (KX_AN_DEV_ID_L).... 2257
23.6.6.5.1 Offset....2257
23.6.6.5.2 Function.... 2257
23.6.6.5.3 Diagram....2257
23.6.6.5.4 Fields....2258
23.6.6.6 KX AN Devices in Package 0 (KX_AN_DEV_PRES0)....2258
23.6.6.6.1 Offset....2258
23.6.6.6.2 Function.... 2258
23.6.6.6.3 Diagram....2258
23.6.6.6.4 Fields....2259
23.6.6.7 KX AN Devices in Package 1 (KX_AN_DEV_PRES1)....2259
23.6.6.7.1 Offset....2260
23.6.6.7.2 Function.... 2260
23.6.6.7.3 Diagram....2260
23.6.6.7.4 Fields....2260
23.6.6.8 KX AN Package Identifier Upper (KX_AN_PKG_ID_H)....2261
23.6.6.8.1 Offset....2261
23.6.6.8.2 Function.... 2261
23.6.6.8.3 Diagram....2261
23.6.6.8.4 Fields....2261
23.6.6.9 KX AN Package Identifier Lower (KX_AN_PKG_ID_L)....2261
23.6.6.9.1 Offset....2261
23.6.6.9.2 Function.... 2261
23.6.6.9.3 Diagram....2262
23.6.6.9.4 Fields....2262

23.6.6.10 KX AN Advertisement 0 (KX_AN_ADVERT0)..... 2262
23.6.6.10.1 Offset.....2262
23.6.6.10.2 Function..... 2262
23.6.6.10.3 Diagram.....2262
23.6.6.10.4 Fields.....2263
23.6.6.11 KX AN Advertisement 1 (KX_AN_ADVERT1)..... 2264
23.6.6.11.1 Offset.....2264
23.6.6.11.2 Function..... 2264
23.6.6.11.3 Diagram.....2264
23.6.6.11.4 Fields.....2264
23.6.6.12 KX AN Advertisement 2 (KX_AN_ADVERT2)..... 2265
23.6.6.12.1 Offset.....2265
23.6.6.12.2 Function..... 2265
23.6.6.12.3 Diagram.....2265
23.6.6.12.4 Fields.....2265
23.6.6.13 KX AN LP Base Page Ability 0 (KX_AN_LP_BASE_PG_ABIL0)..... 2266
23.6.6.13.1 Offset.....2266
23.6.6.13.2 Function..... 2266
23.6.6.13.3 Diagram.....2266
23.6.6.13.4 Fields.....2266
23.6.6.14 KX AN LP Base Page Ability 1 (KX_AN_LP_BASE_PG_ABIL1)..... 2267
23.6.6.14.1 Offset.....2267
23.6.6.14.2 Function..... 2267
23.6.6.14.3 Diagram.....2268
23.6.6.14.4 Fields.....2268
23.6.6.15 KX AN LP Base Page Ability 2 (KX_AN_LP_BASE_PG_ABIL2)..... 2268
23.6.6.15.1 Offset.....2268
23.6.6.15.2 Function..... 2269
23.6.6.15.3 Diagram.....2269

23.6.6.15.4 Fields....2269
23.6.6.16 KX AN XNP Transmit 0 (KX_AN_XNP_TX0)....2269
23.6.6.16.1 Offset....2269
23.6.6.16.2 Function....2270
23.6.6.16.3 Diagram....2270
23.6.6.16.4 Fields....2270
23.6.6.17 KX AN XNP Transmit 1 (KX_AN_XNP_TX1)....2271
23.6.6.17.1 Offset....2271
23.6.6.17.2 Function....2271
23.6.6.17.3 Diagram....2271
23.6.6.17.4 Fields....2271
23.6.6.18 KX AN XNP Transmit 2 (KX_AN_XNP_TX2)....2271
23.6.6.18.1 Offset....2272
23.6.6.18.2 Function....2272
23.6.6.18.3 Diagram....2272
23.6.6.18.4 Fields....2272
23.6.6.19 KX AN LP XNP Ability 0 (KX_AN_LP_XNP_ABIL0)....2272
23.6.6.19.1 Offset....2272
23.6.6.19.2 Function....2272
23.6.6.19.3 Diagram....2273
23.6.6.19.4 Fields....2273
23.6.6.20 KX AN LP XNP Ability 1 (KX_AN_LP_XNP_ABIL1)....2274
23.6.6.20.1 Offset....2274
23.6.6.20.2 Function....2274
23.6.6.20.3 Diagram....2274
23.6.6.20.4 Fields....2274
23.6.6.21 KX AN LP XNP Ability 2 (KX_AN_LP_XNP_ABIL2)....2274
23.6.6.21.1 Offset....2274
23.6.6.21.2 Function....2274

23.6.6.21.3 Diagram....2275
23.6.6.21.4 Fields....2275
23.6.6.22 KX Backplane Ethernet Status (KX_BP_STAT)....2275
23.6.6.22.1 Offset....2275
23.6.6.22.2 Function....2275
23.6.6.22.3 Diagram....2275
23.6.6.22.4 Fields....2276
23.6.6.23 KX Millisecond Count (KX_MS_CNT)....2276
23.6.6.23.1 Offset....2277
23.6.6.23.2 Function....2277
23.6.6.23.3 Diagram....2277
23.6.6.23.4 Fields....2277
23.6.7 MDIO_KX_VENDOR_SPEC register descriptions....2277
23.6.7.1 MDIO_KX_VENDOR_SPEC memory map....2277
23.6.7.2 KX Revision (KX_VND_REV)....2278
23.6.7.2.1 Offset....2278
23.6.7.2.2 Function....2278
23.6.7.2.3 Diagram....2278
23.6.7.2.4 Fields....2278
23.6.7.3 KX Scratch (KX_VND_SCRATCH)....2279
23.6.7.3.1 Offset....2279
23.6.7.3.2 Function....2279
23.6.7.3.3 Diagram....2279
23.6.7.3.4 Fields....2279
23.6.7.4 KX PCS Interrupt Event (KX_VND_PCS_INT)....2279
23.6.7.4.1 Offset....2279
23.6.7.4.2 Function....2280
23.6.7.4.3 Diagram....2280
23.6.7.4.4 Fields....2280

23.6.7.5 KX PCS Interrupt Mask (KX_VND_INT_MSK)..... 2280
23.6.7.5.1 Offset.....2280
23.6.7.5.2 Function..... 2281
23.6.7.5.3 Diagram.....2281
23.6.7.5.4 Fields.....2281
23.6.7.6 KX AN Interrupt Event (KX_VND_AN_INT)..... 2281
23.6.7.6.1 Offset.....2281
23.6.7.6.2 Function..... 2281
23.6.7.6.3 Diagram.....2282
23.6.7.6.4 Fields.....2282
23.6.7.7 KX AN Interrupt Mask (KX_VND_AN_INT_MSK)..... 2282
23.6.7.7.1 Offset.....2283
23.6.7.7.2 Function..... 2283
23.6.7.7.3 Diagram.....2283
23.6.7.7.4 Fields.....2283
23.6.8 MDIO_SGMII register descriptions..... 2284
23.6.8.1 MDIO_SGMII memory map..... 2284
23.6.8.2 SGMII Control (SGMII_CR).....2284
23.6.8.2.1 Offset.....2284
23.6.8.2.2 Function..... 2285
23.6.8.2.3 Diagram.....2285
23.6.8.2.4 Fields.....2285
23.6.8.3 SGMII Status (SGMII_SR).....2286
23.6.8.3.1 Offset.....2286
23.6.8.3.2 Function..... 2286
23.6.8.3.3 Diagram.....2287
23.6.8.3.4 Fields.....2287
23.6.8.4 SGMII PHY Identifier Upper (SGMII_PHY_ID_H)..... 2288
23.6.8.4.1 Offset.....2288

23.6.8.4.2 Function.... 2288
23.6.8.4.3 Diagram....2288
23.6.8.4.4 Fields....2288
23.6.8.5 SGMII PHY Identifier Lower (SGMII_PHY_ID_L).... 2289
23.6.8.5.1 Offset....2289
23.6.8.5.2 Function.... 2289
23.6.8.5.3 Diagram....2289
23.6.8.5.4 Fields....2289
23.6.8.6 SGMII Device Ability for 1000Base-X (SGMII_DEV_ABIL_1KBX).... 2290
23.6.8.6.1 Offset....2290
23.6.8.6.2 Function.... 2290
23.6.8.6.3 Diagram....2290
23.6.8.6.4 Fields....2290
23.6.8.7 SGMII Device Ability for SGMII (SGMII_DEV_ABIL_SGMII)....2291
23.6.8.7.1 Offset....2291
23.6.8.7.2 Function.... 2291
23.6.8.7.3 Diagram....2291
23.6.8.7.4 Fields....2291
23.6.8.8 SGMII Partner Ability for 1000Base-X (SGMII_LP_DEV_ABIL_1KBX).... 2292
23.6.8.8.1 Offset....2292
23.6.8.8.2 Function.... 2292
23.6.8.8.3 Diagram....2292
23.6.8.8.4 Fields....2293
23.6.8.9 SGMII Partner Ability for SGMII (SGMII_LP_DEV_ABIL_SGMII)....2294
23.6.8.9.1 Offset....2294
23.6.8.9.2 Function.... 2294
23.6.8.9.3 Diagram....2294
23.6.8.9.4 Fields....2294
23.6.8.10 SGMII AN Expansion (SGMII_AN_EXP).... 2295

23.6.8.10.1 Offset....2295
23.6.8.10.2 Function....2295
23.6.8.10.3 Diagram....2296
23.6.8.10.4 Fields....2296
23.6.8.11 SGMII Next Page Transmit (SGMII_NP_TX)....2296
23.6.8.11.1 Offset....2296
23.6.8.11.2 Function....2296
23.6.8.11.3 Diagram....2296
23.6.8.11.4 Fields....2297
23.6.8.12 SGMII LP Next Page Receive (SGMII_NP_RX)....2297
23.6.8.12.1 Offset....2297
23.6.8.12.2 Function....2297
23.6.8.12.3 Diagram....2298
23.6.8.12.4 Fields....2298
23.6.8.13 SGMII Extended Status (SGMII_XTND_STAT)....2298
23.6.8.13.1 Offset....2298
23.6.8.13.2 Function....2299
23.6.8.13.3 Diagram....2299
23.6.8.13.4 Fields....2299
23.6.8.14 SGMII Scratch (SGMII_SCRATCH)....2299
23.6.8.14.1 Offset....2299
23.6.8.14.2 Function....2299
23.6.8.14.3 Diagram....2299
23.6.8.14.4 Fields....2300
23.6.8.15 SGMII Design Revision (SGMII_REV)....2300
23.6.8.15.1 Offset....2300
23.6.8.15.2 Function....2300
23.6.8.15.3 Diagram....2300
23.6.8.15.4 Fields....2300

23.6.8.16 SGMII Link Timer Lower (SGMII_LINK_TMR_L)....2301
23.6.8.16.1 Offset....2301
23.6.8.16.2 Function....2301
23.6.8.16.3 Diagram....2301
23.6.8.16.4 Fields....2301
23.6.8.17 SGMII Link Timer Upper (SGMII_LINK_TMR_H)....2302
23.6.8.17.1 Offset....2302
23.6.8.17.2 Function....2302
23.6.8.17.3 Diagram....2302
23.6.8.17.4 Fields....2302
23.6.8.18 SGMII IF Mode (SGMII_IF_MODE)....2302
23.6.8.18.1 Offset....2302
23.6.8.18.2 Function....2302
23.6.8.18.3 Diagram....2303
23.6.8.18.4 Fields....2303
23.6.9 MDIO_QSGMII register descriptions....2303
23.6.9.1 MDIO_QSGMII memory map....2304
23.6.9.2 QSGMII Control (QSGMII_CR)....2304
23.6.9.2.1 Offset....2304
23.6.9.2.2 Function....2304
23.6.9.2.3 Diagram....2305
23.6.9.2.4 Fields....2305
23.6.9.3 QSGMII Status (QSGMII_SR)....2306
23.6.9.3.1 Offset....2306
23.6.9.3.2 Function....2306
23.6.9.3.3 Diagram....2306
23.6.9.3.4 Fields....2307
23.6.9.4 QSGMII PHY Identifier Upper (QSGMII_PHY_ID_H)....2308
23.6.9.4.1 Offset....2308

23.6.9.4.2 Function.... 2308
23.6.9.4.3 Diagram....2308
23.6.9.4.4 Fields....2308
23.6.9.5 QSGMII PHY Identifier Lower (QSGMII_PHY_ID_L)....2309
23.6.9.5.1 Offset....2309
23.6.9.5.2 Function.... 2309
23.6.9.5.3 Diagram....2309
23.6.9.5.4 Fields....2309
23.6.9.6 QSGMII Device Ability for SGMII (QSGMII_DEV_ABIL_SGMII).... 2309
23.6.9.6.1 Offset....2309
23.6.9.6.2 Function.... 2310
23.6.9.6.3 Diagram....2310
23.6.9.6.4 Fields....2310
23.6.9.7 QSGMII Partner Ability for SGMII (QSGMII_LP_DEV_ABIL_SGMII).... 2311
23.6.9.7.1 Offset....2311
23.6.9.7.2 Function.... 2311
23.6.9.7.3 Diagram....2311
23.6.9.7.4 Fields....2311
23.6.9.8 QSGMII AN Expansion (QSGMII_AN_EXP)....2312
23.6.9.8.1 Offset....2312
23.6.9.8.2 Function.... 2312
23.6.9.8.3 Diagram....2313
23.6.9.8.4 Fields....2313
23.6.9.9 QSGMII Next Page Transmit (QSGMII_NP_TX).... 2313
23.6.9.9.1 Offset....2313
23.6.9.9.2 Function.... 2313
23.6.9.9.3 Diagram....2313
23.6.9.9.4 Fields....2314
23.6.9.10 QSGMII LP Next Page Receive (QSGMII_NP_RX)....2314

23.6.9.10.1 Offset....2314
23.6.9.10.2 Function....2314
23.6.9.10.3 Diagram....2315
23.6.9.10.4 Fields....2315
23.6.9.11 QSGMII Extended Status (QSGMII_XTND_STAT)....2315
23.6.9.11.1 Offset....2315
23.6.9.11.2 Function....2316
23.6.9.11.3 Diagram....2316
23.6.9.11.4 Fields....2316
23.6.9.12 QSGMII Scratch (QSGMII_SCRATCH)....2316
23.6.9.12.1 Offset....2316
23.6.9.12.2 Function....2316
23.6.9.12.3 Diagram....2316
23.6.9.12.4 Fields....2317
23.6.9.13 QSGMII Design Revision (QSGMII_REV)....2317
23.6.9.13.1 Offset....2317
23.6.9.13.2 Function....2317
23.6.9.13.3 Diagram....2317
23.6.9.13.4 Fields....2317
23.6.9.14 QSGMII Link Timer Lower (QSGMII_LINK_TMR_L)....2318
23.6.9.14.1 Offset....2318
23.6.9.14.2 Function....2318
23.6.9.14.3 Diagram....2318
23.6.9.14.4 Fields....2318
23.6.9.15 QSGMII Link Timer Upper (QSGMII_LINK_TMR_H)....2319
23.6.9.15.1 Offset....2319
23.6.9.15.2 Function....2319
23.6.9.15.3 Diagram....2319
23.6.9.15.4 Fields....2319

23.6.9.16 QSGMII IF Mode (QSGMII_IF_MODE) 2319
23.6.9.16.1 Offset 2319
23.6.9.16.2 Function 2319
23.6.9.16.3 Diagram 2320
23.6.9.16.4 Fields 2320
23.7 Initialization/Application Information 2320
23.7.1 Initialization 2321
23.7.1.1 1G SGMII 2321
23.7.1.2 2.5G SGMII 2321
23.7.1.3 1000Base-KX 2322
23.7.1.4 QSGMII 2323
23.7.1.5 XFI/10GBase-KR 2323
23.7.2 Unused Lanes 2324
23.7.3 Frequency Negotiation 2324
23.7.4 Soft Reset and Reconfiguring Procedures 2325
23.7.4.1 Lane Reset and Reconfiguration 2325
23.7.4.2 Lane Enable After Powerdown 2325
23.7.4.3 PLL Reset and Reconfiguration 2326
23.7.5 Quiesce Sequences for System Sleep 2326

Chapter 24
Serial Peripheral Interface

24.1 Introduction 2327
24.1.1 Block diagram 2327
24.1.2 Features 2328
24.1.3 Interface configurations 2330
24.1.3.1 SPI configuration 2330
24.1.4 Modes of operation 2331
24.1.4.1 Master mode 2331
24.1.4.2 Slave mode 2331

24.1.4.3 Module Disable mode.... 2331
24.1.4.4 External Stop mode.... 2332
24.1.4.5 Debug mode.... 2332
24.2 Module signal descriptions.... 2332
24.2.1 HT.... 2332
24.2.2 PCS0/SS_b.... 2333
24.2.3 SCK.... 2333
24.2.4 SIN.... 2333
24.2.5 SOUT.... 2334
24.3 SPI register descriptions.... 2334
24.3.1 SPI memory map.... 2334
24.3.2 Module Configuration Register (MCR).... 2335
24.3.2.1 Offset.... 2335
24.3.2.2 Function.... 2335
24.3.2.3 Diagram.... 2335
24.3.2.4 Fields.... 2335
24.3.3 Transfer Count Register (TCR).... 2338
24.3.3.1 Offset.... 2338
24.3.3.2 Function.... 2338
24.3.3.3 Diagram.... 2338
24.3.3.4 Fields.... 2338
24.3.4 Clock and Transfer Attributes Register (in Master mode) (CTAR0 - CTAR1).... 2339
24.3.4.1 Offset.... 2339
24.3.4.2 Function.... 2339
24.3.4.3 Diagram.... 2339
24.3.4.4 Fields.... 2340
24.3.5 Clock and Transfer Attributes Register (in Slave mode) (CTAR0_SLAVE).... 2343
24.3.5.1 Offset.... 2343
24.3.5.2 Function.... 2343

24.3.5.3 Diagram....2344
24.3.5.4 Fields....2344
24.3.6 Status Register (SR)....2345
24.3.6.1 Offset....2345
24.3.6.2 Function....2345
24.3.6.3 Diagram....2345
24.3.6.4 Fields....2346
24.3.7 DMA/Interrupt Request Select and Enable Register (RSER)....2348
24.3.7.1 Offset....2348
24.3.7.2 Function....2348
24.3.7.3 Diagram....2348
24.3.7.4 Fields....2349
24.3.8 PUSH TX FIFO Register In Master Mode (PUSHR)....2351
24.3.8.1 Offset....2351
24.3.8.2 Function....2351
24.3.8.3 Diagram....2351
24.3.8.4 Fields....2352
24.3.9 PUSH TX FIFO Register In Slave Mode (PUSHR_SLAVE)....2353
24.3.9.1 Offset....2353
24.3.9.2 Function....2353
24.3.9.3 Diagram....2354
24.3.9.4 Fields....2354
24.3.10 POP RX FIFO Register (POPR)....2354
24.3.10.1 Offset....2354
24.3.10.2 Function....2354
24.3.10.3 Diagram....2355
24.3.10.4 Fields....2355
24.3.11 Transmit FIFO Registers (TXFR0 - TXFR3)....2355
24.3.11.1 Offset....2355

24.3.11.2 Function 2355
24.3.11.3 Diagram 2356
24.3.11.4 Fields 2356
24.3.12 Receive FIFO Registers (RXFR0 - RXFR3) 2356
24.3.12.1 Offset 2356
24.3.12.2 Function 2357
24.3.12.3 Diagram 2357
24.3.12.4 Fields 2357
24.3.13 Clock and Transfer Attributes Register Extended (CTARE0 - CTARE1) 2357
24.3.13.1 Offset 2357
24.3.13.2 Function 2358
24.3.13.3 Diagram 2358
24.3.13.4 Fields 2358
24.3.14 Status Register Extended (SREX) 2359
24.3.14.1 Offset 2359
24.3.14.2 Function 2359
24.3.14.3 Diagram 2359
24.3.14.4 Fields 2360
24.4 Functional description 2360
24.4.1 Starting and stopping module transfers 2361
24.4.2 SPI configuration 2362
24.4.2.1 Master mode 2362
24.4.2.2 Slave mode 2363
24.4.2.3 FIFO disable operation 2363
24.4.2.4 TX FIFO buffering mechanism 2364
24.4.2.4.1 Filling the TX FIFO 2364
24.4.2.4.2 Draining the TX FIFO 2364
24.4.2.5 CMD FIFO buffering mechanism 2365
24.4.2.6 RX FIFO buffering mechanism 2366

24.4.2.6.1 Filling the RX FIFO....2366
24.4.2.6.2 Draining the RX FIFO....2366
24.4.3 Module baud rate and clock delay generation....2367
24.4.3.1 Baud rate generator....2367
24.4.3.2 PCS to SCK delay (tCSC)....2368
24.4.3.3 After SCK delay (tASC)....2368
24.4.3.4 Delay after transfer (tDT)....2368
24.4.4 Transfer formats....2369
24.4.4.1 Classic SPI transfer format (CPHA = 0)....2370
24.4.4.2 Classic SPI transfer format (CPHA = 1)....2371
24.4.4.3 Continuous selection format....2372
24.4.4.4 Fast continuous selection format....2374
24.4.5 Continuous SCK....2375
24.4.6 Slave mode operation constraints....2377
24.4.7 Parity generation and check....2377
24.4.7.1 Parity for SPI frames....2378
24.4.8 Interrupts/DMA requests....2378
24.4.8.1 EOQ interrupt request....2379
24.4.8.2 Transmit FIFO fill interrupt or DMA request....2379
24.4.8.3 CMD FIFO fill interrupt or DMA request....2380
24.4.8.4 Transmit FIFO invalid write interrupt request....2381
24.4.8.5 Transfer complete interrupt request....2381
24.4.8.6 Command transfer complete interrupt request....2381
24.4.8.7 Transmit FIFO underflow interrupt request....2381
24.4.8.8 Receive FIFO drain interrupt or DMA request....2381
24.4.8.9 Receive FIFO overflow interrupt request....2382
24.4.8.10 SPI frame parity error interrupt request....2382
24.4.9 Power saving features....2382
24.4.9.1 Stop mode (External Stop mode)....2382

24.4.9.2 Module Disable mode.... 2383

24.5 Initialization/application information.... 2383

24.5.1 How to manage queues.... 2383

24.5.2 Switching Master and Slave modes.... 2384

24.5.3 Initializing module in Master/Slave modes.... 2385

24.5.4 Baud rate settings.... 2385

24.5.5 Delay settings.... 2386

24.5.6 Calculation of FIFO pointer addresses.... 2386

24.5.6.1 Address calculation for the first-in entry and last-in entry in the TX FIFO.... 2387

24.5.6.2 Address calculation for the first-in entry and last-in entry in the CMD FIFO.... 2388

24.5.6.3 Address calculation for the first-in entry and last-in entry in the RX FIFO.... 2388

Chapter 25
Trust Zone Modules

25.1 Overview.... 2389

25.2 TrustZone as implemented on the chip.... 2389

25.3 TrustZone Address Space Controller (TZC-400).... 2390

25.3.1 Introduction.... 2390

25.3.1.1 About the TZC-400.... 2390

25.3.1.1.1 TZC-400 overview.... 2390

25.3.1.2 Features.... 2391

25.3.2 TZC400 register descriptions.... 2391

25.3.2.1 TZC400 memory map.... 2391

25.3.2.2 Build Configuration Register (BUILD_CONFIG).... 2393

25.3.2.2.1 Offset.... 2393

25.3.2.2.2 Diagram.... 2393

25.3.2.2.3 Fields.... 2394

25.3.2.3 Action Register (ACTION).... 2394

25.3.2.3.1 Offset.... 2394

25.3.2.3.2 Diagram.... 2394

25.3.2.3.3 Fields....2395
25.3.2.4 Gate Keeper Register (GATE_KEEPER)....2395
25.3.2.4.1 Offset....2395
25.3.2.4.2 Diagram....2395
25.3.2.4.3 Fields....2396
25.3.2.5 Speculation Control Register (SPECULATION_CTRL)....2396
25.3.2.5.1 Offset....2396
25.3.2.5.2 Diagram....2397
25.3.2.5.3 Fields....2397
25.3.2.6 Interrupt Status Register (INT_STATUS)....2397
25.3.2.6.1 Offset....2398
25.3.2.6.2 Diagram....2398
25.3.2.6.3 Fields....2398
25.3.2.7 Interrupt Clear Register (INT_CLEAR)....2399
25.3.2.7.1 Offset....2399
25.3.2.7.2 Diagram....2399
25.3.2.7.3 Fields....2399
25.3.2.8 Fail Address Low Register (FAIL_ADDRESS_LOW_0)....2400
25.3.2.8.1 Offset....2400
25.3.2.8.2 Diagram....2400
25.3.2.8.3 Fields....2400
25.3.2.9 Fail Address High Register (FAIL_ADDRESS_HIGH_0)....2401
25.3.2.9.1 Offset....2401
25.3.2.9.2 Diagram....2401
25.3.2.9.3 Fields....2401
25.3.2.10 Fail Control register (FAIL_CONTROL_0)....2401
25.3.2.10.1 Offset....2401
25.3.2.10.2 Diagram....2402
25.3.2.10.3 Fields....2402

Section number	Title	Page

25.3.2.11 Region 0 Base Address Low Register (REGION_BASE_LOW_0)..... 2403

25.3.2.11.1 Offset.....2403

25.3.2.11.2 Diagram.....2403

25.3.2.11.3 Fields.....2404

25.3.2.12 Region 0 Base Address High Register (REGION_BASE_HIGH_0).....2404

25.3.2.12.1 Offset.....2404

25.3.2.12.2 Diagram.....2404

25.3.2.12.3 Fields.....2404

25.3.2.13 Region 0 Top Address Low Register (REGION_TOP_LOW_0)..... 2405

25.3.2.13.1 Offset.....2405

25.3.2.13.2 Diagram.....2405

25.3.2.13.3 Fields.....2405

25.3.2.14 Region 0 Top Address High Register (REGION_TOP_HIGH_0).....2406

25.3.2.14.1 Offset.....2406

25.3.2.14.2 Diagram.....2406

25.3.2.14.3 Fields.....2406

25.3.2.15 Region 0 Attribute Register (REGION_ATTRIBUTES_0)..... 2406

25.3.2.15.1 Offset.....2407

25.3.2.15.2 Diagram.....2407

25.3.2.15.3 Fields.....2407

25.3.2.16 Region a Base Address Low Register (REGION_BASE_LOW_1 - REGION_BASE_LOW_8).....2408

25.3.2.16.1 Offset.....2408

25.3.2.16.2 Diagram.....2408

25.3.2.16.3 Fields.....2408

25.3.2.17 Region a Base Address High Register (REGION_BASE_HIGH_1 - REGION_BASE_HIGH_8)..... 2408

25.3.2.17.1 Offset.....2409

25.3.2.17.2 Diagram.....2409

25.3.2.17.3 Fields....2409
25.3.2.18 Region a Top Address Low Register (REGION_TOP_LOW_1 - REGION_TOP_LOW_8).2409
25.3.2.18.1 Offset....2409
25.3.2.18.2 Diagram....2410
25.3.2.18.3 Fields....2410
25.3.2.19 Region a Top Address High Register (REGION_TOP_HIGH_1 - REGION_TOP_HIGH_8)....2410
25.3.2.19.1 Offset....2410
25.3.2.19.2 Diagram....2411
25.3.2.19.3 Fields....2411
25.3.2.20 Region a Attribute Register (REGION_ATTRIBUTES_1 - REGION_ATTRIBUTES_8)... 2411
25.3.2.20.1 Offset....2411
25.3.2.20.2 Diagram....2411
25.3.2.20.3 Fields....2412
25.3.2.21 Peripheral Identification 0 register (PID0)....2412
25.3.2.21.1 Offset....2412
25.3.2.21.2 Function.... 2413
25.3.2.21.3 Diagram....2413
25.3.2.21.4 Fields....2413
25.3.2.22 Peripheral Identification 1 register (PID1)....2413
25.3.2.22.1 Offset....2413
25.3.2.22.2 Function.... 2413
25.3.2.22.3 Diagram....2414
25.3.2.22.4 Fields....2414
25.3.2.23 Peripheral Identification 2 register (PID2)....2414
25.3.2.23.1 Offset....2414
25.3.2.23.2 Function.... 2414
25.3.2.23.3 Diagram....2415
25.3.2.23.4 Fields....2415

25.3.2.24 Peripheral Identification 3 register (PID3)....2415
25.3.2.24.1 Offset....2415
25.3.2.24.2 Function....2416
25.3.2.24.3 Diagram....2416
25.3.2.24.4 Fields....2416
25.3.2.25 Component Identification 0 (CID0)....2416
25.3.2.25.1 Offset....2416
25.3.2.25.2 Function....2417
25.3.2.25.3 Diagram....2417
25.3.2.25.4 Fields....2417
25.3.2.26 Component Identification 1 (CID1)....2417
25.3.2.26.1 Offset....2417
25.3.2.26.2 Function....2418
25.3.2.26.3 Diagram....2418
25.3.2.26.4 Fields....2418
25.3.2.27 Component Identification 2 (CID2)....2418
25.3.2.27.1 Offset....2418
25.3.2.27.2 Function....2418
25.3.2.27.3 Diagram....2419
25.3.2.27.4 Fields....2419
25.3.2.28 Component Identification 3 (CID3)....2419
25.3.2.28.1 Offset....2419
25.3.2.28.2 Function....2419
25.3.2.28.3 Diagram....2420
25.3.2.28.4 Fields....2420
25.3.3 Functional Description....2420
25.3.3.1 TZC-400 operation....2420
25.3.3.1.1 Access filtering with regions....2421
25.3.3.1.1.1 Region access rules....2421

25.3.3.1.1.2 Example region setup.... 2422
25.3.3.1.1.3 Region setup.... 2423
25.3.3.1.1.4 Access permissions to regions.... 2423
25.3.3.1.2 Speculative accesses.... 2424
25.3.3.1.3 Fast path selection.... 2424
25.3.3.1.4 Gate keeper.... 2425
25.3.3.1.5 Barriers.... 2425
25.3.3.1.6 Exclusive accesses.... 2426
25.3.3.1.7 Denied ACE-Lite transactions.... 2426
25.3.3.1.8 Reset.... 2427
25.3.3.2 Constraints of use.... 2427
25.3.3.2.1 Changing TZC-400 region settings on an active system.... 2427
25.4 TrustZone Protection Controller (TZPC).... 2428
25.4.1 About the TrustZone Protection Controller.... 2428
25.4.2 TZPC Functional description.... 2429
25.4.3 TZPC register descriptions.... 2429
25.4.3.1 TZPC Peripheral Identification Registers 0-3.... 2429
25.4.3.2 Identification Registers 0-3.... 2430
25.4.3.3 TZPC memory map.... 2431
25.4.3.4 Secure RAM Region Size Register (TZPCR0SIZE).... 2431
25.4.3.4.1 Offset.... 2431
25.4.3.4.2 Function.... 2432
25.4.3.4.3 Diagram.... 2432
25.4.3.4.4 Fields.... 2432
25.4.3.5 TZDECPROT Register 0 Stat (TZDECPROT0_Stat).... 2433
25.4.3.5.1 Offset.... 2433
25.4.3.5.2 Function.... 2433
25.4.3.5.3 Diagram.... 2433
25.4.3.5.4 Fields.... 2433

Section number	Title	Page

25.4.3.6 TZDECPROT Register 0 Set (TZDECPROT0_Set) 2434

25.4.3.6.1 Offset 2434

25.4.3.6.2 Function 2434

25.4.3.6.3 Diagram 2434

25.4.3.6.4 Fields 2435

25.4.3.7 TZDECPROT Register 0 Clr (TZDECPROT0_Clr) 2436

25.4.3.7.1 Offset 2436

25.4.3.7.2 Function 2436

25.4.3.7.3 Diagram 2436

25.4.3.7.4 Fields 2436

25.4.3.8 TZDECPROT Register 1 Stat (TZDECPROT1_Stat) 2437

25.4.3.8.1 Offset 2437

25.4.3.8.2 Function 2437

25.4.3.8.3 Diagram 2437

25.4.3.8.4 Fields 2438

25.4.3.9 TZDECPROT Register 1 Set (TZDECPROT1_Set) 2438

25.4.3.9.1 Offset 2438

25.4.3.9.2 Function 2439

25.4.3.9.3 Diagram 2439

25.4.3.9.4 Fields 2439

25.4.3.10 TZDECPROT Register 1 Clr (TZDECPROT1_Clr) 2440

25.4.3.10.1 Offset 2440

25.4.3.10.2 Function 2440

25.4.3.10.3 Diagram 2440

25.4.3.10.4 Fields 2441

25.4.3.11 TZPC Peripheral Identification Register 0 (TZPCPERIPHID0) 2441

25.4.3.11.1 Offset 2441

25.4.3.11.2 Function 2441

25.4.3.11.3 Diagram 2441

25.4.3.11.4 Fields....2442
25.4.3.12 TZPC Peripheral Identification Register 1 (TZPCPERIPHID1).... 2442
25.4.3.12.1 Offset....2442
25.4.3.12.2 Function.... 2442
25.4.3.12.3 Diagram....2442
25.4.3.12.4 Fields....2443
25.4.3.13 TZPC Peripheral Identification Register 2 (TZPCPERIPHID2).... 2443
25.4.3.13.1 Offset....2443
25.4.3.13.2 Function.... 2443
25.4.3.13.3 Diagram....2443
25.4.3.13.4 Fields....2444
25.4.3.14 TZPC Peripheral Identification Register 3 (TZPCPERIPHID3).... 2444
25.4.3.14.1 Offset....2444
25.4.3.14.2 Function.... 2444
25.4.3.14.3 Diagram....2444
25.4.3.14.4 Fields....2445
25.4.3.15 TZPC Identification Register 0 (TZPCPCELLID0).... 2445
25.4.3.15.1 Offset....2445
25.4.3.15.2 Function.... 2445
25.4.3.15.3 Diagram....2445
25.4.3.15.4 Fields....2446
25.4.3.16 TZPC Identification Register 1 (TZPCPCELLID1).... 2446
25.4.3.16.1 Offset....2446
25.4.3.16.2 Function.... 2446
25.4.3.16.3 Diagram....2446
25.4.3.16.4 Fields....2447
25.4.3.17 TZPC Identification Register 2 (TZPCPCELLID2).... 2447
25.4.3.17.1 Offset....2447
25.4.3.17.2 Function.... 2447

25.4.3.17.3 Diagram....2447
25.4.3.17.4 Fields....2448
25.4.3.18 TZPC Identification Register 3 (TZPCPCELLID3)....2448
25.4.3.18.1 Offset....2448
25.4.3.18.2 Function....2448
25.4.3.18.3 Diagram....2448
25.4.3.18.4 Fields....2449
25.5 TrustZone Memory Adapter (TZMA)....2449
25.5.1 About the AXI TrustZone memory adapter....2449
25.5.2 Functional description....2450
25.5.2.1 Functional blocks....2450
25.5.2.1.1 Write channel router....2451
25.5.2.1.2 Read channel router....2451
25.5.2.1.3 Address comparator....2451
25.5.2.2 Interface attributes....2451

Chapter 26
Trust Architecture 3.0

26.1 Trust architecture objectives....2453
26.2 Characteristics and claims....2453
26.3 Non-claims....2455
26.4 Related resources....2456

Chapter 27
Thermal Monitoring Unit (TMU)

27.1 The TMU module as implemented on the chip....2459
27.1.1 Local temperature sensor placement....2459
27.1.2 Initialization Information....2460
27.2 Thermal Monitoring Unit Introduction....2462
27.2.1 TMU Overview....2462
27.2.2 Features....2463

Section number	Title	Page

27.2.3 Modes of Operation 2464

27.3 TMU register descriptions 2464

27.3.1 TMU memory map 2464

27.3.2 TMU mode register (TMR) 2465

27.3.2.1 Offset 2465

27.3.2.2 Function 2465

27.3.2.3 Diagram 2465

27.3.2.4 Fields 2466

27.3.3 TMU status register (TSR) 2466

27.3.3.1 Offset 2467

27.3.3.2 Function 2467

27.3.3.3 Diagram 2467

27.3.3.4 Fields 2467

27.3.4 TMU monitor temperature measurement interval register (TMTMIR) 2468

27.3.4.1 Offset 2468

27.3.4.2 Function 2468

27.3.4.3 Diagram 2468

27.3.4.4 Fields 2469

27.3.5 TMU interrupt enable register (TIER) 2470

27.3.5.1 Offset 2470

27.3.5.2 Function 2470

27.3.5.3 Diagram 2470

27.3.5.4 Fields 2470

27.3.6 TMU interrupt detect register (TIDR) 2471

27.3.6.1 Offset 2471

27.3.6.2 Function 2471

27.3.6.3 Diagram 2471

27.3.6.4 Fields 2472

27.3.7 TMU interrupt site capture register (TISCR) 2472

27.3.7.1 Offset....2472
27.3.7.2 Function.... 2472
27.3.7.3 Diagram....2472
27.3.7.4 Fields....2473
27.3.8 TMU interrupt critical site capture register (TICSCR)....2473
27.3.8.1 Offset....2473
27.3.8.2 Function.... 2473
27.3.8.3 Diagram....2474
27.3.8.4 Fields....2474
27.3.9 TMU monitor high temperature capture register (TMHTCR)....2474
27.3.9.1 Offset....2474
27.3.9.2 Function.... 2475
27.3.9.3 Diagram....2475
27.3.9.4 Fields....2475
27.3.10 TMU monitor low temperature capture register (TMLTCR).... 2475
27.3.10.1 Offset....2475
27.3.10.2 Function.... 2476
27.3.10.3 Diagram....2476
27.3.10.4 Fields....2476
27.3.11 TMU monitor high temperature immediate threshold register (TMHTITR)....2477
27.3.11.1 Offset....2477
27.3.11.2 Function.... 2477
27.3.11.3 Diagram....2477
27.3.11.4 Fields....2477
27.3.12 TMU monitor high temperature average threshold register (TMHTATR)....2478
27.3.12.1 Offset....2478
27.3.12.2 Function.... 2478
27.3.12.3 Diagram....2478
27.3.12.4 Fields....2478

27.3.13 TMU monitor high temperature average critical threshold register (TMHTACTR).....2479
27.3.13.1 Offset.....2479
27.3.13.2 Function.....2479
27.3.13.3 Diagram.....2479
27.3.13.4 Fields.....2480
27.3.14 TMU temperature configuration register (TTCFGR).....2480
27.3.14.1 Offset.....2480
27.3.14.2 Function.....2480
27.3.14.3 Diagram.....2481
27.3.14.4 Fields.....2481
27.3.15 TMU sensor configuration register (TSCFGR).....2481
27.3.15.1 Offset.....2481
27.3.15.2 Function.....2481
27.3.15.3 Diagram.....2482
27.3.15.4 Fields.....2482
27.3.16 TMU report immediate temperature site register a (TRITSR0 - TRITSR1).....2482
27.3.16.1 Offset.....2482
27.3.16.2 Function.....2483
27.3.16.3 Diagram.....2483
27.3.16.4 Fields.....2483
27.3.17 TMU report average temperature site register a (TRATSR0 - TRATSR1).....2483
27.3.17.1 Offset.....2483
27.3.17.2 Function.....2484
27.3.17.3 Diagram.....2484
27.3.17.4 Fields.....2484
27.3.18 TMU temperature range 0 control register (TTR0CR).....2484
27.3.18.1 Offset.....2485
27.3.18.2 Function.....2485
27.3.18.3 Diagram.....2485

27.3.18.4 Fields....2485
27.3.19 TMU temperature range 1 control register (TTR1CR)....2486
27.3.19.1 Offset....2486
27.3.19.2 Function....2486
27.3.19.3 Diagram....2487
27.3.19.4 Fields....2487
27.3.20 TMU temperature range 2 control register (TTR2CR)....2487
27.3.20.1 Offset....2488
27.3.20.2 Function....2488
27.3.20.3 Diagram....2488
27.3.20.4 Fields....2488
27.3.21 TMU temperature range 3 control register (TTR3CR)....2489
27.3.21.1 Offset....2489
27.3.21.2 Function....2489
27.3.21.3 Diagram....2490
27.3.21.4 Fields....2490
27.4 Functional Description....2491
27.4.1 Monitoring....2491
27.4.2 Reporting....2491

Chapter 28
Universal Serial Bus Interface 3.0

28.1 Overview....2493
28.1.1 Features....2493
28.1.2 Modes of Operation....2494
28.1.3 External Signals....2494
28.2 USB3.0 register descriptions....2495
28.2.1 USB3 memory map....2496
28.2.2 Capability registers length and HC interface version number (CAPLENGTH)....2499
28.2.2.1 Offset....2500

28.2.2.2 Diagram....2500
28.2.2.3 Fields....2500
28.2.3 Host controller structural parameters 1 (HCSPARAMS1)....2500
28.2.3.1 Offset....2500
28.2.3.2 Diagram....2501
28.2.3.3 Fields....2501
28.2.4 Host controller structural parameters 2 (HCSPARAMS2)....2501
28.2.4.1 Offset....2501
28.2.4.2 Diagram....2502
28.2.4.3 Fields....2502
28.2.5 Host controller structural parameters 3 (HCSPARAMS3)....2503
28.2.5.1 Offset....2503
28.2.5.2 Diagram....2503
28.2.5.3 Fields....2503
28.2.6 Host controller capability parameters 1 (HCCPARAMS1)....2504
28.2.6.1 Offset....2504
28.2.6.2 Diagram....2504
28.2.6.3 Fields....2504
28.2.7 Doorbell offset (DBOFF)....2505
28.2.7.1 Offset....2505
28.2.7.2 Diagram....2505
28.2.7.3 Fields....2506
28.2.8 Runtime register space offset (RTSOFF)....2506
28.2.8.1 Offset....2506
28.2.8.2 Diagram....2507
28.2.8.3 Fields....2507
28.2.9 Host controller capability parameters 2 (HCCPARAMS2)....2507
28.2.9.1 Offset....2507
28.2.9.2 Diagram....2507

28.2.9.3 Fields....2508
28.2.10 Global SoC bus configuration register 0 (GSBUSCFG0)....2508
28.2.10.1 Offset....2508
28.2.10.2 Function....2509
28.2.10.3 Diagram....2511
28.2.10.4 Fields....2512
28.2.11 Global SoC bus configuration register 1 (GSBUSCFG1)....2513
28.2.11.1 Offset....2513
28.2.11.2 Function....2514
28.2.11.3 Diagram....2514
28.2.11.4 Fields....2514
28.2.12 Global Tx threshold control register (GTXTHRCFG)....2515
28.2.12.1 Offset....2515
28.2.12.2 Function....2515
28.2.12.3 Diagram....2515
28.2.12.4 Fields....2516
28.2.13 Global Rx threshold control register (GRXTHRCFG)....2517
28.2.13.1 Offset....2517
28.2.13.2 Function....2517
28.2.13.3 Diagram....2517
28.2.13.4 Fields....2517
28.2.14 Global core control register (GCTL)....2518
28.2.14.1 Offset....2518
28.2.14.2 Diagram....2519
28.2.14.3 Fields....2519
28.2.15 Global status register (GSTS)....2522
28.2.15.1 Offset....2522
28.2.15.2 Diagram....2522
28.2.15.3 Fields....2522

Section number	Title	Page

28.2.16 Global user control register 1 (GUCTL1) 2523

28.2.16.1 Offset 2523

28.2.16.2 Diagram 2524

28.2.16.3 Fields 2524

28.2.17 Global user ID register (GUID) 2528

28.2.17.1 Offset 2528

28.2.17.2 Function 2528

28.2.17.3 Diagram 2528

28.2.17.4 Fields 2529

28.2.18 Global user control register (GUCTL) 2529

28.2.18.1 Offset 2529

28.2.18.2 Function 2529

28.2.18.3 Diagram 2529

28.2.18.4 Fields 2530

28.2.19 Global SoC bus error address register low (GBUSERRADDRLO) 2532

28.2.19.1 Offset 2532

28.2.19.2 Function 2533

28.2.19.3 Diagram 2533

28.2.19.4 Fields 2533

28.2.20 Global SoC bus error address register high (GBUSERRADDRHI) 2534

28.2.20.1 Offset 2534

28.2.20.2 Function 2534

28.2.20.3 Diagram 2534

28.2.20.4 Fields 2534

28.2.21 Global SS port to bus instance mapping register - low (GPRTBIMAPLO) 2534

28.2.21.1 Offset 2535

28.2.21.2 Function 2535

28.2.21.3 Diagram 2535

28.2.21.4 Fields 2535

Section number	Title	Page

28.2.22 Global SS port to bus instance mapping register - high (GPRTBIMAPHI) 2536

28.2.22.1 Offset 2536

28.2.22.2 Function 2536

28.2.22.3 Diagram 2536

28.2.22.4 Fields 2536

28.2.23 Global hardware parameters register 0 (GHWPARAMS0) 2537

28.2.23.1 Offset 2537

28.2.23.2 Diagram 2537

28.2.23.3 Fields 2537

28.2.24 Global hardware parameters register 1 (GHWPARAMS1) 2538

28.2.24.1 Offset 2538

28.2.24.2 Function 2538

28.2.24.3 Diagram 2539

28.2.24.4 Fields 2539

28.2.25 Global hardware parameters register 2 (GHWPARAMS2) 2541

28.2.25.1 Offset 2541

28.2.25.2 Diagram 2541

28.2.25.3 Fields 2541

28.2.26 Global hardware parameters register 3 (GHWPARAMS3) 2542

28.2.26.1 Offset 2542

28.2.26.2 Diagram 2542

28.2.26.3 Fields 2543

28.2.27 Global hardware parameters register 4 (GHWPARAMS4) 2544

28.2.27.1 Offset 2544

28.2.27.2 Diagram 2545

28.2.27.3 Fields 2546

28.2.28 Global hardware parameters register 5 (GHWPARAMS5) 2546

28.2.28.1 Offset 2546

28.2.28.2 Function 2547

28.2.28.3 Diagram....2547
28.2.28.4 Fields....2547
28.2.29 Global hardware parameters register 6 (GHWPARAMS6)....2548
28.2.29.1 Offset....2548
28.2.29.2 Diagram....2548
28.2.29.3 Fields....2549
28.2.30 Global hardware parameters register 7 (GHWPARAMS7)....2550
28.2.30.1 Offset....2550
28.2.30.2 Diagram....2550
28.2.30.3 Fields....2551
28.2.31 Global high-speed port to bus instance mapping register - low (GPRTBIMAP_HSLO)....2551
28.2.31.1 Offset....2551
28.2.31.2 Function....2551
28.2.31.3 Diagram....2552
28.2.31.4 Fields....2552
28.2.32 Global high-speed port to bus instance mapping register - high (GPRTBIMAP_HSHI)....2552
28.2.32.1 Offset....2552
28.2.32.2 Function....2552
28.2.32.3 Diagram....2553
28.2.32.4 Fields....2553
28.2.33 Global USB2 PHY configuration register (GUSB2PHYCFG)....2553
28.2.33.1 Offset....2553
28.2.33.2 Function....2553
28.2.33.3 Diagram....2554
28.2.33.4 Fields....2554
28.2.34 Global USB 3.0 PIPE control register (GUSB3PIPECTL)....2556
28.2.34.1 Offset....2556
28.2.34.2 Function....2556
28.2.34.3 Diagram....2556

Section number	Title	Page

28.2.34.4 Fields....2557

28.2.35 Global transmit FIFO size register (GTXFIFOSIZ_0 - GTXFIFOSIZ_3)....2558

28.2.35.1 Offset....2559

28.2.35.2 Function....2559

28.2.35.3 Diagram....2559

28.2.35.4 Fields....2560

28.2.36 Global receive FIFO size register (GRXFIFOSIZ_0 - GRXFIFOSIZ_2)....2560

28.2.36.1 Offset....2560

28.2.36.2 Function....2560

28.2.36.3 Diagram....2561

28.2.36.4 Fields....2561

28.2.37 Global event buffer address (low) register (GEVNTADRLO)....2561

28.2.37.1 Offset....2561

28.2.37.2 Function....2562

28.2.37.3 Diagram....2562

28.2.37.4 Fields....2562

28.2.38 Global event buffer address (high) register (GEVNTADRHI)....2562

28.2.38.1 Offset....2562

28.2.38.2 Function....2563

28.2.38.3 Diagram....2563

28.2.38.4 Fields....2563

28.2.39 Global event buffer size register (GEVNTSIZ)....2563

28.2.39.1 Offset....2563

28.2.39.2 Function....2563

28.2.39.3 Diagram....2564

28.2.39.4 Fields....2564

28.2.40 Global event buffer count register (GEVNTCOUNT)....2564

28.2.40.1 Offset....2565

28.2.40.2 Function....2565

28.2.40.3 Diagram....2565
28.2.40.4 Fields....2565
28.2.41 Global hardware parameters register 8 (GHWPARAMS8)....2566
28.2.41.1 Offset....2566
28.2.41.2 Diagram....2566
28.2.41.3 Fields....2566
28.2.42 Global device TXFIFO DMA priority register (GTXFIFOPRIDEV)....2567
28.2.42.1 Offset....2567
28.2.42.2 Function....2567
28.2.42.3 Diagram....2567
28.2.42.4 Fields....2568
28.2.43 Global host TXFIFO DMA priority register (GTXFIFOPRIHST)....2568
28.2.43.1 Offset....2568
28.2.43.2 Function....2568
28.2.43.3 Diagram....2569
28.2.43.4 Fields....2569
28.2.44 Global host RXFIFO DMA priority register (GRXFIFOPRIHST)....2570
28.2.44.1 Offset....2570
28.2.44.2 Function....2570
28.2.44.3 Diagram....2571
28.2.44.4 Fields....2571
28.2.45 Global host FIFO DMA high-low priority ratio register (GDMAHLRATIO)....2571
28.2.45.1 Offset....2571
28.2.45.2 Function....2572
28.2.45.3 Diagram....2572
28.2.45.4 Fields....2572
28.2.46 Global frame length adjustment register (GFLADJ)....2573
28.2.46.1 Offset....2573
28.2.46.2 Function....2573

28.2.46.3 Diagram....2573
28.2.46.4 Fields....2574
28.2.47 Device configuration register (DCFG)....2575
28.2.47.1 Offset....2575
28.2.47.2 Function....2575
28.2.47.3 Diagram....2575
28.2.47.4 Fields....2576
28.2.48 Device control register (DCTL)....2577
28.2.48.1 Offset....2577
28.2.48.2 Function....2577
28.2.48.3 Diagram....2578
28.2.48.4 Fields....2578
28.2.49 Device event enable register (DEVTEN)....2581
28.2.49.1 Offset....2581
28.2.49.2 Function....2582
28.2.49.3 Diagram....2582
28.2.49.4 Fields....2582
28.2.50 Device status register (DSTS)....2583
28.2.50.1 Offset....2583
28.2.50.2 Function....2583
28.2.50.3 Diagram....2584
28.2.50.4 Fields....2584
28.2.51 Device generic command parameter register (DGCMDPAR)....2586
28.2.51.1 Offset....2586
28.2.51.2 Function....2586
28.2.51.3 Diagram....2586
28.2.51.4 Fields....2587
28.2.52 Device generic command register (DGCMD)....2587
28.2.52.1 Offset....2587

28.2.52.2 Function.... 2587
28.2.52.3 Diagram....2589
28.2.52.4 Fields....2589
28.2.53 Device active USB endpoint enable register (DALEPENA)....2590
28.2.53.1 Offset....2590
28.2.53.2 Function.... 2590
28.2.53.3 Diagram....2591
28.2.53.4 Fields....2591
28.2.54 Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_0 - DEPCMDPAR2_7).... 2592
28.2.54.1 Offset....2592
28.2.54.2 Function.... 2592
28.2.54.3 Diagram....2592
28.2.54.4 Fields....2592
28.2.55 Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_0 - DEPCMDPAR1_7).... 2593
28.2.55.1 Offset....2593
28.2.55.2 Function.... 2593
28.2.55.3 Diagram....2593
28.2.55.4 Fields....2594
28.2.56 Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_0 - DEPCMDPAR0_7).... 2594
28.2.56.1 Offset....2594
28.2.56.2 Function.... 2594
28.2.56.3 Diagram....2595
28.2.56.4 Fields....2595
28.2.57 Device physical endpoint-n command register (DEPCMD_0 - DEPCMD_7)....2595
28.2.57.1 Offset....2595
28.2.57.2 Function.... 2596
28.2.57.3 Diagram....2596
28.2.57.4 Fields....2596
28.2.58 ADP configuration register (ADPCFG)....2598

28.2.58.1 Offset....2598
28.2.58.2 Diagram....2598
28.2.58.3 Fields....2598
28.2.59 ADP control register (ADPCTL)....2599
28.2.59.1 Offset....2599
28.2.59.2 Diagram....2599
28.2.59.3 Fields....2600
28.2.60 ADP event register (ADPEVT)....2600
28.2.60.1 Offset....2601
28.2.60.2 Function....2601
28.2.60.3 Diagram....2601
28.2.60.4 Fields....2601
28.2.61 ADP event enable register (ADPEVTEN)....2602
28.2.61.1 Offset....2602
28.2.61.2 Diagram....2602
28.2.61.3 Fields....2603
28.3 USB 3.0 Controller and PHY Configuration....2604
28.4 USB PHY SuperSpeed register descriptions....2604
28.4.1 USB_PHY_SS memory map....2604
28.4.2 SUP_IDCODE_LO (IP_IDCODE_LO)....2604
28.4.2.1 Offset....2604
28.4.2.2 Diagram....2605
28.4.2.3 Fields....2605
28.4.3 SUP_IDCODE_HI (SUP_IDCODE_HI)....2605
28.4.3.1 Offset....2605
28.4.3.2 Diagram....2605
28.4.3.3 Fields....2606
28.4.4 MPLL_LOOP_CTL (MPLL_LOOP_CTL)....2606
28.4.4.1 Offset....2606

28.4.4.2 Diagram....2606
28.4.4.3 Fields....2606
28.4.5 LANE0_RX_OVRD_IN_HI (LANE0_RX_OVRD_IN_HI)....2606
28.4.5.1 Offset....2607
28.4.5.2 Function....2607
28.4.5.3 Diagram....2607
28.4.5.4 Fields....2607
28.5 Functional Description....2608
28.5.1 System memory descriptor and data buffers....2608
28.5.2 Device descriptor structures....2609
28.5.2.1 Structures....2610
28.5.2.1.1 Normal (Control-Data/Bulk/Interrupt), Isochronous, and Status Transfer Request Block Structure....2613
28.5.2.1.2 Setup and Status TRB Structure....2614
28.5.2.1.3 Link TRB Structure....2615
28.5.2.1.4 Chaining buffers (CHN) and interrupt on completion (IOC) usage....2616
28.5.2.1.5 Interrupt on Short Packet (ISP) and Continue on Short Packet (CSP) Usage..2622
28.5.2.1.6 Example of Setting Up TRBs....2622
28.5.3 Device Programming Model....2626
28.5.3.1 Register Initialization....2626
28.5.3.1.1 Device Power-On or Soft Reset....2627
28.5.3.1.2 Initialization on USB reset....2628
28.5.3.1.3 Initialization on connect done....2629
28.5.3.1.4 Initialization on SetAddress request....2629
28.5.3.1.5 Initialization on SetConfiguration or SetInterface Request....2629
28.5.3.1.6 Alternate initialization on SetInterface request....2630
28.5.3.1.7 Initialization on Disconnect Event....2631
28.5.3.1.8 Device-initiated disconnect....2631
28.5.3.1.9 Reconnect after Device-Initiated Disconnect....2631

Section number	Title	Page

28.5.3.2 Operational Model 2631

28.5.3.2.1 USB and Physical Endpoints 2632

28.5.3.2.2 Event Buffers 2632

28.5.3.2.3 Transfer and Buffer Rules 2637

28.5.3.2.3.1 Number of TRBs Rule 2639

28.5.3.2.3.2 TRB Control Bit Rules 2639

28.5.3.2.3.3 Buffer size rules and zero-length packets 2640

28.5.3.2.3.4 Transfer setup recommendations 2641

28.5.3.2.4 Transfer Resource Usage and Transfer State 2643

28.5.3.2.5 Transfer Descriptions 2644

28.5.3.2.5.1 Non-Isochronous OUT Transfers 2645

28.5.3.2.5.2 Isochronous OUT Transfers 2646

28.5.3.2.5.3 Non-isochronous IN transfers 2646

28.5.3.2.5.4 Isochronous IN Transfers 2648

28.5.3.2.6 Handling ENDPOINT_HALT 2649

28.5.3.2.7 Handling L1 Event During a Transfer 2649

28.5.3.3 Isochronous Transfer Programming Model 2649

28.5.3.3.1 Definitions 2650

28.5.3.3.2 Endpoint configuration 2651

28.5.3.3.3 Transfer configuration 2651

28.5.3.3.4 Starting a Transfer 2652

28.5.3.3.5 Core Behavior During an Interval 2653

28.5.3.3.6 Checking interval status 2655

28.5.3.3.7 Adding intervals to a transfer 2656

28.5.3.3.8 Moderating Events 2657

28.5.3.3.9 Other Types of Isochronous Endpoints 2657

28.5.3.3.9.1 FIFO-based isochronous IN Endpoints 2657

28.5.3.3.9.2 FIFO-based isochronous OUT Endpoints 2659

28.5.3.3.10 End a Transfer 2659

28.5.3.4 Control transfer programming model.... 2659
28.5.3.4.1 Two-stage control transfer programming model.... 2661
28.5.3.4.2 Three-stage control transfer programming model.... 2661
28.5.3.4.3 Handling fewer requests than wLength.... 2662
28.5.3.4.4 Control OUT transfer examples.... 2663
28.5.3.4.5 Control IN transfer examples.... 2664
28.5.3.5 Stream handling in SuperSpeed.... 2665
28.5.3.5.1 Stream IDs and transfer resources.... 2666
28.5.3.5.2 Stream selection and stream programming model.... 2666
28.5.3.5.3 Data movement within a stream.... 2667
28.5.3.6 Low power operation.... 2668
28.5.3.6.1 Low power operation of USB.... 2668
28.5.3.6.2 Low power operation of core.... 2669
28.5.3.6.2.1 Clock-Gating Mode.... 2670
28.5.4 Host programming model.... 2670
28.5.4.1 Initializing host registers.... 2670
28.5.4.2 Host controller capability registers.... 2670
28.5.4.3 xHCI implementation details.... 2670
28.5.4.3.1 LHCRST behavior.... 2671
28.5.4.3.2 ENT requirements.... 2671
28.5.4.3.3 Behavior on babble error.... 2671
28.5.4.3.4 Max_exit_latency_too_large message.... 2671
28.5.5 Device physical endpoint-specific commands.... 2673
28.5.5.1 Command 1: Set endpoint configuration: DEPCFG.... 2673
28.5.5.2 Command 2: Set endpoint transfer resource configuration (DEPXFERCFG).... 2675
28.5.5.3 Command 3: Get endpoint state (DEPGETSTATE).... 2676
28.5.5.4 Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).... 2676
28.5.5.5 Command 6: Start transfer (DEPSTRTXFER).... 2677
28.5.5.6 Command 7: Update transfer (DEPUPDXFER).... 2678

Section number	Title	Page

28.5.5.7 Command 8: End transfer (DEPENDXFER) 2679

28.5.5.8 Command 9: Start new configuration (DEPSTARTCFG) 2680

28.5.6 OTG 2681

28.5.6.1 OTG 2.0 for USB 3.0 Functionality 2681

28.5.6.1.1 Core OTG functions 2681

28.5.6.1.1.1 HNP Polling and Enable 2682

28.5.6.1.2 ADP functions 2682

28.5.6.1.2.1 Internal ADP controller 2682

28.5.6.1.3 Software flow 2684

28.5.6.1.3.1 A-device activity concise flow 2685

28.5.6.1.3.2 B-device activity concise flow 2686

28.5.6.1.4 Programming model 2687

28.5.6.1.4.1 Initializing global registers 2687

28.5.6.1.4.2 Initializing host registers 2688

28.5.6.1.4.3 Initializing device registers 2688

28.5.6.1.4.4 Initializing OTG registers 2688

28.5.6.1.4.5 Programming flow for OTG in USB 3.0 2688

28.5.6.1.5 Common driver tasks 2690

28.5.6.1.6 A-device flow 2695

28.5.6.1.7 SRP detection by the core (Timeline for ADevSRPDetEvnt) 2701

28.5.6.1.8 VBUS turned ON by the core (Timeline for ADevBSessEndEvnt) 2701

28.5.6.1.9 Core entering A-Host in HS/FS mode (Timeline for ADevBHostEvnt) 2702

28.5.6.1.10 B-device flow 2702

28.5.6.1.11 Core entering b3_us_peripheral in B-Peripheral in HS/FS mode (Timeline for BDevSessVldDetEvnt) 2708

28.5.6.1.12 VBUS change detected on USB (Timeline for BDevVBusChngEvnt) 2708

28.5.6.1.13 Internal ADP controller logic 2708

28.5.7 Initialization/application information 2709

28.5.8 Power management overview 2709

28.5.8.1 Clock gating 2710
28.5.8.2 Hardware-Controlled LPM 2711
28.5.8.2.1 Special consideration for OTG 2711

Chapter 29
Watchdog Timer Unit

29.1 Watchdog unit 2713
29.2 Peripheral Identification Registers 2714
29.3 PrimeCell Identification Registers 2715
29.4 WDOG register descriptions 2716
29.4.1 WDOG memory map 2716
29.4.2 Watchdog Load Register (WDOGLOAD) 2717
29.4.2.1 Offset 2717
29.4.2.2 Function 2717
29.4.2.3 Diagram 2717
29.4.2.4 Fields 2718
29.4.3 Watchdog Value Register (WDOGVALUE) 2718
29.4.3.1 Offset 2718
29.4.3.2 Function 2718
29.4.3.3 Diagram 2718
29.4.3.4 Fields 2718
29.4.4 Watchdog Control Register (WDOGCONTROL) 2719
29.4.4.1 Offset 2719
29.4.4.2 Function 2719
29.4.4.3 Diagram 2719
29.4.4.4 Fields 2719
29.4.5 Watchdog Clear Interrupt Register (WDOGINTCLR) 2720
29.4.5.1 Offset 2720
29.4.5.2 Function 2720
29.4.5.3 Diagram 2720

29.4.5.4 Fields....2721
29.4.6 Watchdog Raw Interrupt Status Register (WDOGRIS)....2721
29.4.6.1 Offset....2721
29.4.6.2 Function....2721
29.4.6.3 Diagram....2722
29.4.6.4 Fields....2722
29.4.7 Watchdog Interrupt Status Register (WDOGMIS)....2722
29.4.7.1 Offset....2722
29.4.7.2 Function....2723
29.4.7.3 Diagram....2723
29.4.7.4 Fields....2723
29.4.8 Watchdog Lock Register (WDOGLOCK)....2724
29.4.8.1 Offset....2724
29.4.8.2 Function....2724
29.4.8.3 Diagram....2724
29.4.8.4 Fields....2725
29.4.9 Watchdog Integration Test Control Register (WDOGITCR)....2725
29.4.9.1 Offset....2725
29.4.9.2 Function....2726
29.4.9.3 Diagram....2726
29.4.9.4 Fields....2726
29.4.10 Watchdog Integration Test Output Set Register (WDOGITOP)....2726
29.4.10.1 Offset....2727
29.4.10.2 Function....2727
29.4.10.3 Diagram....2727
29.4.10.4 Fields....2727
29.4.11 Peripheral Identification Register 0 (WDOGPERIPHID0)....2728
29.4.11.1 Offset....2728
29.4.11.2 Function....2728

29.4.11.3 Diagram....2728
29.4.11.4 Fields....2729
29.4.12 Peripheral Identification Register 1 (WDOGPERIPHID1)....2729
29.4.12.1 Offset....2729
29.4.12.2 Function....2729
29.4.12.3 Diagram....2729
29.4.12.4 Fields....2730
29.4.13 Peripheral Identification Register 2 (WDOGPERIPHID2)....2730
29.4.13.1 Offset....2730
29.4.13.2 Function....2730
29.4.13.3 Diagram....2730
29.4.13.4 Fields....2731
29.4.14 Peripheral Identification Register 3 (WDOGPERIPHID3)....2731
29.4.14.1 Offset....2731
29.4.14.2 Function....2731
29.4.14.3 Diagram....2731
29.4.14.4 Fields....2732
29.4.15 PrimeCell Identification Register 0 (WDOGPCELLID0)....2732
29.4.15.1 Offset....2732
29.4.15.2 Function....2732
29.4.15.3 Diagram....2732
29.4.15.4 Fields....2733
29.4.16 PrimeCell Identification Register 1 (WDOGPCELLID1)....2733
29.4.16.1 Offset....2733
29.4.16.2 Function....2733
29.4.16.3 Diagram....2733
29.4.16.4 Fields....2734
29.4.17 PrimeCell Identification Register 2 (WDOGPCELLID2)....2734
29.4.17.1 Offset....2734

29.4.17.2 Function.... 2734
29.4.17.3 Diagram....2734
29.4.17.4 Fields....2735
29.4.18 PrimeCell Identification Register 3 (WDOGPCELLID3)....2735
29.4.18.1 Offset....2735
29.4.18.2 Function.... 2735
29.4.18.3 Diagram....2735
29.4.18.4 Fields....2736

Chapter 1
Overview

1.1 Introduction

The QorIQ LS1088A processor is built on NXP's software-aware, core-agnostic DPAA2 architecture, which delivers scalable acceleration elements sized for application needs, unprecedented efficiency, and smarter, more capable networks. When coupled with ease-of-use facilities such as real-time monitoring and debug, virtualization, and software management utilities, the available toolkits allow for both hardware and software engineers to bring a complete solution to market faster than ever.

The LS1088A integrated multicore processor combines eight Arm® Cortex®-A53 processor cores with high-performance data path acceleration logic, common network and peripheral bus interfaces, the LS1088A is well suited for networking, telecom/datacom, wireless infrastructure, storage and military/aerospace applications.

The LS1088A processor is supported by a consistent API that provides both basic and complex manipulation of the hardware peripherals in the device, releasing the developer from the classic programming challenges of interfacing with new peripherals at the hardware level.

1.2 Block diagram

The figure below shows the major functional units within the chip.

Figure 1-1. LS1088A Block Diagram

1.3 Features summary

This chip includes the following functions and features:

- Eight 32-bit / 64-bit Arm v8 Cortex-A53 CPUs
  - Arranged as two clusters of four cores sharing a 1 MB L2 cache
  - Up to 1.6 GHz
  - Single-threaded cores with 32 KB L1 data cache and 32 KB L1 instruction cache
- Cache coherent interconnect (CCI-400)
  - Hardware-managed data coherency
  - Up to 700 MHz
- One 64-bit DDR4 SDRAM memory controller with ECC
  - Up to 2.1 GT/s
  - Supports 64-bit, 32-bit, and 16-bit operation.
  - Support for x8 and x16 devices
  - Accumulated mode ECC for 32-bit, and 16-bit use
  - Supports DDR4 pseudo-open drain mode.
- Data path acceleration architecture 2.0 (DPAA2) based hardware / logic, accelerating the following functions:

  - Packet parsing, classification, and distribution with Wire Rate IO processor (WRIOP)
  - Queue management for scheduling, packet sequencing, and congestion management (QMan)
  - Hardware buffer management for buffer allocation and de-allocation (BMan)
  - Cryptography acceleration (SEC 5.2) at up to 10 Gbps
  - Advanced I/O processing (AIOP) programmable acceleration of complex IP Fowarding
  - qDMA engine
  - Eight SerDes lanes
    - Two lanes running at up to 10.3125 Gbps
  - Two SerDes blocks with four lanes each and two PLLs per SerDes lane
- Ethernet interfaces
  - Up to two five-speed Ethernet MACs 10 G, 1 G, 2.5 G, 100 M and 10 M support
  - Up to four three-speed Ethernet MACs 1 G, 100 M and 10 M support
  - 2.5 G support for the two quad-speed Ethernet MACs
  - Four of the three-speed MACs support MACsec.
  - Support for RGMII, SGMII (and 1000 Base-KX), SGMII2.5x, QSGMII
  - Support for XFI (and 10 GBase-KR)
  - Energy Efficient Ethernet support (IEEE802.3az)
  - IEEE1588 support
- High-speed peripheral interfaces
  - Three PCIe 3.0 controllers, one supporting SR-IOV
  - One serial ATA (SATA 3.0) controller
- Additional peripheral interfaces
  - Two high-speed USB 3.0 controllers with integrated PHY
  - Enhanced secure digital host controller (eSDHC/eMMC)
  - Serial peripheral interface (SPI) controller
  - Four I^2C controllers
  - Two DUARTs
  - Integrated flash controller (IFC 2.0) supporting NAND and NOR flash
  - One uQE supporting TDM/HLDC/UART
  - One quad serial peripheral interface (QSPI) controller
- Support for hardware virtualization and partitioning enforcement
- Implements QorIQ Layerscape Trust Architecture combined with Arm TrustZone®
  - Service processor (SP) provides pre-boot initialization and secure boot capabilities
- Single-source clocking solution enabling generation of Core, Platform, DDR, SerDes and USB clocks from a single external clock oscillator.
- Thermal Monitor Unit (TMU) with +/- 3° C accuracy

- Eight WatchDog timers, one per Cortex-A53 processor and one TrustZone WatchDog timer
- Four Flextimers and one generic timer
- Debug supporting run control, data acquisition, high-speed trace, and performance/event monitoring.

1.4 Core and CPU clusters

The chip offers eight 64-bit Arm® Cortex®-A53 cores. The Cortex-A53 CPUs are arranged in two clusters of four cores with a shared 1 MB L2 cache per cluster.

1.4.1 Cortex-A53 core features

The Cortex-A53 core subsystem includes the following features:

- Up to 1.6 GHz
- Full implementation of the Armv8-A architecture
- In-order pipeline with symmetric dual-issue of most instructions
- Advanced SIMD and floating-point extension for integer and floating-point vector operations
- Embedded trace macrocell (ETM) based on the ETMv4 architecture
- Performance monitor unit (PMU) support based on the PMUv3 architecture
- Cross trigger interface (CTI) for multiprocessor debugging
- Generic interrupt controller (GIC) CPU interface based on GICv4 architecture
- Support for power management with multiple power domains to control both dynamic and static power dissipation
- 40-bit physical addressing
- Memory management unit
    - 10-entry fully-associative instruction micro TLB
    - 10-entry fully-associative data micro TLB
    - 4-way set-associative, 512-entry unified main TLB
- L1 caches, running at core frequency
    - 32 KB instruction, 2-way set-associative
        - Single-bit parity protection
    - 32 KB data, 4-way set-associative
        - ECC protection
    - Hardware page table walk
    - 64-byte cache line size

Details of the banked L2 are provided below.

- 1 MB cache with ECC protection (data, tag, & status)
  - Operates at same frequency as the CPUs in the cluster
- Snoop control unit (SCU)- L2 cache stashing
- 64-byte cache line size
- 16 way, set associative

1.5 Memory complex

The chip's memory complex consists of the DDR controller for main memory, and the memory controllers associated with the integrated flash controller (IFC).

The chip also includes a 1 MB Packet express buffer (PEB) memory that is used by the WRIOP for low-latency packet storage.

1.5.1 DDR memory controller

The chip offers one 64-bit DDR controller supporting ECC protected memories. The 64-bit DDR controller operate at up to 2.133 GT/s for DDR4. Some key DDR controller features are as follows:

- The 64-bit DDR controller supports x8, and x16 memory widths
  - Programmable support for single, dual, and quad ranked devices and modules
  - Support for both unbuffered and registered DIMMs
  - 4 chip-selects
  - 40-bit address support, up to 1 TB memory
- The chip can be configured to retain the currently active SDRAM page for pipelined burst accesses. Page mode support of up to 64 simultaneously open pages can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, page mode can save up to ten memory clock cycles for subsequent burst accesses that hit in an active page.
- Using ECC, the chip detects and corrects all single-bit errors and detects all double-bit errors and all errors within a nibble.
- Upon detection of a loss of power signal from external logic, the DDR controller can put compliant DDR SDRAM DIMMs into self-refresh mode, allowing systems to implement battery-backed main memory protection.
- In addition, the DDR controller offers an initialization bypass feature for use by system designers to prevent re-initialization of main memory during system restore from exiting self refresh.
- Support active zeroization of system memory upon detection of a user-defined security violation.

1.6 Nonvolatile memory interfaces

1.6.1 Integrated flash controller (IFC) 2.0

The chip incorporates an integrated flash controller (IFC) similar to the one used in some previous generation QorIQ chips. The IFC supports both NAND and NOR flash, as well as a general purpose memory mapped interface for connecting low speed ASICs and FPGAs.

1.6.1.1 NAND flash features

- x8 NAND flash interface
- Support for ONFi-3.1 asynchronous interface (8 Bit)/NVDDR interface up to mode 4 and mandatory commands
- Support for toggle 2.0 interface (8 bit) and mandatory commands.
- Optional ECC generation/checking
- Flexible timing control to allow interfacing with proprietary NAND devices
- SLC and MLC flash devices support with configurable page sizes of up to 16 KB
- Support advance NAND commands like cache, copy-back, and multiplane programming
- Internal SRAM of 18 KB
    - Memory-mapped
    - Can be configured as boot RAM
    - Acts as a data buffer for normal operation
- Boot chip-select (CS0) available after system reset, with boot block size of 16 KB, for boot loading from NAND flash (in asynchronous mode)

1.6.1.2 NOR flash features

- Data bus width of 16-bit
- Compatible with asynchronous/synchronous NOR flash
- Directly memory mapped
- Supports address data multiplexed (ADM) NOR device
- Flexible timing control allows interfacing with proprietary NOR devices
- Boot chip-select (CS0) available at system reset
- Programmable burst size control

1.6.1.3 General-purpose chip-select machine (GPCM)

The IFC's GPCM supports the following features:

- Normal GPCM
  - Support for x8/16-bit device
  - Compatible with general purpose addressable device, for example, SRAM and ROM
  - External clock is supported with programmable division ratio (2, 3, 4, and so on, up to 16)
- Generic ASIC interface
  - Support for x8/16-bit device
  - Address and data are shared on I/O bus
  - Following address and data sequences are supported on I/O bus:
    - 16-bit I/O: AADD
    - 8-bit I/O: AAAADDDD
  - Configurable even/odd parity on address/data bus supported
  - Parity error detection supported

1.6.2 Enhanced secure digital host controller (eSDHC)

The chip supports SD/eSDXC/eMMC card interfaces. The eSDHC (SD/eSDXC/eMMC) controller includes a DMA engine, allowing it to move data from flash cards to external or internal memory following straightforward initiation by software.

Detailed features of the eSDHC (SD/eSDXC/eMMC) controller include the following:

- Conforms to the *SD Host Controller Standard Specification version 3.0*
- Compatible with the *eMMC System Specification version*
- Compatible with the *SD Memory Card Physical Layer Specification version 3.01*
- Compatible with the *SDIO Card Specification version 3.0*
- Designed to work with SD memory, SDIO, SD combo, eMMC, and variants like mini, micro, embedded, and so on
- Card bus clock frequency up to 208 MHz
- Supports SD UHS-1 speed modes: SDR12, SDR25, SDR50, SDR104, and DDR50
- Supports eMMC, HS200 and DDR modes
- Programmable burst size control

1.6.3 Serial peripheral interface (SPI)

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master mode
- Slave mode
- Data streaming operation in slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) entries
- Support for 8/16-bit accesses to the PUSH TX FIFO Register Data Field
- Buffered receive operation using the receive FIFO (RX FIFO) entries
- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues
- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
    - Two transfer attribute registers
    - Two extended transfer attribute registers
    - Serial clock (SCK) with programmable polarity and phase
    - Various programmable delays
    - Programmable serial frame size: 4 to 16 bits
        - SPI frames longer than 16 bits can be supported using the continuous selection format.
    - Continuously held chip select capability
    - Parity control

1.6.4 Quad serial peripheral interface (QuadSPI)

The QuadSPI supports the following features:

- Flexible sequence engine to support various flash vendor devices.
- Single, dual, quad and octal mode of operation.
- DDR/DTR mode wherein the data is generated on every edge of the serial flash clock.
- Support for flash data strobe signal for data sampling in DDR and SDR mode.
- Two identical serial flash devices can be connected and accessed in parallel for data read operations, forming one (virtual) flash memory with doubled readout bandwidth.
- Multimaster accesses with priority
    - Flexible and configurable buffer for each master
- 13 interrupt conditions

- Memory mapped read access to connected flash devices.
- Programmable sequence engine to cater to future command/protocol changes and able to support all existing vendor commands and operations.
    - Supports 3-byte and 4-byte addressing.

1.7 Universal serial bus (USB) 3.0

The two Universal Serial Bus (USB) 3.0 controllers with integrated PHY provide point-to-point connectivity that complies with the USB specification, Rev. 3.0. Each of the USB controllers with integrated PHY can be configured to operate as a stand-alone host, a stand-alone device, or with both host and device functions operating simultaneously.

Key features of the USB 3.0 controller include the following:

- Complies with USB specification, Rev. 3.0
- Supports super-speed (5 Gbps), high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) operations
- Each of the controllers supports operation as a stand-alone USB host controller
    - Supports USB root hub with one downstream-facing port
    - eXtensible Host Controller Interface (xHCI)-compatible
- Each of the controllers supports operation as a stand-alone USB device
    - Supports one upstream-facing port
    - Supports six programmable USB endpoints
- Each of the controllers supports OTG mode
- USB can support to 8 endpoints

The host and device functions are both configured to support all four USB transfer types:

- Bulk
- Control
- Interrupt
- Isochronous

1.8 QUICC Engine

Key features of QUICC Engine-Lite include the following:

- Single 32-bit RISC controller for flexible support of communication peripherals
- Serial DMA channel for reception and transmission on all serial channels
- Two UCCs supporting the following interfaces (not all of them simultaneously):
    - Serial

- ISDN
- UART
- Asynchronous HDLC (256 Channels) (bit rate up to 2 Mbit/s)
- TDM interfaces supporting up to 128 QUICC multichannel controller channels

1.9 High-speed peripheral interface complex (HSSI)

The chip offers a variety of high-speed serial interfaces (HSSI), sharing a set of 8 SerDes lanes. Each interface is backed by a high speed serial interface controller. The chip has the following types of controllers:

- Three PCI Express controllers, one supporting SR-IOV
- SATA 3.0 Controller
- Up to ten Ethernet controllers with various protocols

1.9.1 PCI Express

Each of the chip's three PCI Express controllers is compatible with the *PCI Express Base Specification Revision 3.0*. Key features of each PCI Express controller include the following:

- Power-on reset configuration options allow root complex or endpoint functionality.
- The physical layer operates at 2.5, 5, or 8 Gbaud data rate per lane.
- x4, x2, and x1 link widths supported
    - One controller supports x4, x2, x1
    - One controller supports x2, x1 only
    - One controller supports x1 only
- Both 32-bit and 64-bit addressing and 256-byte maximum payload size
- Inbound INTx transactions
- Locality specific peripheral interrupts (LPI)
- Message Signaled Interrupt (MSI) transactions
- PCI Express controller 1 (PCIe1) supports end-point SR-IOV
    - Two physical functions (PF0 and PF1)
    - 16 virtual functions per physical function (VF1-VF16)
    - Eight MSI-X per virtual function
- Hot-plug support; hot-swap with external component.

1.9.2 Serial Advanced Technology Attachment (SATA)

Serial Advanced Technology Attachment (SATA) controller is compliant with the *Serial ATA 3.0 Specification*. SATA controller has the following features:

- Supports speeds: 1.5 Gbps (first-generation SATA), 3 Gbps (second-generation SATA), and 6 Gbps (third-generation SATA)
- Contains high-speed descriptor-based DMA controller
- Supports native command queuing (NCQ) commands
- Supports port multiplier operation
- Supports hot plug including asynchronous signal recovery

1.10 Data Path Acceleration Architecture (DPAA2)

Data Path Acceleration Architecture (DPAA2) is at the heart of the chip and is designed to deliver the next generation of network processing acceleration. The DPAA2 supports simplified and secure sharing of programmable networking resources under multiple virtual environments. In addition to the features listed in the Features summary section, the DPAA2 includes acceleration engines and a security co-processor.

DPAA2 provides a management framework that allows for dynamic service provisioning while being easy to use. Software developers are able to use DPAA2 hardware peripherals through a consistent API framework. The APIs allow for full control over these peripherals, releasing the developer from the classic programming challenges. Full programmability is provided through straightforward embedded C-based structured programming with tools for simple configuration, and run-time maintenance.

1.10.1 DPAA2 management complex

For ease of use of the DPAA2, NXP introduces the management complex (MC). The MC organizes the interface to the physical world by composing networking functions as collections of software networking objects.

NXP has also developed a "toolbox" of DPAA2 object services. These services abstract the DPAA2 hardware into software-managed objects that are:

- Simpler to use than directly managing the hardware
- Application-oriented in terminology and use
- Based on concepts generally familiar to programmers and system architects

The DPAA2 object services are provided by firmware running on the DPAA2 hardware. Customers do not need to program the hardware directly in order to use it; they simply use the DPAA2 object services and the NXP-supplied and tested firmware. These two things combine into the management complex (MC).

With the use of the MC, users are able to develop complex networking tasks without detailed knowledge of the DPAA2 hardware; for example :

- DPAA2 resource allocation, tracking and recovery in fault scenarios
- Device discovery and configuration management
- Virtualization of ports and very low-speed devices.

The MC is a trusted entity, and only executes trusted firmware. It is isolated from the rest of the chip so it cannot be compromised by malicious software running on the GPPs.

1.10.2 Other DPAA2 components

The Wire Rate I/O Processor (WRIOP) is an interface between the external world and the DPAA2. The WRIOP receives datagrams via Ethernet and queues them to other DPAA2 entities, as well as de-queues datagrams from other DPAA2 entities for transmission. The AIOP is a highly efficient programmable engine targeted at packet processing applications.The SEC and content accelerators de-queue processing requests (typically from software) and en-queue results to the configured next consumer.

1.10.2.1 Wire rate IO processor (WRIOP)

WRIOP is a packet processing engine that receives and transmits packets from and to Ethernet interfaces. The WRIOP combines Ethernet MACs with packet parsing and classification logic to provide intelligent distribution and queuing decisions for incoming traffic. The WRIOP is capable of processing up to 28 Gbps (2x10G + 8x1G).

The WRIOP performs two flows: ingress flow and egress flow. The ingress flow receives packets from the MAC interfaces, runs parse, classify, and police, and drops the packet or allocates buffers from BMan and enqueues the packet to the QMan. The egress flow dequeues packets from the QMan, and transmits them to the MAC interface. Buffers are released back to BMan. Optionally, it is possible to configure the WRIOP to enqueue the transmitted packets back to a queue for confirmation. The WRIOP flows are fixed, with some limited configurability mainly related to memory buffering features.

The WRIOP includes the following features:

- Support for up to 13 ports including physical ports and recycled path ports

- Ten physical ports, one management and two recycle ports.
  - Two ports capable of operating at 10/2.5/1 Gbps,100/10 Mbps
  - Eight ports capable of operating at 1 Gbps /100 Mbps/10 Mbps
- Up to 10 Ethernet MAC network interfaces
  - MDIO for each Ethernet controller as part of the Ethernet MAC
  - Dedicated MDIO for PHY management
  - Optional MACsec support for four of the eight1Gbps capable MACs
- Provide burst tolerance in system where the speed of the physical links is higher than the average sustained rate of the device
- Split the traffic in a way that allows for 'intelligent' QoS (back pressure and/or discard)
- Detect errors, discard or isolate the frames if needed
- IEEE Std. 802.1Qbb (Priority-based flow control) for up to eight queues/priorities
- 64 WRIOP interface profiles (IFPs)
- WRIOP management commands
- Statistics per port or per interface profile
- Error reporting through QMan and interrupts
- Interfaces with the queue manager (QMan)
  - Dequeues/enqueues frame descriptors (FD) to/from the QMan
  - Supports drop on tail-drop/WRED per the QMan response
  - Supports transmit pause frame due to queue congestion state
- Interfaces with the buffer manager (BMan)
  - Allocates/deallocates frame buffers
- Ethernet MAC features
  - IEEE Std. 802.3ae (full MAC layer and reconciliation sub-layer implementation)
  - Energy efficient Ethernet (IEEE Std. 802.3az)
  - Eight MAC address comparison on receive and one MAC address overwrite on transmit for NIC applications.
  - Ethernet pause frame (802.3 Annex 31A) termination providing fully automated flow control without any user application overhead
  - Priority flow control (PFC) frame support allowing 8 classes for higher layer congestion management
  - Magic packet detection
  - 802.3 basic and mandatory managed objects statistic counters and IETF management
  - Information database (MIB) package (RFC2665) and remote network monitoring (RMON) counters
  - Programmable clause 22 and clause 45 MDIO master interface for PHY device configuration and management
- The Ethernet MAC controller IEEE 1588 timestamp
  - Precise time-stamping of ingress frames and egress frames

1.10.2.1.1 Classifier and table lookup unit (CTLU)

Frames in the ingress flow are subject to classification through the WRIOP classifier. The WRIOP classifier is capable of performing a series of functions upon a stream of frames. It is possible to skip the classification and use values from the initial interface profile for the processing of the frame.

The classifier and table lookup unit (CTLU) is composed of the following main components:

- Portable classifier
- QoS mapping and policer

The portable classifier is used to parse the frame headers, generate a key, and perform basic table lookup by performing an exact match on the key.

Parsing consists of examining the structure of a frame (first 256 bytes) according to its inherent knowledge of standard protocols in order to determine attributes of the frame and extract network values. Soft paring instruction may be added to extend the protocols supported and/or extract additional attributes or network values.

Lookups can be defined uniquely for the particular interface profile being selected. One frame attribute lookup and multiple value-based (that is, lookup key) lookups can be performed. Each value-based lookup is preceded by a new key composition. All lookup tables reside in internal memory on the device.

The primary purpose of executing lookups is to determine the "direction" for a frame. Different elements of direction for the same frame may have resulted from different lookups because entries of a particular table may be configured to make assertion about only a subset of the direction elements. The elements of the "direction" are placed in the frame control value (FCV). The WRIOP uses the values in the FCV to determine the action being taken on the packet. The output of the classifier in its simplest form is the queuing destination ID (QDID), which is used by QMan to enqueue the frame to a specific queue (FQID). In the WRIOP, the classifier also selects the final interface profile (thus determining the storage profile) and is able to override individual fields in the action descriptor (destination QDID, OMB bits, and so on).

Other functions that are performed by the CTLU are:

- A rate-based policer that can apply a policer profile as directed by classification. The policer color that is applied to the frames is used by the QMan WRED.

- A QoS mapping function that associates a normalized priority and color to a frame based on direction from classify that specifies value or mapping method. Frame may be tagged or altered, and values travel with frame.
- A "distribution bin" determination function that produces a bin value by hashing a set of values from the frame. The composition of the distribution hash key is directed by the outcome of classification. Note that the actual distribution of frames is performed here. It can be done within QMan or elsewhere, and it uses the bin value determined here as the basis.

Frames processed by the egress flow go through some classification functions. The egress CTLU is used for the following purposes:

- Place the checksum in the outgoing frame (parser)
- Determine the Spanning Tree Protocol State (STP) of the outgoing port (table lookup)
- Determine VLAN manipulation mode

1.10.2.1.1.1 Table lookup unit (TLU) features

- Key composition
  - Up to 124 bytes key for hash lookup
  - Count composition rule hit
- Internal lookup tables
  - Longest prefix match (LPM)
  - Tables in internal memory
- Hash value calculation for frame queue distribution

1.10.2.1.1.2 QoS mapping and policer features

- QoS and initial color determination
  - Derive a 4-bit priority value and a 2-bit color value for the frame
  - Determination method options
    - Based on bits of VLAN priority (three bits)
    - Based on IP TOS (six bits)
    - Fixed priority based on a classification result
    - Use of conversion tables to convert from VLAN priority/TOS/QMan TC to frame priority and color
- Policer
  - Supports RFC-2698: A two-rate three-color marker
  - Supports RFC-4115: A differentiated service two-rate three-color marker with efficient handling of in-profile traffic
  - Work in either color-blind or color-aware mode

- Support one policing action per frame
- 256 policer profiles

1.10.2.1.1.3 Parser and attributes select features

- Parsing
  - Done on the first 256 bytes of the frame
  - Parsing and validation of packet headers with hard coded parser functions for the known and stable protocols
  - Hard-wired parsing capabilities can be supplemented with user programmed parse functions to support protocols not supported by the hard-wired functionality including proprietary protocols and "shim" headers inserted between otherwise well known protocols
  - Can start parsing from any header/layer, in any offset, from the beginning of the frame
- Frame attribute lookup
  - Key for attribute lookup
    - Frame attributes result from parsing
    - Initial IFPID
    - Isolation Context Identifier ICID
- Checksum update (IPv4, TCP/UDP L4)

1.10.2.2 Advanced IO processor (AIOP)

The AIOP is a C-programmable engine that enables power efficient, packet-oriented processing by using a single-threaded per flow model; this model allows for the simple use of the AIOP's tightly coupled accelerators. It performs packet processing slower than wire rate with software parse-classify-distribute (PCD).

The AIOP includes capabilities for the following:

- Acceleration for firewall functions
- Acceleration for L3 IP-forwarding (IPv4/ v6)
- Acceleration for L3 IP-sec
- L3 autonomous forwarding
- Acceleration for network address translation (NAT)
- Acceleration for network address and port translation (NAPT)
- Robust header compression (ROHC)
- Netflow/IPFix offload
- ARP Response offload
- NetFlow flow-based statistics (based on sampling) with stateful flows on the order of millions

- ACL lookups (for source-based routing and firewall)
- Unicast routing protocol
- Link aggregation
- Layer 3+ unicast and multicast routing protocols

1.10.2.2.1 AIOP features

Features of the AIOP include:

- C-programmable tasks entry point defined by hardware classification
- Acceleration for commonly used functions in packet processing
  - Table lookup unit (exact match and wildcard matches) with parse engine targeting internal SRAM and external DRAM
  - Statistics acceleration (fire and forget increments)
  - DMA (with optional semaphore lock/check)
  - Timer unit
  - Reassembly accelerator
- Accelerator proxy interfaces for traditional lookaside accelerators
  - AIOP accelerator proxy AAS manages accelerator interface internal to AIOP and is the conduit for jobs/data into and out of the external accelerator
- Standard core based debug support
- Nexus tracing to trace buffer and chip tracing interface
- Hardware infrastructure components
  - Order scope manager that implements order preservation and order restoration semantics
  - Task initialization/creation
  - Frame presentation DMA
    - Creates initial frame presentation into the workspace RAM, and provides capability for software to have additional presentation of packet data (either source packet or other packets stored in a frame description format)
  - Task termination/enqueue output DMA
    - Provides the task termination/cleanup and enqueue to QMan along with several other packet and task capabilities
  - QMan interface
  - BMan interface
  - Core task scheduler that manages the scheduling of tasks to the core based upon arbitration policy and ordering requirements
  - Accelerator scheduler that schedules tasks to accelerators based upon arbitration policy and ordering requirements
  - Workspace RAM

    - Tightly coupled RAM that contains various task state (packet header, stack frame, and so on).
  - Partition manager
    - Responsible for bandwidth sharing of the occupancy time in AIOP based upon isolation context

1.10.2.3 Queue manager (QMan)

The queue manager (QMan) is the primary infrastructure component in the DPAA2, allowing for simplified sharing of network interfaces and hardware accelerators by multiple CPU cores. It also provides a simple and consistent message and data passing mechanism for dividing processing tasks among multiple vCPUs.

The queue manager offers the following features:

- Common interface between software and all hardware
  - Controls the prioritized queuing of data between multiple processor cores, network interfaces, and hardware accelerators.
  - Supports both dedicated and pool channels, allowing both push and pull models of multicore load spreading.
- Atomic access to common queues without software locking overhead
- Mechanisms to guarantee order preservation with atomicity and order restoration following parallel processing on multiple CPUs
- Egress queuing for Ethernet interfaces
  - Hierarchical (2-level) scheduling and dual-rate shaping
  - Dual-rate shaping to meet service-level agreements (SLAs) parameters
  - Configurable combinations of strict priority and fair scheduling (weighted queuing) between the queues
  - Algorithms for shaping and fair scheduling are based on bytes
- Queuing to cores and accelerators
  - Two level queuing hierarchy with one or more channels per endpoint, eight work queues per channel, and numerous frame queues per work queue
  - Priority and work conserving fair scheduling between the work queues and the frame queues
- Loss-less flow control for ingress network interfaces
- Congestion avoidance (RED/WRED) and congestion management with tail discard

1.10.2.4 Buffer manager (BMan)

The buffer manager (BMan) manages pools of buffers on behalf of software for both hardware (accelerators and network interfaces) and software use.

The buffer manager offers the following features:

- Common interface for software and hardware
- Guarantees atomic access to shared buffer pools
- Supports 64 buffer pools
  - Software, hardware buffer consumers can request different size buffers and buffers in different memory partitions
- Supports depletion thresholds with congestion notifications
- On-chip per pool buffer stockpile to minimize access to memory for buffer pool management
- LIFO (last in first out) buffer allocation policy
  - Optimizes cache usage and allocation
  - A released buffer is immediately used for receiving new data

1.10.2.5 SEC

The SEC can perform full protocol processing for the following security protocols:

- IPsec
- SSL/TLS
- 3GPP RLC encryption/decryption
- LTE PDCP
- SRTP
- IEEE 802.1AE MACSec
- IEEE 802.16e WiMax MAC layer

The SEC supports the following :

- Public key hardware accelerators (PKHA)
  - RSA and Diffie-Hellman (to 4096b)
  - Elliptic curve cryptography (1023b)
- Data encryption standard accelerators (DESA)
  - DES, 3DES (2-key, 3-key)
  - ECB, CBC, OFB, and CFB modes
- Advanced encryption standard accelerators (AESA)
  - Key lengths of 128-bit, 192-bit, and 256-bit
  - Confidentiality modes
    - ECB, CBC, OFB, CFB, CTR and XTS
  - Authenticated encryption modes
    - CCM and GCM
- Message digest hardware accelerators (MDHA)
  - SHA-1, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256 digests

  - MD5 128-bit digest
  - HMAC with all algorithms
- Kasumi/F8 hardware accelerators (KFHA)
  - F8, F9 as required for 3GPP
  - A5/3 for GSM and EDGE, GEA-3 for GPRS
- Snow 3G hardware accelerators (STHA)
  - Implements Snow 3.0, F8 and F9 modes
- ZUC hardware accelerators (ZHA)
  - Implements 128-EEA3 & 128-EIA3
- CRC unit
  - Standard and user-defined polynomials
- Random number generator (RNG)
  - Incorporates TRNG entropy generator for seeding and deterministic engine (SHA-256)
  - Supports random IV generation
- DTLS
- IEEE Std 802.11 Wi-Fi

1.10.2.6 Queue direct memory access (qDMA)

The queue direct memory access (qDMA) controller transfers blocks of data between one source and one or more destinations. The qDMA supports the following features:

- Frame descriptor prefetch
- Channel virtualization
- Transaction based destination stride and source stride
- Destination and source address hold function
- Transaction based cache attribute setup
- Transaction based QoS
- Destination based throttle control
- DMA transfers with misaligned addresses and any transaction size
- Global DMA bandwidth control where transaction generated by the DMA is stalled when the system bandwidth consumption threshold allocated to the DMA has been reached
- Source/destination data scatter/gather through S/G table located in system memory
- Interrupt messaging

1.11 Advanced power management

Power dissipation is always a major design consideration in embedded applications; system designers need to balance the desire for maximum compute and IO density against single-chip and board-level thermal limits.

Junction temperature is a critical factor in comparing embedded processor specifications. NXP specs max power at 105°C junction, standard for commercial, embedded operating conditions. Not all multicore chips adhere to a 105°C junction for specifying worst case power.

The chip implements a number of software transparent and performance transparent power management features. Nontransparent power management features are also available, allowing for significant reductions in power consumption when the chip is under lighter loads; however, non-transparent power savings are not assumed in chip power specifications.

1.11.1 Transparent power management

The chip's commitment to low power begins with the decision to fabricate the chip in 28 nm bulk CMOS technology. This process technology offers low leakage, reducing both static and dynamic power. Whereas 28 nm offers inherent power savings, transistor leakage varies from lot to lot and device to device. Leakier parts are capable of faster transistor switching, but they also consume more power. By running devices from the leakier end of the process spectrum at less than nominal voltage, the system can achieve the required operating frequency within the specified max power. During manufacturing, NXP determines the voltage required to achieve the target frequency bin and program this Voltage ID into each device, so that initialization software can program the system's voltage regulator to the appropriate value.

Dynamic power is further reduced through fine-grained clock control. Many components and sub components in the chip automatically sleep (turn off their clocks) when they are not actively processing data. Such modules can return to full operating frequency on the clock cycle after work is dispatched to them. A portion of these dynamic power savings are built into the chip max power specification on the basis of impossibility of all processing elements and interfaces in the chip switching concurrently. The percent switching factors are considered quite conservative, and measured typical power consumption on QorIQ chips is well below the maximum in the data sheet.

As noted in WRIOP section, the chip supports energy-efficient Ethernet. During periods of extended inactivity on the transmit side, the chip transparently sends a low power idle (LPI) signal to the external PHY, effectively telling it to sleep.

Additional power savings can be achieved by users statically disabling unused components. Developers can turn off the clocks to individual logic modules (including CPUs) within the chip that the system isn't using. Based on a finite number of SerDes, any given application is expected to have some inactive Ethernet MACs or PCI Express controllers. Re-enabling clocks to a logic module generally requires a chip reset, which makes this type of power management infrequent (effectively static) and transparent to runtime software.

1.11.2 Arm power management

The Arm cores cluster support low power initiated by Arm ISA and Arm dynamic power management logic.

The table below summarizes the Arm cores cluster low power modes.

Table 1-1. Arm thread/core/cluster low power modes

Power management mode	Comments
CPU WFI	When the Arm CPU is in WFI state, the clocks in the CPU can dynamically and temporarily enabled to service snoops or debug resources access.
CPU State Retention Power Gating (SRPG)	When the Arm CPU is in WFI state (not recommended for WFE because that is a short low power state), the external A53 power controller can put the CPU into a retention state via the CPU Q-channel interface.
CPU power down	When the Arm CPU is in WFI state (not recommended for WFE because that is a short low power state), the CPU can enter power down based on the setting of CPUECTLR[6] - SMPEN.
L2 WFI	When the Arm processor is in L2 WFI state, all internal clocks in the processor are gated except for the PCLKDBG.

1.12 Resource partitioning and QorIQ trust architecture

Consolidation of discrete CPUs into a single, multicore chip and potential repartitioning of legacy software on those cores introduces many opportunities for unintended resource contentions to arise, but the QorIQ trust architecture can reduce the risk of these issues.

A system may exhibit erratic behavior if the multiple CPUs do not effectively partition and share system resources. While it can be challenging to prevent unintended resource contention, stopping malicious software is much more difficult. Device consolidation, combined with a trend toward embedded systems becoming more open (or more likely to run third-party or open-source software on at least one of the cores) creates opportunities for malicious code to enter a system.

The chip offers a new level of hardware partitioning support, allowing system developers to ensure software running on any CPU only accesses the resources (memory, peripherals, and so on) that it is explicitly authorized to access.

1.12.1 System memory management unit (SMMU)

MMU-based access control works for software running on CPUs; however, these are not the only bus masters in the chip. Internal components with bus mastering capability (WRIOP, PCI Express controller, SEC, and so on) also need to be prevented from reading and writing to certain memory regions. These components do not spontaneously generate access attempts; however, if programmed to do so by malicious or glitch in software, any of them could read or write sensitive data registers and crash the system. For this reason, the chip also includes a distributed function referred to as the system memory management unit (SMMU).

The SMMU provides address translation and access control mechanisms for transactions initiated by hardware components, using the same two-stage translation provided by the Arm processor MMU for software components. Using SMMU, a hypervisor allocates bus master components to guest operating systems, and guest operating systems in turn can allocate those hardware components directly to applications.

In addition, the SMMU supports Arm TrustZone®, allowing a secure operating system to partition bus master components between secure and non-secure worlds.

SMMU access control is based on the Access Management Qualifiers (AMQ) advertised by a bus master for a given transaction. Access Management Qualifiers (AMQ) can be static or they can be dynamic, based on the ID of the CPU that programmed the initiator.

1.12.2 Secure boot and sensitive data protection

The core MMUs and SMMU allow the chip to enforce a consistent set of memory access permissions on a per-partition basis. When combined with an embedded hypervisor for safe sharing of resources, the chip becomes highly resilient to poorly tested or malicious code. For system developers building high reliability/high security platforms, rigorous testing of code of known origin is the norm.

For this reason, the chip offers a secure boot option, in which the system developer digitally signs the code to be executed by the CPUs, and the chip ensures that only an unaltered version of that code runs on the platform. The chip offers both boot time and run time code authenticity checking, with configurable consequences when the authenticity check fails. The chip also supports protected internal and external storage of developer-provisioned sensitive instructions and data. For example, a system developer may provision each system with a number of RSA private keys to be used in mutual authentication and key exchange. These values would initially be stored as encrypted blobs in external non-volatile memory; but, following secure boot, these values can be decrypted into on-chip protected memory (portion of platform cache dedicated as SRAM). Session keys, which may number in the thousands to tens of thousands, are not good candidates for on-chip storage, so the chip offers session key encryption. Session keys are stored in main memory, and are decrypted (transparently to software and without impacting SEC throughput) as these are brought into the SEC for decryption of session traffic.

1.12.3 Service processor

The service processor complex supports following features:

- TrustZone-capable processor
- Secure boot starting point for power-on reset and hard reset
    - Authenticates PBI during pre-boot phase
    - Authenticates ESBC during boot phase
- Supports TrustZone trusted execution environment at run-time for system-wide trust functions like a soft TPM
- Supports TrustZone debug authorization at run-time

1.13 Debug support

Processing on a multicore chip with shared caches and peripherals leads to greater concurrency and an increased potential for unintended interactions between device components. To ensure that software developers have the same or better visibility into the device as they would with multiple discrete communications processors, NXP developed an Advanced Multicore Debug Architecture.

The debugging and performance monitoring capability enabled by the device hardware coexists within a debug ecosystem that offers a rich variety of tools at different levels of the hardware/software stack. Software development and debug tools from NXP (CodeWarrior), as well as third-party vendors, provide a rich set of options for configuring, controlling, and analyzing debug and performance related events.

Chapter 2
Memory Map

2.1 Memory Map Overview

There are several address domains within the device, including the following:

- Logical, virtual, and physical (real) address spaces within the Arm Architecture core(s)
- Internal configuration, control, and status register (CCSR) address space, which is a special-purpose subset of the internal local address space
- Internal debug control and status register (DCSR) address space, which is another special-purpose set of registers mapped in the internal local address space
- External memory, I/O, and configuration address spaces of the PCI Express links

The MMU in the core handles translation of logical (effective) addresses, into virtual addresses, and ultimately to the physical addresses for the local address space. The local address map refers to the physical address space seen by the core as it accesses memory and I/O space.

2.2 System Memory Map

Start address	End address	Size	Allocation	Comment
0x0000_0000_0000	0x0000_000F_FFFF	1MB	CCSR - Boot ROM	64KB
0x0000_0010_0000	0x0000_00FF_FFFF	15MB	Reserved	
0x0000_0100_0000	0x0000_0FFF_FFFF	240MB	CCSR	
0x0000_1000_0000	0x0000_10FF_FFFF	16MB	Reserved	
0x0000_1100_0000	0x0000_11FF_FFFF	16MB	Reserved	
0x0000_1200_0000	0x0000_13FF_FFFF	32MB	Reserved	
0x0000_1400_0000	0x0000_17FF_FFFF	64MB	Reserved	
0x0000_1800_0000	0x0000_181F_FFFF	2MB	OCRAM	128KB
0x0000_1820_0000	0x0000_18FF_FFFF	14MB	Reserved	

Table continues on the next page...

0x0000_1900_0000	0x0000_19FF_FFFF	16MB	CoreSight STM	16MB
0x0000_1A00_0000	0x0000_1BFF_FFFF	32MB	Reserved	
0x0000_1C00_0000	0x0000_1CFF_FFFF	16MB	Reserved	
0x0000_1D00_0000	0x0000_1FFF_FFFF	48MB	Reserved	
0x0000_2000_0000	0x0000_2FFF_FFFF	256MB	Quad SPI Region #1 (0-256MB)	More QSPI space below 256MB
0x0000_3000_0000	0x0000_3FFF_FFFF	256MB	IFC Region #1 (0-256MB)	More IFC space below 256MB
0x0000_4000_0000	0x0000_5FFF_FFFF	512MB	Reserved	
0x0000_6000_0000	0x0000_7FFF_FFFF	512MB	Reserved	
0x0000_8000_0000	0x0000_9FFF_FFFF	512MB	GPP DRAM Region #1 (0-2GB)	
0c0000_A000_0000	0x0000_BFFF_FFFF	512MB		
0x0000_C000_0000	0x0000_DFFF_FFFF	512MB		
0x0000_E000_0000	0x0000_FFFF_FFFF	512MB		
0x0001_0000_0000	0x0001_FFFF_FFFF	4GB	Reserved	
0x0002_0000_0000	0x0003_FFFF_FFFF	8GB		
0x0004_0000_0000	0x0004_0FFF_FFFF	256MB	Hole	QSPI space #1 maps on top of this space
0x0004_1000_0000	0x0004_FFFF_FFFF	3.75GB	Quad SPI Region #2 (256MB-4GB)	3.75GB
0x0005_0000_0000	0x0005_0FFF_FFFF	256MB	Hole	IFC space #1 maps on top of this space
0x0005_1000_0000	0x0005_FFFF_FFFF	3.75GB	IFC Region #2 (256MB-4GB)	3.75GB
0x0006_0000_0000	0x0006_FFFF_FFFF	4GB	Reserved	
0x0007_0000_0000	0x0007_3FFF_FFFF	1GB	DCSR	
0x0007_4000_0000	0x0007_FFFF_FFFF	3GB	Reserved	
DPAA2 Portal Map				
0x0008_0000_0000	0x0008_03FF_FFFF	64MB	Reserved	
0x0008_0400_0000	0x0008_07FF_FFFF	64MB	Reserved	
0x0008_0800_0000	0x0008_0BFF_FFFF	64MB	Reserved	
0x0008_0C00_0000	0x0008_0FFF_FFFF	64MB	MC - 1024 portals	32MB (512 portal)
0x0008_1000_0000	0x0008_17FF_FFFF	128MB	Reserved	
0x0008_1800_0000	0x0008_1FFF_FFFF	128MB	QBMAN portals	128MB
0x0008_0000_0000	0x000B_FFFF_FFFF	15.5GB	Reserved	
0x000C_0000_0000	0x000F_FFFF_FFFF	16GB	Reserved	
High-speed I/O (PCIe)				
0x0010_0000_0000	0x0011_FFFF_FFFF	8GB	Reserved	
0x0012_0000_0000	0x0013_FFFF_FFFF	8GB	Reserved	
0x0014_0000_0000	0x0015_FFFF_FFFF	8GB	Reserved	
0x0016_0000_0000	0x0017_FFFF_FFFF	8GB	Reserved	
0x0018_0000_0000	0x0019_FFFF_FFFF	8GB	Reserved	
0x001A_0000_0000	0x001B_FFFF_FFFF	8GB	Reserved	

Table continues on the next page...

0x001C_0000_0000	0x001D_FFFF_FFFF	8GB	Reserved	
0x001E_0000_0000	0x001F_FFFF_FFFF	8GB	Reserved	
0x0020_0000_0000	0x0027_FFFF_FFFF	32GB	PCIe1	
0x0028_0000_0000	0x002F_FFFF_FFFF	32GB	PCIe2	
0x0030_0000_0000	0x0037_FFFF_FFFF	32GB	PCIe3	
0x0038_0000_0000	0x003F_FFFF_FFFF	32GB	Reserved	
DPAA2 External address map				
0x0040_0000_0000	0x0040_FFFF_FFFF	4GB	Reserved	
0x0041_0000_0000	0x0041_FFFF_FFFF	4GB	Reserved	
0x0042_0000_0000	0x0042_FFFF_FFFF	4GB	Reserved	
0x0043_0000_0000	0x0043_FFFF_FFFF	4GB	WRIOP access window	
0x0044_0000_0000	0x0047_FFFF_FFFF	16GB	Reserved	
0x0048_0000_0000	0x0048_FFFF_FFFF	4GB	Reserved	
0x0049_0000_0000	0x0049_FFFF_FFFF	4GB	Reserved	
0x004A_0000_0000	0x004A_FFFF_FFFF	4GB	Reserved	
0x004B_0000_0000	0x004B_FFFF_FFFF	4GB	AIOP access window	
0x004C_0000_0000	0x004F_FFFF_FFFF	16GB	Packet Express Buffer	1MB
0x0050_0000_0000	0x005F_FFFF_FFFF	64GB	Reserved	
0x0060_0000_0000	0x007F_FFFF_FFFF	128GB	Reserved	
0x0080_0000_0000	0x0080_7FFF_FFF	2GB	Hole	
0x0080_8000_0000	0x00FF_FFFF_FFFF	510GB	GPP DRAM Region #2 (2-512GB)	

2.3 CCSR Address Map

The following table lists the absolute block base address assigned to each block.

Table 2-1. CCSR Block Base Address Map

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x100_0000 - 0x107_FFFF	Reserved	-
0x108_0000 - 0x108_FFFF	DDR memory controller	-
0x109_0000 - 0x109_FFFF	Reserved	-
0x10A_0000 - 0x111_FFFF	Reserved	
0x110_0000 - 0x110_FFFF	Trust Zone Address Space Controller (TZASC)	
0x111_0000 - 0x111_FFFF	Reserved	
0x120_0000 - 0x120_FFFF	Initiator NoC Error Reporting	Error reporting blocks

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x121_0000 - 0x121_FFFF	Target NoC Error Reporting	
0x122_0000 - 0x122_FFFF	HSIO NoC Error Reporting	
0x123_0000 - 0x123_FFFF	Memory Complex NoC Error Reporting	
0x124_0000 - 0x124_FFFF	DPAA2 NoC Error Reporting	
0x125_0000 - 0x125_FFFF	ACP NoC Error Reporting	
0x126_0000 - 0x126_FFFF	CCSR NoC Error Reporting	CCSR TrustZone security violations
0x127_0000 - 0x12F_FFFF	Reserved	
0x130_0000 - 0x130_FFFF	Clock Generation Unit (CGU-CGA)	
0x131_0000 - 0x131_FFFF	Reserved	
0x132_0000 - 0x135_FFFF	Reserved	
0x136_0000 - 0x136_FFFF	Platform Clock Generation Unit (CGU-P)	
0x137_0000 - 0x137_FFFF	Clocking Registers	
0x138_0000 - 0x138_FFFF	DDR Clock Generation Unit (D-CGU)	
0x139_0000 - 0x139_FFFF	Reserved	
0x13A_0000 - 0x13B_FFFF	Reserved	
0x13C_0000 - 0x13C_FFFF	Reserved	
0x13D_0000 - 0x1DF_FFFF	Reserved	
0x1E0_0000 - 0x1E0_FFFF	DCFG (Device Configuration)	
0x1E1_0000 - 0x1E2_FFFF	Reserved	
0x1E3_0000 - 0x1E3_FFFF	Power Management Unit	
0x1E4_0000 - 0x1E5_FFFF	Reserved	
0x1E6_0000 - 0x1E6_FFFF	Reset	
0x1E7_0000 - 0x1E7_FFFF	Reserved	
0x1E8_0000 - 0x1E8_FFFF	SFP (Security Fuse Processor)	
0x1E9_0000 - 0x1E9_FFFF	Security Monitor	
0x1EA_0000 - 0x1EA_FFFF	SerDes 1	
0x1EB_0000 - 0x1EB_FFFF	SerDes 2	
0x1EC_0000 - 0x1EF_FFFF	Reserved	
0x1F0_0000 - 0x1F0_FFFF	Service Processor	
0x1F1_0000 - 0x1F6_FFFF	Reserved	
0x1F7_0000 - 0x1F7_FFFF	ISC (Interrupt Sampling Control)	
0x1F8_0000 - 0x1F8_FFFF	TMU (Thermal Monitor Unit)	
0x1F9_0000 - 0x1FB_FFFF	Reserved	
0x1FC_0000 - 0x1FC_FFFF	Reserved	
0x1FD_0000 - 0x1FE_FFFF	Reserved	
0x1FF_0000 - 0x1FF_7FFF	Pin Control	
0x1FF_8000 - 0x1FF_FFFF	Pin Analog Control	

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x200_0000 - 0x200_FFFF	I2C 1	
0x201_0000 - 0x201_FFFF	I2C 2	
0x202_0000 - 0x202_FFFF	I2C 3	
0x203_0000 - 0x203_FFFF	I2C 4	
0x204_0000 - 0x20B_FFFF	Reserved	
0x20C_0000 - 0x20C_FFFF	QSPI (Quad SPI)	
0x20D_0000 - 0x20F_FFFF	Reserved	
0x210_0000 - 0x210_FFFF	SPI	
0x211_0000 - 0x213_FFFF	Reserved	
0x214_0000 - 0x214_FFFF	eSDHC	
0x215_0000 - 0x21B_FFFF	Reserved	
0x21C_0000 - 0x21C_FFFF	DUART 1	UART1 0x21C_0500 UART2 0x21C_0600
0x21D_0000 - 0x21D_FFFF	DUART 2	UART3 0x21D_0500 UART4 0x21D_0600
0x21E_0000 - 0x21F_FFFF	Reserved	
0x220_0000 - 0x220_FFFF	OCRAM TZPC	
0x221_0000 - 0x223_FFFF	Reserved	
0x224_0000 - 0x224_FFFF	IFC Global	IFC requires 2 pages. Internally IFC has two pages of 4KB. Total 8KB sequential addresses on single CCSR port. NOC will map to the 0x224_0000 to first 4KB (IFC Global) and 0x225_0000 to second 4KB (IFC Run-time)
0x225_0000 - 0x225_FFFF	IFC Run-time	
0x226_0000 - 0x22F_FFFF	Reserved	
0x230_0000 - 0x230_FFFF	GPIO 1	GPIO block only supports up to 32 I/Os
0x231_0000 - 0x231_FFFF	GPIO 2	
0x232_0000 - 0x232_FFFF	GPIO 3	
0x233_0000 - 0x233_FFFF	GPIO 4	
0x234_0000 - 0x23B_FFFF	Reserved	
0x23C_0000 - 0x23C_FFFF	TrustZone Watchdog Timer	
0x23D_0000 - 0x23D_FFFF	Reserved	
0x23E_0000 - 0x23E_FFFF	Global Generic Reference Timer - control (CNTControlBase)	This is the timebase for all of the SoC GPPs (see Section I1 of the Armv8 Architecture Ref. manual)
0x23F_0000 - 0x23F_FFFF	Global Generic Reference Timer - status (CNTReadBase)	This is the timebase for all of the SoC GPPs (see Section I1 of the Armv8 Architecture Ref. manual)

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x240_0000 - 0x27F_FFFF	HDLC(uQE)	
0x280_0000 - 0x280_FFFF	FlexTimer 1	
0x281_0000 - 0x281_FFFF	FlexTimer 2	
0x282_0000 - 0x282_FFFF	FlexTimer 3	
0x283_0000 - 0x283_FFFF	FlexTimer 4	
0x284_0000 - 0x30F_FFFF	Reserved	
0x310_0000 - 0x310_FFFF	USB 1	
0x311_0000 - 0x311_FFFF	USB 2	
0x312_0000 - 0x312_FFFF	Reserved	
0x313_0000 - 0x313_FFFF	Reserved	
0x314_0000 - 0x314_FFFF	Reserved	
0x315_0000 - 0x31F_FFFF	Reserved	
0x320_0000 - 0x320_FFFF	SATA Controller	
0x321_0000 - 0x321_FFFF	Reserved	
0x322_0000 - 0x33F_FFFF	Reserved	
0x340_0000 - 0x340_FFFF	PCI-Express Controller 1 PF0/shared	First 4kB is used for all PCI-Express operation modes; the remaining are used for SR-IOV mode only
0x341_0000 - 0x341_FFFF	Reserved	
0x342_0000 - 0x342_FFFF	Reserved	
0x343_0000 - 0x343_FFFF	Reserved	
0x344_0000 - 0x344_FFFF	Reserved	
0x345_0000 - 0x345_FFFF	Reserved	
0x346_0000 - 0x346_FFFF	Reserved	
0x347_0000 - 0x347_FFFF	Reserved	
0x348_0000 - 0x348_FFFF	PCI-Express Controller 1 Lookup table	
0x349_0000_0x34B_FFFF	Reserved	
0x34C_0000_0x34C_FFFF	PCI-Express Controller 1 PF0 controls	
0x34D_0000 - 0x34F_FFFF	Reserved	
0x350_0000 - 0x350_FFFF	PCI-Express Controller 2 PF0/shared	
0x351_0000 - 0x351_FFFF	Reserved	
0x352_0000 - 0x352_FFFF		
0x353_0000 - 0x353_FFFF	Reserved	
0x354_0000 - 0x354_FFFF	Reserved	
0x355_0000 - 0x355_FFFF	Reserved	
0x356_0000 - 0x356_FFFF	Reserved	
0x357_0000 - 0x357_FFFF	Reserved	

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x358_0000 - 0x358_FFFF	PCI-Express Controller 2 Look up table	
0x359_0000 - 0x35B_FFFF	Reserved	
0x35C_0000 - 0x35C_FFFF	PCI-Express Controller 2 PF0 controls	
0x35D_0000 - 0x35F_FFFF	Reserved	
0x360_0000 - 0x360_FFFF	PCI-Express Controller 3 PF0/shared	
0x361_0000 - 0x361_FFFF	Reserved	
0x362_0000 - 0x362_FFFF	Reserved	
0x363_0000 - 0x363_FFFF	Reserved	
0x364_0000 - 0x364_FFFF	Reserved	
0x365_0000 - 0x365_FFFF	Reserved	
0x366_0000 - 0x366_FFFF	Reserved	
0x367_0000 - 0x367_FFFF	Reserved	
0x368_0000 - 0x368_FFFF	PCI-Express Controller 3 Lookup table	
0x369_0000 - 0x36B_FFFF	Reserved	
0x36C_0000 - 0x36C_FFFF	PCI-Express Controller 3 PF0 controls	
0x36D_0000 - 0x378_FFFF	Reserved	
0x379_0000 - 0x408_FFFF	Reserved	
0x409_0000 - 0x409_FFFF	CCI-400 Configuration	
0x40A_0000 - 0x4FF_FFFF	Reserved	
0x500_0000 - 0x500_FFFF	MMU500 Global Register Space 0	SMMU GR0_BASE
0x501_0000 - 0x501_FFFF	MMU500 Global Register Space 1	SMMU_GR1_BASE
0x502_0000 - 0x502_FFFF	Reserved	
0x503_0000 - 0x503_FFFF	PerfMon Space	SMMU_PM_BASE
0x504_0000 - 0x504_FFFF	Reserved	
0x505_0000 - 0x53F_FFFF	Reserved	
0x540_0000 - 0x540_FFFF	Translation Context Bank 0	SMMU_TOP (32 contexts) SMMU_GLOBAL_TOP (64 contexts) SMMU_CB_BASE (64 contexts)
0x541_0000 - 0x541_FFFF	Translation Context Bank 1	
0x542_0000 - 0x542_FFFF	Translation Context Bank 2	
0x543_0000 - 0x543_FFFF	Translation Context Bank 3	
0x544_0000 - 0x544_FFFF	Translation Context Bank 4	
0x545_0000 - 0x545_FFFF	Translation Context Bank 5	
0x546_0000 - 0x546_FFFF	Translation Context Bank 6	
0x547_0000 - 0x547_FFFF	Translation Context Bank 7	
0x548_0000 - 0x548_FFFF	Translation Context Bank 8	
0x549_0000 - 0x549_FFFF	Translation Context Bank 9	

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x54A_0000 - 0x54A_FFFF	Translation Context Bank 10	
0x54B_0000 - 0x54B_FFFF	Translation Context Bank 11	
0x54C_0000 - 0x54C_FFFF	Translation Context Bank 12	
0x54D_0000 - 0x54D_FFFF	Translation Context Bank 13	
0x54E_0000 - 0x54E_FFFF	Translation Context Bank 14	
0x54F_0000 - 0x54F_FFFF	Translation Context Bank 15	
0x550_0000 - 0x550_FFFF	Translation Context Bank 16	
0x551_0000 - 0x551_FFFF	Translation Context Bank 17	
0x552_0000 - 0x552_FFFF	Translation Context Bank 18	
0x553_0000 - 0x553_FFFF	Translation Context Bank 19	
0x554_0000 - 0x554_FFFF	Translation Context Bank 20	
0x555_0000 - 0x555_FFFF	Translation Context Bank 21	
0x556_0000 - 0x556_FFFF	Translation Context Bank 22	
0x557_0000 - 0x557_FFFF	Translation Context Bank 23	
0x558_0000 - 0x558_FFFF	Translation Context Bank 24	
0x559_0000 - 0x559_FFFF	Translation Context Bank 25	
0x55A_0000 - 0x55A_FFFF	Translation Context Bank 26	
0x55B_0000 - 0x55B_FFFF	Translation Context Bank 27	
0x55C_0000 - 0x55C_FFFF	Translation Context Bank 28	
0x55D_0000 - 0x55D_FFFF	Translation Context Bank 29	
0x55E_0000 - 0x55E_FFFF	Translation Context Bank 30	
0x55F_0000 - 0x55F_FFFF	Translation Context Bank 31	
0x560_0000 - 0x560_FFFF	Translation Context Bank 32	
0x561_0000 - 0x561_FFFF	Translation Context Bank 33	
0x562_0000 - 0x562_FFFF	Translation Context Bank 34	
0x563_0000 - 0x563_FFFF	Translation Context Bank 35	
0x564_0000 - 0x564_FFFF	Translation Context Bank 36	
0x565_0000 - 0x565_FFFF	Translation Context Bank 37	
0x566_0000 - 0x566_FFFF	Translation Context Bank 38	
0x567_0000 - 0x567_FFFF	Translation Context Bank 39	
0x568_0000 - 0x568_FFFF	Translation Context Bank 40	
0x569_0000 - 0x569_FFFF	Translation Context Bank 41	
0x56A_0000 - 0x56A_FFFF	Translation Context Bank 42	
0x56B_0000 - 0x56B_FFFF	Translation Context Bank 43	
0x56C_0000 - 0x56C_FFFF	Translation Context Bank 44	
0x56D_0000 - 0x56D_FFFF	Translation Context Bank 45	
0x56E_0000 - 0x56E_FFFF	Translation Context Bank 46	

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x56F_0000 - 0x56F_FFFF	Translation Context Bank 47	
0x570_0000 - 0x570_FFFF	Translation Context Bank 48	
0x571_0000 - 0x571_FFFF	Translation Context Bank 49	
0x572_0000 - 0x572_FFFF	Translation Context Bank 50	
0x573_0000 - 0x573_FFFF	Translation Context Bank 51	
0x574_0000 - 0x574_FFFF	Translation Context Bank 52	
0x575_0000 - 0x575_FFFF	Translation Context Bank 53	
0x576_0000 - 0x576_FFFF	Translation Context Bank 54	
0x577_0000 - 0x577_FFFF	Translation Context Bank 55	
0x578_0000 - 0x578_FFFF	Translation Context Bank 56	
0x579_0000 - 0x579_FFFF	Translation Context Bank 57	
0x57A_0000 - 0x57A_FFFF	Translation Context Bank 58	
0x57B_0000 - 0x57B_FFFF	Translation Context Bank 59	
0x57C_0000 - 0x57C_FFFF	Translation Context Bank 60	
0x57D_0000 - 0x57D_FFFF	Translation Context Bank 61	
0x57E_0000 - 0x57E_FFFF	Translation Context Bank 62	
0x57F_0000 - 0x57F_FFFF	Translation Context Bank 63	
0x580_0000 - 0x5FF_FFFF	Reserved	
0x600_0000 - 0x6FF_FFFF	GICD registers (GIC500)	
0x601_0000 - 0x601_FFFF	reserved	
0x602_0000 - 0x602_FFFF	GITS registers (GIC500)	ITS control registers
0x603_0000 - 0x603_FFFF	GITS_TRANSLATER (GIC500)	ITS translation register
0x604_0000 - 0x60F_FFFF	Reserved	
0x610_0000 - 0x610_FFFF	CPU 0 control, physical LPIs (RD_base)	A53_1
0x611_0000 - 0x611_FFFF	CPU 0 SGIs, PPIs (SGI_base)	
0x612_0000 - 0x612_FFFF	CPU 1 control, physical LPIs (RD_base)	
0x613_0000 - 0x613_FFFF	CPU 1 SGIs, PPIs (SGI_base)	
0x614_0000 - 0x614_FFFF	CPU 2 control, physical LPIs (RD_base)	
0x615_0000 - 0x615_FFFF	CPU 2 SGIs, PPIs (SGI_base)	
0x616_0000 - 0x616_FFFF	CPU 3 control, physical LPIs (RD_base)	
0x617_0000 - 0x617_FFFF	CPU 3 SGIs, PPIs (SGI_base)	
0x618_0000 - 0x618_FFFF	CPU 4 control, physical LPIs (RD_base)	A53_2
0x619_0000 - 0x619_FFFF	CPU 4 SGIs, PPIs (SGI_base)	
0x61A_0000 - 0x61A_FFFF	CPU 5 control, physical LPIs (RD_base)	
0x61B_0000 - 0x61B_FFFF	CPU 5 SGIs, PPIs (SGI_base)	
0x61C_0000 - 0x61C_FFFF	CPU 6 control, physical LPIs (RD_base)	
0x61D_0000 - 0x61D_FFFF	CPU 6 SGIs, PPIs (SGI_base)	

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
0x61E_0000 - 0x61E_FFFF	CPU 7 control, physical LPIs (RD_base)	
0x61F_0000 - 0x61F_FFFF	CPU 7 SGIs, PPIs (SGI_base)	
0x620_0000 - 0x6FF_FFFF	Reserved	
0x700_0000 - 0x7FF_FFFF	Reserved	
0x800_0000 - 0x80F_FFFF	SEC	
0x810_0000 - 0x811_FFFF	Reserved	
0x812_0000 - 0x812_FFFF	Reserved	
0x813_0000 - 0x815_FFFF	Reserved	
0x816_0000 - 0x816_FFFF	Reserved	
0x817_0000 - 0x817_FFFF	Reserved	
0x818_0000 - 0x819_FFFF	QBMAN	
0x81A_0000 - 0x81F_FFFF	Reserved	
0x820_0000 - 0x820_FFFF	PEBM	
0x821_0000 - 0x821_FFFF	Reserved	
0x822_0000 - 0x823_FFFF	Reserved	
0x824_0000 - 0x824_FFFF	LCFG	DPAA2 config space
0x825_0000 - 0x828_FFFF	Reserved	
0x829_0000 - 0x829_FFFF	DP-PM (Power Management) Unit	DPAA2 power management
0x82A_0000 - 0x833_FFFF	Reserved	
0x834_0000 - 0x837_FFFF	Management Complex (MC)	
0x838_0000 - 0x838_FFFF	qDMA general configuration	
0x839_0000 - 0x839_FFFF	qDMA management	
0x83A_0000 - 0x87F_FFFF	Reserved	
0x880_0000 - 0x883_FFFF	WRIOP NI	Config space; portals are in memory space
0x884_0000 - 0x88F_FFFF	Reserved	
0x890_0000 - 0x895_FFFF	AIOP	
0x896_0000 - 0x8B7_FFFF	Reserved	
0x8B8_0000 - 0x8B9_FFFF	WRIOP (glbl - global)	Config space; ports are in a separate offset
0x8BA_0000 - 0x8BF_FFFF	Reserved	
0x8C0_0000 - 0x8CF_FFFF	WRIOP (port)	
0x8D0_0000 - 0xBFF_FFFF	Reserved	
0xC00_0000 - 0xC00_FFFF	Cluster 1 core 0 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC01_0000 - 0xC01_FFFF	Cluster 1 core 1 peripherals	offset 0x0000 - 0x0FFF - watchdog timer

Table continues on the next page...

Table 2-1. CCSR Block Base Address Map (continued)

Block Base Address (Hex) (CCSR offset 100_0000 + Address Offset of the block in 240 MB CCSR space)	Block	Comments
		offset 0x1000 - 0xFFFF - reserved
0xC02_0000 - 0xC02_FFFF	Cluster 1 core 2 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC03_0000 - 0xC03_FFFF	Cluster 1 core 3 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC04_0000 - 0xC0F_FFFF		
0xC10_0000 - 0xC10_FFFF	Cluster 2 core 0 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC11_0000 - 0xC11_FFFF	Cluster 2 core 1 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC12_0000 - 0xC12_FFFF	Cluster 2 core 2 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC13_0000 - 0xC13_FFFF	Cluster 2 core 3 peripherals	offset 0x0000 - 0x0FFF - watchdog timer offset 0x1000 - 0xFFFF - reserved
0xC14_0000 - 0xFFF_FFFF	Reserved	

2.4 DCSR address map

The following table lists the absolute block base address assigned to each block.

Table 2-2. DCSR block base address map

Block Base Address (Hex)	Block	Comments
0x6_0000 - 0x6_0FFF	EPU	-
0x460_0000 - 0x460_FFFF	USBPHY1	-
0x461_0000 - 0x461_FFFF	USBPHY2	-

2.5 Source ID Assignments

This table shows the Source ID Assignments for the chip.

SourceID value	Transaction source	Notes
0x00	Reserved - DO NOT USE	Reserved for invalid source ID value for error logging
0x02	Reserved	
0x03	Reserved	
0x04	Reserved	
0x05	Reserved	
0x06	Reserved	
0x07	Reserved	
0x08	Reserved	
0x09	Reserved	
0x0A	Reserved	
0x0B	Reserved	
0x0C	Reserved	
0x0D	Reserved	
0x0E	Reserved	
0x0F	Reserved	
0x10	Reserved	
0x11	Reserved	
0x12	Reserved	
0x13	Reserved	
0x14	Reserved	
0x15	Reserved	
0x16	Reserved	
0x17	Reserved	
0x18	PCI-Express 1	
0x19	PCI-Express 2	
0x1A	PCI-Express 3	
0x1B	Reserved	
0x1C	Reserved	
0x1D	Reserved	
0x1E	Reserved	
0x1F	Reserved	
0x20	Reserved	
0x21	Reserved	
0x22	Reserved	
0x23	Reserved	
0x24	Reserved	
0x25	Reserved	
0x26	Reserved	
0x27	Reserved	
0x28	Reserved	

Table continues on the next page...

0x29	Reserved	
0x2A	Reserved	
0x2B	Reserved	
0x2C	Reserved	
0x2D	Reserved	
0x2E	Reserved	
0x2F	Reserved	
0x30	GIC-500 GITS target ID value	
0x31		
0x32		
0x33		
0x34		
0x35		
0x36		
0x37		
0x38		
0x39		
0x3A		
0x3B		
0x3C		
0x3D		
0x3E		
0x3F		
0x40	Service Processor	
0x41	SAP	
0x42	Reserved	
0x43	Reserved	
0x44	eSDHC/(SD/MMC)	
0x45	qDMA	
0x46	Reserved	
0x47	Reserved	
0x48	SMMU	Tablewalks/ATS
0x49	GIC	ITS
0x4A	Reserved	
0x4B	Reserved	
0x4C	Reserved	
0x4D	Reserved	
0x4E	DAP	
0x4F	DTC	
0x50	MC (Management Complex)	
0x51	Reserved	
0x52	Reserved	

Table continues on the next page...

0x53	Reserved	
0x54	Reserved	
0x55	Reserved	
0x56	Reserved	
0x57	Reserved	
0x58	Reserved	
0x59	Reserved	
0x5A	Reserved	
0x5B	Reserved	
0x5C	Reserved	
0x5D	Reserved	
0x5E	Reserved	
0x5F	Reserved	
0x60	QBMAN	
0x61	Reserved	
0x62	Reserved	
0x63	Reserved	
0x64	Reserved	
0x65	Reserved	
0x66	Reserved	
0x67	Reserved	
0x68	Reserved	
0x69	Reserved	
0x6A	Reserved	
0x6B	Reserved	
0x6C	Reserved	
0x6D	Reserved	
0x6E	Reserved	
0x6F	Reserved	
0x70	Reserved	
0x71	SEC	
0x72	Reserved	
0x73	Reserved for DP Hardware Accelerator	
0x74	Reserved for DP Hardware Accelerator	
0x75	Reserved for DP Hardware Accelerator	
0x76	Reserved for DP Hardware Accelerator	
0x77	Reserved for DP Hardware Accelerator	
0x78	Reserved for DP Hardware Accelerator	
0x79	Reserved for DP Hardware Accelerator	
0x7A	Reserved for DP Hardware Accelerator	
0x7B	Reserved for DP Hardware Accelerator	
0x7C	Reserved for DP Hardware Accelerator	

Table continues on the next page...

0x7D	Reserved for DP Hardware Accelerator	
0x7E	Reserved for DP Hardware Accelerator	
0x7F	Reserved for DP Hardware Accelerator	
0x80	AIOP 1 FDMA	packet data
0x81	AIOP 1 CDMA	context data
0x82	AIOP 1 CTLU/MFLU	table lookups
0x83	AIOP1 other	STE, TMan, ATU, etc.
0x84	Reserved	
0x85	Reserved	
0x86	Reserved	
0x87	Reserved	
0x88	Reserved	
0x89	Reserved	
0x8A	Reserved	
0x8B	Reserved	
0x8C	Reserved	
0x8D	Reserved	
0x8E	Reserved	
0x8F	Reserved	
0x90	WRIOP 1	
0x91	Reserved	
0x92	Reserved	
0x93	Reserved	
0x94	Reserved	
0x95	Reserved	
0x96	Reserved	
0x97	Reserved	
0x98	Reserved	
0x99	Reserved	
0x9A	Reserved	
0x9B	Reserved	
0x9C	Reserved	
0x9D	Reserved	
0x9E	Reserved	
0x9F	Reserved	
0xA0	USB 1	
0xA1	USB 2	
0xA2	Reserved	
0xA3	Reserved	
0xA4	Reserved	
0xA5	Reserved	
0xA6	Reserved	

Table continues on the next page...

0xA7	Reserved	
0xA8	HDLC(µQE)	
0xA9	Reserved	
0xAA	Reserved	
0xAB	Reserved	
0xAC	Reserved	
0xAD	Reserved	
0xAE	Reserved	
0xAF	Reserved	
0xB0	SATA 1	
0xB1	Reserved	
0xB2	Reserved	
0xB3	Reserved	
0xB4	Reserved	
0xB5	Reserved	
0xB6	Reserved	
0xB7	Reserved	
0xB8	Reserved	
0xB9	Reserved	
0xBA	Reserved	
0xBB	Reserved	
0xBC	Reserved	
0xBD	Reserved	
0xBE	Reserved	
0xBF	Reserved	
* Cx is cluster ([1-8]); Py is Arm core ([0-7]); assumes all cores are single-threaded Cluster number is ((SourceIDvalue[3] || SourceIDvalue[5:4]) + 1); core number is (SourceIDvalue[2:0]) This is organized to provide better granularity for clusters across CCI-400.		
0xC0	General purpose processor ID 0	Cluster 1
0xC1	General purpose processor ID 1	
0xC2	General purpose processor ID 2	
0xC3	General purpose processor ID 3	
0xC4	Reserved	
0xC5	Reserved	
0xC6	Reserved	
0xC7	Reserved	
0xC8	Reserved	Cluster 5
0xC9	Reserved	
0xCA	Reserved	
0xCB	Reserved	
0xCC	Reserved	
0xCD	Reserved	

Table continues on the next page...

0xCE	Reserved	
0xCF	Reserved	
0xD0	General purpose processor ID 8	Cluster 2
0xD1	General purpose processor ID 9	
0xD2	General purpose processor ID 10	
0xD3	General purpose processor ID 11	
0xD4	Reserved	
0xD5	Reserved	
0xD6	Reserved	
0xD7	Reserved	
0xD8	Reserved	Cluster 6
0xD9	Reserved	
0xDA	Reserved	
0xDB	Reserved	
0xDC	Reserved	
0xDD	Reserved	
0xDE	Reserved	
0xDF	Reserved	
0xE0	Reserved	Cluster 3
0xE1	Reserved	
0xE2	Reserved	
0xE3	Reserved	
0xE4	Reserved	
0xE5	Reserved	
0xE6	Reserved	
0xE7	Reserved	Cluster 7
0xE8	Reserved	
0xE9	Reserved	
0xEA	Reserved	
0xEB	Reserved	
0xEC	Reserved	
0xED	Reserved	
0xEE	Reserved	
0xEF	Reserved	
0xF0	Reserved	Cluster 4
0xF1	Reserved	
0xF2	Reserved	
0xF3	Reserved	
0xF4	Reserved	
0xF5	Reserved	
0xF6	Reserved	
0xF7	Reserved	

Table continues on the next page...

0xF8	Reserved	Cluster 8
0xF9	Reserved	
0xFA	Reserved	
0xFB	Reserved	
0xFC	Reserved	
0xFD	Reserved	
0xFE	Reserved	
0xFF	Reserved	

2.6 OCRAM Initialization

For those systems which provide interrupt handling of multi-bit ECC errors (MBEE), prior to use, the OCRAM must first be initialized with 64-bit writes and then a write performed to address 0x0010_0534 with the value 0x0000_0008. This may be accomplished in one of two ways:

- Using NXP-supplied secure firmware and utilizing the OCRAM init built-in option to perform the above initialization steps, OR
- Performing the above OCRAM initialization steps as part of customer boot loader code.

For more information on secure firmware, see the PPA User's Guide.

2.7 DRAM Region Remapping

The Arm memory map places DRAM regions such that there is an upper DRAM region for a 36-, 40-, 44-, or 48-bit physical address space. This region is therefore used for other targets. This leads to the SoC address to DRAM address remapping shown below.

Table 2-3. SoC Memory Address Remapping

SoC Address	Region	Size	Address presented to the DDR Controller	Notes
0x00_8000_0000 - 0x00_FFFF_FFFF	5, 6, 7, 8	2GB	0x00_0000_0000 - 0x00_7FFF_FFFF	DRAM for 32-bit and greater physical address cores; overlaid SoC address space 0x80_0000_0000 - 0x80_7FFF_FFFF is reserved
0x08_0000_0000 - 0x08_1FFF_FFFF	12	512MB	-	DPAA2 portal space, so no SoC re-mapping and

Table continues on the next page...

Table 2-3. SoC Memory Address Remapping (continued)

SoC Address	Region	Size	Address presented to the DDR Controller	Notes
				no mapping to DRAM space
0x08_2000_0000 - 0x0F_FFFF_FFFF	12	31.5GB	-	Reserved, no mapping to DRAM space
0x80_0000_0000 - 0x80_7FFF_FFFF	16	2GB	-	Invalid DRAM space - this is the “hole” for the 2GB DRAM space in the 32-bit address map
0x80_8000_0000 - 0xFF_FFFF_FFFF	16	510GB (62GB to DRAM controller)	0x00_8000_0000 - 0x7F_FFFF_FFFF	DRAM for 40-bit and greater physical address cores

2.8 IFC, QSPI Address Remapping

Both IFC and QSPI support large memorie (up to 4GB each) and those memories may be accessed during the SoC reset sequence for loading RCW (IFC only) or the boot code. Since this occurs before the system MMUs are configured, some IFC and QSPI memory must be available in the lower 4GB space of the SoC address map. Therefore, both IFC and QSPI memory regions were split into two parts: a 256MB region that is located in the least significant 4GB of the SoC address space and a 3.75GB region that is located above the least significant 4GB of the SoC address space.

The 4GB QSPI address space map is shown below.

Table 2-4. QSPI Address Remapping

SoC Address	QSPI Address	Notes
0x00_2000_0000 - 0x00_2FFF_FFFF	0x00_0000_0000 - 0x00_0FFF_FFFF	First 256MB
0x04_1000_0000 - 0x04_FFFF_FFFF	0x00_1000_0000 - 0x00_FFFF_FFFF	Last 3.75GB

The 4GB IFC address space map is shown below.

Table 2-5. IFC Address Remapping

SoC Address	QSPI Address	Notes
0x00_3000_0000 - 0x00_3FFF_FFFF	0x00_0000_0000 - 0x00_0FFF_FFFF	First 256MB
0x05_1000_0000 - 0x05_FFFF_FFFF	0x00_1000_0000 - 0x00_FFFF_FFFF	Last 3.75GB

Chapter 3
Signal Descriptions

3.1 Signals Introduction

This chapter describes the external signals and is organized into the following sections:

- Overview of signals and cross-references for signals that serve multiple functions
- List of output signal states at reset

3.2 Signals Overview

Note that individual chapters of this document provide details for each signal, describing each signal's behavior when the signal is asserted or negated and when the signal is an input or an output.

The following tables provides a summary of the signals grouped by function. This table details the signal name, interface, alternate functions, number of signals, and whether the signal is an input, output, or bidirectional. The direction of the multiplexed signals applies for the primary signal function listed in the left-most column of the table for that row (and does not apply for the state of the reset configuration signals). Finally, the tables provide a pointer to the table where the signal function is described.

Table 3-1. LS1088A Signal Reference by Functional Block

Name	Description	Alternate Function(s)	Pin type
DDR SDRAM Memory Interface 1			
D1_MA00	Address	-	O
D1_MA01	Address	-	O
D1_MA02	Address	-	O
D1_MA03	Address	-	O
D1_MA04	Address	-	O
D1_MA05	Address	-	O

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
D1_MA06	Address	-	O
D1_MA07	Address	-	O
D1_MA08	Address	-	O
D1_MA09	Address	-	O
D1_MA10	Address	-	O
D1_MA11	Address	-	O
D1_MA12	Address	-	O
D1_MA13	Address	-	O
D1_MACT_B	Activate	-	O
D1_MALERT_B	Alert		I
D1_MBA0	Bank Select	-	O
D1_MBA1	Bank Select	-	O
D1_MBG0	Bank Group	-	O
D1_MBG1	Bank Group	-	O
D1_MCAS_B	Column Address Strobe / MA[15]	-	O
D1_MCK0	Clock	-	O
D1_MCK0_B	Clock Complement	-	O
D1_MCK1	Clock	-	O
D1_MCK1_B	Clock Complement	-	O
D1_MCKE0	Clock Enable	-	O
D1_MCKE1	Clock Enable	-	O
D1_MCS0_B	Chip Select	-	O
D1_MCS1_B	Chip Select	-	O
D1_MCS2_B	Chip Select	-	O
D1_MCS3_B	Chip Select	-	O
D1_MDIC0	Driver Impedence Calibration	-	IO
D1_MDIC1	Driver Impedence Calibration	-	IO
D1_MDM0	Data Mask	-	O
D1_MDM1	Data Mask	-	O
D1_MDM2	Data Mask	-	O
D1_MDM3	Data Mask	-	O
D1_MDM4	Data Mask	-	O
D1_MDM5	Data Mask	-	O
D1_MDM6	Data Mask	-	O
D1_MDM7	Data Mask	-	O
D1_MDM8	Data Mask	-	O
D1_MDQ00	Data	-	IO
D1_MDQ01	Data	-	IO
D1_MDQ02	Data	-	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
D1_MDQ03	Data	-	IO
D1_MDQ04	Data	-	IO
D1_MDQ05	Data	-	IO
D1_MDQ06	Data	-	IO
D1_MDQ07	Data	-	IO
D1_MDQ08	Data	-	IO
D1_MDQ09	Data	-	IO
D1_MDQ10	Data	-	IO
D1_MDQ11	Data	-	IO
D1_MDQ12	Data	-	IO
D1_MDQ13	Data	-	IO
D1_MDQ14	Data	-	IO
D1_MDQ15	Data	-	IO
D1_MDQ16	Data	-	IO
D1_MDQ17	Data	-	IO
D1_MDQ18	Data	-	IO
D1_MDQ19	Data	-	IO
D1_MDQ20	Data	-	IO
D1_MDQ21	Data	-	IO
D1_MDQ22	Data	-	IO
D1_MDQ23	Data	-	IO
D1_MDQ24	Data	-	IO
D1_MDQ25	Data	-	IO
D1_MDQ26	Data	-	IO
D1_MDQ27	Data	-	IO
D1_MDQ28	Data	-	IO
D1_MDQ29	Data	-	IO
D1_MDQ30	Data	-	IO
D1_MDQ31	Data	-	IO
D1_MDQ32	Data	-	IO
D1_MDQ33	Data	-	IO
D1_MDQ34	Data	-	IO
D1_MDQ35	Data	-	IO
D1_MDQ36	Data	-	IO
D1_MDQ37	Data	-	IO
D1_MDQ38	Data	-	IO
D1_MDQ39	Data	-	IO
D1_MDQ40	Data	-	IO
D1_MDQ41	Data	-	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
D1_MDQ42	Data	-	IO
D1_MDQ43	Data	-	IO
D1_MDQ44	Data	-	IO
D1_MDQ45	Data	-	IO
D1_MDQ46	Data	-	IO
D1_MDQ47	Data	-	IO
D1_MDQ48	Data	-	IO
D1_MDQ49	Data	-	IO
D1_MDQ50	Data	-	IO
D1_MDQ51	Data	-	IO
D1_MDQ52	Data	-	IO
D1_MDQ53	Data	-	IO
D1_MDQ54	Data	-	IO
D1_MDQ55	Data	-	IO
D1_MDQ56	Data	-	IO
D1_MDQ57	Data	-	IO
D1_MDQ58	Data	-	IO
D1_MDQ59	Data	-	IO
D1_MDQ60	Data	-	IO
D1_MDQ61	Data	-	IO
D1_MDQ62	Data	-	IO
D1_MDQ63	Data	-	IO
D1_MDQS0	Data Strobe	-	IO
D1_MDQS0_B	Data Strobe	-	IO
D1_MDQS1	Data Strobe	-	IO
D1_MDQS1_B	Data Strobe	-	IO
D1_MDQS2	Data Strobe	-	IO
D1_MDQS2_B	Data Strobe	-	IO
D1_MDQS3	Data Strobe	-	IO
D1_MDQS3_B	Data Strobe	-	IO
D1_MDQS4	Data Strobe	-	IO
D1_MDQS4_B	Data Strobe	-	IO
D1_MDQS5	Data Strobe	-	IO
D1_MDQS5_B	Data Strobe	-	IO
D1_MDQS6	Data Strobe	-	IO
D1_MDQS6_B	Data Strobe	-	IO
D1_MDQS7	Data Strobe	-	IO
D1_MDQS7_B	Data Strobe	-	IO
D1_MDQS8	Data Strobe	-	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
D1_MDQS8_B	Data Strobe	-	IO
D1_MECC0	Error Correcting Code	-	IO
D1_MECC1	Error Correcting Code	-	IO
D1_MECC2	Error Correcting Code	-	IO
D1_MECC3	Error Correcting Code	-	IO
D1_MECC4	Error Correcting Code	-	IO
D1_MECC5	Error Correcting Code	-	IO
D1_MECC6	Error Correcting Code	-	IO
D1_MECC7	Error Correcting Code	-	IO
D1_MODT0	On Die Termination	-	O
D1_MODT1	On Die Termination	-	O
D1_MPAR	Address Parity Out	-	O
D1_MRAS_B	Row Address Strobe / MA[16]	-	O
D1_MWE_B	Write Enable / MA[14]	-	O
Integrated Flash Controller			
IFC_A00	IFC Address	GPIO1_16 QSPI_A_CS0	O
IFC_A01	IFC Address	GPIO1_17 QSPI_A_CS1	O
IFC_A02	IFC Address	GPIO1_18 QSPI_A_SCK	O
IFC_A03	IFC Address	GPIO1_19 QSPI_B_CS0	O
IFC_A04	IFC Address	GPIO1_20 QSPI_B_CS1	O
IFC_A05	IFC Address	GPIO1_21 QSPI_B_SCK cfg_dram_type	O
IFC_A06 / IFC_WP1_B	IFC Address	GPIO2_00 QSPI_A_DATA0	O
IFC_A07 / IFC_WP2_B	IFC Address	GPIO2_01 QSPI_A_DATA1	O
IFC_A08 / IFC_WP3_B	IFC Address	GPIO2_02 QSPI_A_DATA2	O
IFC_A09 / IFC_RB2_B / IFC_CS_B4	IFC Address	GPIO2_03 QSPI_A_DATA3	O
IFC_A10 / IFC_RB3_B / IFC_CS_B5	IFC Address	GPIO2_04 QSPI_A_DQS	O
IFC_A11 / IFC_CS_B6	IFC Address	GPIO2_05 QSPI_B_DQS	O
IFC_AD00	IFC Address / Data	GPIO1_00 cfg_gpinput0	IO
IFC_AD01	IFC Address / Data	GPIO1_01	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
		cfg_gpinput1	
IFC_AD02	IFC Address / Data	GPIO1_02 cfg_gpinput2	IO
IFC_AD03	IFC Address / Data	GPIO1_03 cfg_gpinput3	IO
IFC_AD04	IFC Address / Data	GPIO1_04 cfg_gpinput4	IO
IFC_AD05	IFC Address / Data	GPIO1_05 cfg_gpinput5	IO
IFC_AD06	IFC Address / Data	GPIO1_06 cfg_gpinput6	IO
IFC_AD07	IFC Address / Data	GPIO1_07 cfg_gpinput7	IO
IFC_AD08	IFC Address / Data	GPIO1_08 cfg_rcw_src1	IO
IFC_AD09	IFC Address / Data	GPIO1_09 cfg_rcw_src2	IO
IFC_AD10	IFC Address / Data	GPIO1_10 cfg_rcw_src3	IO
IFC_AD11	IFC Address / Data	GPIO1_11 cfg_rcw_src4	IO
IFC_AD12	IFC Address / Data	GPIO1_12 cfg_rcw_src5	IO
IFC_AD13	IFC Address / Data	GPIO1_13 cfg_rcw_src6	IO
IFC_AD14	IFC Address / Data	GPIO1_14 cfg_rcw_src7	IO
IFC_AD15	IFC Address / Data	GPIO1_15 cfg_rcw_src8	IO
IFC_ALE	IFC Address Latch Enable	GPIO1_24	O
IFC_BCTL	IFC Buffer control	GPIO2_12	O
IFC_CLE	IFC NAND Command Latch Enable / Write Enable 1 / NOR Address active-low Valid	GPIO1_25 cfg_rcw_src0	O
IFC_CLK0	IFC Clock	GPIO2_17	O
IFC_CLK1	IFC Clock	GPIO2_18	O
IFC_CS0_B	IFC Chip Select	GPIO2_08	O
IFC_CS1_B	IFC Chip Select	GPIO2_09	O
IFC_CS2_B	IFC Chip Select	GPIO2_10	O
IFC_CS3_B	IFC Chip Select	GPIO2_11 QSPI_B_DATA3 QSPI_A_DATA7	O
IFC_CS_B4 / **IFC_A09** / IFC_RB2_B	IFC Chip Select	GPIO2_03 QSPI_A_DATA3	O

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
IFC_CS_B5 / **IFC_A10** / IFC_RB3_B	IFC Chip Select	GPIO2_04 QSPI_A_DQS	O
IFC_CS_B6 / **IFC_A11**	IFC Chip Select	GPIO2_05 QSPI_B_DQS	O
IFC_NDDQS	IFC DQS Strobe	GPIO2_13	IO
IFC_NDWE_B	IFC NAND Write Enable / NAND DDR Clock	GPIO2_19	O
IFC_OE_B	IFC Output Enable	GPIO1_26 cfg_eng_use1	O
IFC_PAR0	IFC Address & Data Parity	GPIO2_06 QSPI_B_DATA0 QSPI_A_DATA4	IO
IFC_PAR1	IFC Address & Data Parity	GPIO2_07 QSPI_B_DATA1 QSPI_A_DATA5	IO
IFC_PERR_B	IFC Parity Error	GPIO2_16 QSPI_B_DATA2 QSPI_A_DATA6	I
IFC_RB0_B	IFC Ready/Busy CS0	GPIO2_14	I
IFC_RB1_B	IFC Ready/Busy CS1	GPIO2_15	I
IFC_RB2_B / **IFC_A09** / IFC_CS_B4	IFC Ready/Busy CS 2	GPIO2_03 QSPI_A_DATA3	I
IFC_RB3_B / **IFC_A10** / IFC_CS_B5	IFC Ready/Busy CS 3	GPIO2_04 QSPI_A_DQS	I
IFC_TE	IFC External Transceiver Enable	GPIO1_23 cfg_ifc_te	O
IFC_WE0_B	IFC Write Enable 0 / Start of Frame	GPIO1_22 cfg_eng_use0	O
IFC_WP0_B	IFC Write Protect	GPIO1_27 cfg_eng_use2	O
IFC_WP1_B / **IFC_A06**	IFC Write Protect	GPIO2_00 QSPI_A_DATA0	O
IFC_WP2_B / **IFC_A07**	IFC Write Protect	GPIO2_01 QSPI_A_DATA1	O
IFC_WP3_B / **IFC_A08**	IFC Write Protect	GPIO2_02 QSPI_A_DATA2	O
DUART1			
UART1_CTS_B	Clear To Send	GPIO3_10 UART3_SIN	I
UART1_RTS_B	Ready to Send	GPIO3_08 UART3_SOUT	O
UART1_SIN	Receive Data		I
UART1_SOUT	Transmit Data		O
DUART2			
UART2_CTS_B	Clear To Send	GPIO3_11	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
		UART4_SIN	
UART2_RTS_B	Ready to Send	GPIO3_09 UART4_SOUT	O
UART2_SIN	Receive Data	GPIO3_07	I
UART2_SOUT	Transmit Data	GPIO3_06	O
DUART3 and 4			
UART3_SIN	Receive Data	**UART1_CTS_B** GPIO3_10	I
UART3_SOUT	Transmit Data	**UART1_RTS_B** GPIO3_08	O
UART4_SIN	Receive Data	**UART2_CTS_B** GPIO3_11	I
UART4_SOUT	Transmit Data	**UART2_RTS_B** GPIO3_09	O
I2C1			
IIC1_SCL	Serial Clock		IO
IIC1_SDA	Serial Data		IO
I2C2			
IIC2_SCL	Serial Clock	GPIO3_12 SDHC_CD_B CLK9 BRGO2	IO
IIC2_SDA	Serial Data	GPIO3_13 SDHC_WP CLK10 BRGO3	IO
I2C3 and 4			
IIC3_SCL	Serial Clock	GPIO4_28 EVT5_B USB2_DRVVBUS BRGO4 CLK11	IO
IIC3_SDA	Serial Data	GPIO4_29 EVT6_B USB2_PWRFAULT BRGO1 CLK12_CLK8	IO
IIC4_SCL	Serial Clock	GPIO4_30 EVT7_B TDMA_RQ UC1_CDB_RXER	IO
IIC4_SDA	Serial Data	GPIO4_31 EVT8_B TDMB_RQ UC3_CDB_RXER	IO
SPI Interface			

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
SPI_PCS0	SPI Chip Select	GPIO3_17 SDHC_DAT4 SDHC_VS	IO
SPI_PCS1	SPI Chip Select	GPIO3_18 SDHC_DAT5 SDHC_CMD_DIR	O
SPI_PCS2	SPI Chip Select	GPIO3_19 SDHC_DAT6 SDHC_DAT0_DIR	O
SPI_PCS3	SPI Chip Select	GPIO3_20 SDHC_DAT7 SDHC_DAT123_DIR	O
SPI_SCK	SPI Clock	GPIO3_16 SDHC_GATE_IN	IO
SPI_SIN	Master In Slave Out	GPIO3_15 SDHC_CLK_SYNC_IN	I
SPI_SOUT	Master Out Slave In	GPIO3_14 SDHC_CLK_SYNC_OUT	O
eSDHC			
SDHC_CD_B	Command	**IIC2_SCL** GPIO3_12 CLK9 BRGO2	I
SDHC_CLK	Host to Card Clock	GPIO3_26	O
SDHC_CLK_SYNC_IN	IN	**SPI_SIN** GPIO3_15	I
SDHC_CLK_SYNC_OUT	OUT	**SPI_SOUT** GPIO3_14	O
SDHC_CMD	Command/Response	GPIO3_21	IO
SDHC_CMD_DIR / SDHC_DAT5	DIR	**SPI_PCS1** GPIO3_18	O
SDHC_DAT0	Data	GPIO3_22	IO
SDHC_DAT0_DIR / SDHC_DAT6	DIR	**SPI_PCS2** GPIO3_19	O
SDHC_DAT1	Data	GPIO3_23	IO
SDHC_DAT123_DIR / SDHC_DAT7	DIR	**SPI_PCS3** GPIO3_20	O
SDHC_DAT2	Data	GPIO3_24	IO
SDHC_DAT3	Data	GPIO3_25	IO
SDHC_DAT4 / SDHC_VS	Data	**SPI_PCS0** GPIO3_17	IO
SDHC_DAT5 / SDHC_CMD_DIR	Data	**SPI_PCS1** GPIO3_18	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
SDHC_DAT6 / SDHC_DAT0_DIR	Data	**SPI_PCS2** GPIO3_19	IO
SDHC_DAT7 / SDHC_DAT123_DIR	Data	**SPI_PCS3** GPIO3_20	IO
SDHC_GATE_IN	IN	**SPI_SCK** GPIO3_16	I
SDHC_VS / SDHC_DAT4	VS	**SPI_PCS0** GPIO3_17	O
SDHC_WP	Write Protect	**IIC2_SDA** GPIO3_13 CLK10 BRGO3	I
Interrupt Controller			
EVT9_B	Interrupt Output	GPIO4_10	IO
IRQ00	External Interrupt		I
IRQ01	External Interrupt		I
IRQ02	External Interrupt		I
IRQ03	External Interrupt	GPIO3_27 TDMB_TSYNC UC3_RTSB_TXEN	I
IRQ04	External Interrupt	GPIO3_28 TDMA_RXD UC1_RXD7 TDMA_TXD	I
IRQ05	External Interrupt	GPIO3_29 TDMA_RSYNC UC1_CTSB_RXDV	I
IRQ06	External Interrupt	GPIO4_04 TDMA_RXD_EXC TDMA_TXD UC1_TXD7	I
IRQ07	External Interrupt	GPIO4_05 TDMA_TSYNC UC1_RTSB_TXEN	I
IRQ08	External Interrupt	GPIO4_06 TDMB_RXD UC3_RXD7 TDMB_TXD	I
IRQ09	External Interrupt	GPIO4_07 TDMB_RSYNC UC3_CTSB_RXDV	I
IRQ10	External Interrupt	GPIO4_08 TDMB_RXD_EXC TDMB_TXD UC3_TXD7	I
IRQ11	External Interrupt	GPIO4_09	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
Debug			
ASLEEP	Asleep	GPIO1_28 cfg_soc_use	O
CKSTP_OUT_B	Checkstop Out	-	O
CLK_OUT	Clock Out	-	O
EVT0_B	Event 0	-	IO
EVT1_B	Event 1	-	IO
EVT2_B	Event 2	-	IO
EVT3_B	Event 3	-	IO
EVT4_B	Event 4	-	IO
EVT5_B	Event 5	**IIC3_SCL** GPIO4_28 USB2_DRVVBUS BRGO4 CLK11	IO
EVT6_B	Event 6	**IIC3_SDA** GPIO4_29 USB2_PWRFAULT BRGO1 CLK12_CLK8	IO
EVT7_B	Event 7	**IIC4_SCL** GPIO4_30 TDMA_RQ UC1_CDB_RXER	IO
EVT8_B	Event 8	**IIC4_SDA** GPIO4_31 TDMB_RQ UC3_CDB_RXER	IO
Trust			
TA_BB_TMP_DETECT_B	Battery Backed Tamper Detect	-	I
TA_TMP_DETECT_B	Tamper Detect		I
System Control			
HRESET_B	Hard Reset	-	IO
PORESET_B	Power On Reset	-	I
RESET_REQ_B	Reset Request (POR or Hard)		O
Clocking			
DIFF_SYSCLK	Single Source System Clock Differential (positive)	-	I
DIFF_SYSCLK_B	Single Source System Clock Differential (negative)	-	I
RTC	Real Time Clock	GPIO3_30	I
SYSCLK	System Clock	-	I
DDR Clocking			
DDRCLK	DDR Controller Clock	-	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
DFT			
JTAG_BSR_VSEL	Reserved	-	I
SCAN_MODE_B	Reserved	-	I
TBSCAN_EN_B	Test Boundary Scan Enable	-	I
TEST_SEL_B	Reserved	-	I
JTAG			
TCK	Test Clock	-	I
TDI	Test Data In	-	I
TDO	Test Data Out	-	O
TMS	Test Mode Select	-	I
TRST_B	Test Reset	-	I
Analog Signals			
D1_TPA	Reserved	-	IO
FA_ANALOG_G_V	Reserved	-	IO
FA_ANALOG_PIN	Reserved	-	IO
TD1_ANODE	Thermal diode anode	-	IO
TD1_CATHODE	Thermal diode cathode	-	IO
TH_TPA	Reserved	-	-
SerDes 1			
SD1_IMP_CAL_RX	SerDes Receive Impedence Calibration	-	I
SD1_IMP_CAL_TX	SerDes Transmit Impedance Calibration	-	I
SD1_PLL1_TPA	SerDes PLL 1 Test Point Analog	-	O
SD1_PLL1_TPD	SerDes Test Point Digital	-	O
SD1_PLL2_TPA	SerDes PLL 2 Test Point Analog	-	O
SD1_PLL2_TPD	SerDes Test Point Digital	-	O
SD1_REF_CLK1_N	SerDes PLL 1 Reference Clock Complement	-	I
SD1_REF_CLK1_P	SerDes PLL 1 Reference Clock	-	I
SD1_REF_CLK2_N	SerDes PLL 2 Reference Clock Complement	-	I
SD1_REF_CLK2_P	SerDes PLL 2 Reference Clock	-	I
SD1_RX0_N	SerDes Receive Data (negative)	-	I
SD1_RX0_P	SerDes Receive Data (positive)	-	I
SD1_RX1_N	SerDes Receive Data (negative)	-	I
SD1_RX1_P	SerDes Receive Data (positive)	-	I
SD1_RX2_N	SerDes Receive Data (negative)	-	I
SD1_RX2_P	SerDes Receive Data (positive)	-	I
SD1_RX3_N	SerDes Receive Data (negative)	-	I
SD1_RX3_P	SerDes Receive Data (positive)	-	I
SD1_TX0_N	SerDes Transmit Data (negative)	-	O
SD1_TX0_P	SerDes Transmit Data (positive)	-	O

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
SD1_TX1_N	SerDes Transmit Data (negative)	-	O
SD1_TX1_P	SerDes Transmit Data (positive)	-	O
SD1_TX2_N	SerDes Transmit Data (negative)	-	O
SD1_TX2_P	SerDes Transmit Data (positive)	-	O
SD1_TX3_N	SerDes Transmit Data (negative)	-	O
SD1_TX3_P	SerDes Transmit Data (positive)	-	O
SerDes 2			
SD2_IMP_CAL_RX	SerDes Receive Impedence Calibration	-	I
SD2_IMP_CAL_TX	SerDes Transmit Impedance Calibration	-	I
SD2_PLL1_TPA	SerDes PLL 1 Test Point Analog	-	O
SD2_PLL1_TPD	SerDes Test Point Digital	-	O
SD2_PLL2_TPA	SerDes PLL 2 Test Point Analog	-	O
SD2_PLL2_TPD	SerDes Test Point Digital	-	O
SD2_REF_CLK1_N	SerDes PLL 1 Reference Clock Complement	-	I
SD2_REF_CLK1_P	SerDes PLL 1 Reference Clock	-	I
SD2_REF_CLK2_N	SerDes PLL 2 Reference Clock Complement	-	I
SD2_REF_CLK2_P	SerDes PLL 2 Reference Clock	-	I
SD2_RX0_N	SerDes Receive Data (negative)	-	I
SD2_RX0_P	SerDes Receive Data (positive)	-	I
SD2_RX1_N	SerDes Receive Data (negative)	-	I
SD2_RX1_P	SerDes Receive Data (positive)	-	I
SD2_RX2_N	SerDes Receive Data (negative)	-	I
SD2_RX2_P	SerDes Receive Data (positive)	-	I
SD2_RX3_N	SerDes Receive Data (negative)	-	I
SD2_RX3_P	SerDes Receive Data (positive)	-	I
SD2_TX0_N	SerDes Transmit Data (negative)	-	O
SD2_TX0_P	SerDes Transmit Data (positive)	-	O
SD2_TX1_N	SerDes Transmit Data (negative)	-	O
SD2_TX1_P	SerDes Transmit Data (positive)	-	O
SD2_TX2_N	SerDes Transmit Data (negative)	-	O
SD2_TX2_P	SerDes Transmit Data (positive)	-	O
SD2_TX3_N	SerDes Transmit Data (negative)	-	O
SD2_TX3_P	SerDes Transmit Data (positive)	-	O
USB PHY 1			
USB1_D_M	USB PHY HS Data (-)	-	IO
USB1_D_P	USB PHY HS Data (+)	-	IO
USB1_ID	USB PHY ID Detect	-	I
USB1_RESREF	USB PHY Impedance Calibration	-	IO
USB1_RX_M	USB PHY SS Receive Data (-)	-	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
USB1_RX_P	USB PHY SS Receive Data (+)	-	I
USB1_TX_M	USB PHY SS Transmit Data (-)	-	O
USB1_TX_P	USB PHY SS Transmit Data (+)	-	O
USB1_VBUS	USB PHY VBUS	-	I
USB PHY 2			
USB2_D_M	USB PHY HS Data (-)	-	IO
USB2_D_P	USB PHY HS Data (+)	-	IO
USB2_ID	USB PHY ID Detect	-	I
USB2_RESREF	USB PHY Impedance Calibration	-	IO
USB2_RX_M	USB PHY SS Receive Data (-)	-	I
USB2_RX_P	USB PHY SS Receive Data (+)	-	I
USB2_TX_M	USB PHY SS Transmit Data (-)	-	O
USB2_TX_P	USB PHY SS Transmit Data (+)	-	O
USB2_VBUS	USB PHY VBUS	-	I
USB1 and 2			
USB2_DRVVBUS	DRV VBus	**IIC3_SCL** GPIO4_28 EVT5_B BRGO4 CLK11	O
USB2_PWRFAULT	PWR Fault	**IIC3_SDA** GPIO4_29 EVT6_B BRGO1 CLK12_CLK8	I
USB_DRVVBUS	USB_DRVVBUS	GPIO4_02	O
USB_PWRFAULT	USB_PWRFAULT	GPIO4_03	I
Ethernet Management Interface 1			
EMI1_MDC	Management Data Clock	GPIO4_00	O
EMI1_MDIO	Management Data In/Out	GPIO4_01	IO
Ethernet Management Interface 2			
EMI2_MDC	Management Data Clock	GPIO2_20	O
EMI2_MDIO	Management Data In/Out	GPIO2_21	IO
Ethernet Controller 1			
EC1_GTX_CLK	Transmit Clock Out	GPIO2_27	O
EC1_GTX_CLK125	Reference Clock	GPIO2_28	I
EC1_RXD0	Receive Data	GPIO4_12	I
EC1_RXD1	Receive Data	GPIO4_11	I
EC1_RXD2	Receive Data	GPIO2_30	I
EC1_RXD3	Receive Data	GPIO2_29	I
EC1_RX_CLK	Receive Clock	GPIO4_13	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
EC1_RX_DV	Receive Data Valid	GPIO4_14	I
EC1_TXD0	Transmit Data	GPIO2_25	O
EC1_TXD1	Transmit Data	GPIO2_24	O
EC1_TXD2	Transmit Data	GPIO2_23	O
EC1_TXD3	Transmit Data	GPIO2_22	O
EC1_TX_EN	Transmit Enable	GPIO2_26	O
Ethernet Controller 2			
EC2_GTX_CLK	Transmit Clock Out	GPIO4_20	O
EC2_GTX_CLK125	Reference Clock	GPIO4_21	I
EC2_RXD0	Receive Data	GPIO4_25	I
EC2_RXD1	Receive Data	GPIO4_24	I
EC2_RXD2	Receive Data	GPIO4_23	I
EC2_RXD3	Receive Data	GPIO4_22	I
EC2_RX_CLK	Receive Clock	GPIO4_26	I
EC2_RX_DV	Receive Data Valid	GPIO4_27	I
EC2_TXD0	Transmit Data	GPIO4_18	O
EC2_TXD1	Transmit Data	GPIO4_17	O
EC2_TXD2	Transmit Data	GPIO4_16	O
EC2_TXD3	Transmit Data	GPIO4_15	O
EC2_TX_EN	Transmit Enable	GPIO4_19	O
General Purpose Input/Output			
GPIO1_00	General Purpose Input/Output	**IFC_AD00** cfg_gpinput0	O
GPIO1_01	General Purpose Input/Output	**IFC_AD01** cfg_gpinput1	O
GPIO1_02	General Purpose Input/Output	**IFC_AD02** cfg_gpinput2	O
GPIO1_03	General Purpose Input/Output	**IFC_AD03** cfg_gpinput3	O
GPIO1_04	General Purpose Input/Output	**IFC_AD04** cfg_gpinput4	O
GPIO1_05	General Purpose Input/Output	**IFC_AD05** cfg_gpinput5	O
GPIO1_06	General Purpose Input/Output	**IFC_AD06** cfg_gpinput6	O
GPIO1_07	General Purpose Input/Output	**IFC_AD07** cfg_gpinput7	O
GPIO1_08	General Purpose Input/Output	**IFC_AD08** cfg_rcw_src1	O
GPIO1_09	General Purpose Input/Output	**IFC_AD09** cfg_rcw_src2	O
GPIO1_10	General Purpose Input/Output	**IFC_AD10**	O

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
		cfg_rcw_src3	
GPIO1_11	General Purpose Input/Output	**IFC_AD11** cfg_rcw_src4	O
GPIO1_12	General Purpose Input/Output	**IFC_AD12** cfg_rcw_src5	O
GPIO1_13	General Purpose Input/Output	**IFC_AD13** cfg_rcw_src6	O
GPIO1_14	General Purpose Input/Output	**IFC_AD14** cfg_rcw_src7	O
GPIO1_15	General Purpose Input/Output	**IFC_AD15** cfg_rcw_src8	O
GPIO1_16	General Purpose Input/Output	**IFC_A00** QSPI_A_CS0	O
GPIO1_17	General Purpose Input/Output	**IFC_A01** QSPI_A_CS1	O
GPIO1_18	General Purpose Input/Output	**IFC_A02** QSPI_A_SCK	O
GPIO1_19	General Purpose Input/Output	**IFC_A03** QSPI_B_CS0	O
GPIO1_20	General Purpose Input/Output	**IFC_A04** QSPI_B_CS1	O
GPIO1_21	General Purpose Input/Output	**IFC_A05** QSPI_B_SCK cfg_dram_type	O
GPIO1_22	General Purpose Input/Output	**IFC_WE0_B** cfg_eng_use0	O
GPIO1_23	General Purpose Input/Output	**IFC_TE** cfg_ifc_te	O
GPIO1_24	General Purpose Input/Output	**IFC_ALE**	O
GPIO1_25	General Purpose Input/Output	**IFC_CLE** cfg_rcw_src0	O
GPIO1_26	General Purpose Input/Output	**IFC_OE_B** cfg_eng_use1	O
GPIO1_27	General Purpose Input/Output	**IFC_WP0_B** cfg_eng_use2	O
GPIO1_28	General Purpose Input/Output	**ASLEEP** cfg_soc_use	O
GPIO2_00	General Purpose Input/Output	**IFC_A06** IFC_WP1_B QSPI_A_DATA0	IO
GPIO2_01	General Purpose Input/Output	**IFC_A07** IFC_WP2_B QSPI_A_DATA1	IO
GPIO2_02	General Purpose Input/Output	**IFC_A08** IFC_WP3_B QSPI_A_DATA2	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
GPIO2_03	General Purpose Input/Output	**IFC_A09** IFC_RB2_B IFC_CS_B4 QSPI_A_DATA3	IO
GPIO2_04	General Purpose Input/Output	**IFC_A10** IFC_RB3_B IFC_CS_B5 QSPI_A_DQS	IO
GPIO2_05	General Purpose Input/Output	**IFC_A11** IFC_CS_B6 QSPI_B_DQS	IO
GPIO2_06	General Purpose Input/Output	**IFC_PAR0** QSPI_B_DATA0 QSPI_A_DATA4	IO
GPIO2_07	General Purpose Input/Output	**IFC_PAR1** QSPI_B_DATA1 QSPI_A_DATA5	IO
GPIO2_08	General Purpose Input/Output	**IFC_CS0_B**	IO
GPIO2_09	General Purpose Input/Output	**IFC_CS1_B**	IO
GPIO2_10	General Purpose Input/Output	**IFC_CS2_B**	IO
GPIO2_11	General Purpose Input/Output	**IFC_CS3_B** QSPI_B_DATA3 QSPI_A_DATA7	IO
GPIO2_12	General Purpose Input/Output	**IFC_BCTL**	IO
GPIO2_13	General Purpose Input/Output	**IFC_NDDQS**	IO
GPIO2_14	General Purpose Input/Output	**IFC_RB0_B**	IO
GPIO2_15	General Purpose Input/Output	**IFC_RB1_B**	IO
GPIO2_16	General Purpose Input/Output	**IFC_PERR_B** QSPI_B_DATA2 QSPI_A_DATA6	IO
GPIO2_17	General Purpose Input/Output	**IFC_CLK0**	IO
GPIO2_18	General Purpose Input/Output	**IFC_CLK1**	IO
GPIO2_19	General Purpose Input/Output	**IFC_NDWE_B**	IO
GPIO2_20	General Purpose Input/Output	**EMI2_MDC**	IO
GPIO2_21	General Purpose Input/Output	**EMI2_MDIO**	IO
GPIO2_22	General Purpose Input/Output	**EC1_TXD3**	IO
GPIO2_23	General Purpose Input/Output	**EC1_TXD2**	IO
GPIO2_24	General Purpose Input/Output	**EC1_TXD1**	IO
GPIO2_25	General Purpose Input/Output	**EC1_TXD0**	IO
GPIO2_26	General Purpose Input/Output	**EC1_TX_EN**	IO
GPIO2_27	General Purpose Input/Output	**EC1_GTX_CLK**	IO
GPIO2_28	General Purpose Input/Output	**EC1_GTX_CLK125**	IO
GPIO2_29	General Purpose Input/Output	**EC1_RXD3**	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
GPIO2_30	General Purpose Input/Output	**EC1_RXD2**	IO
GPIO3_06	General Purpose Input/Output	**UART2_SOUT**	IO
GPIO3_07	General Purpose Input/Output	**UART2_SIN**	IO
GPIO3_08	General Purpose Input/Output	**UART1_RTS_B** UART3_SOUT	IO
GPIO3_09	General Purpose Input/Output	**UART2_RTS_B** UART4_SOUT	IO
GPIO3_10	General Purpose Input/Output	**UART1_CTS_B** UART3_SIN	IO
GPIO3_11	General Purpose Input/Output	**UART2_CTS_B** UART4_SIN	IO
GPIO3_12	General Purpose Input/Output	**IIC2_SCL** SDHC_CD_B CLK9 BRGO2	IO
GPIO3_13	General Purpose Input/Output	**IIC2_SDA** SDHC_WP CLK10 BRGO3	IO
GPIO3_14	General Purpose Input/Output	**SPI_SOUT** SDHC_CLK_SYNC_OUT	IO
GPIO3_15	General Purpose Input/Output	**SPI_SIN** SDHC_CLK_SYNC_IN	IO
GPIO3_16	General Purpose Input/Output	**SPI_SCK** SDHC_GATE_IN	IO
GPIO3_17	General Purpose Input/Output	**SPI_PCS0** SDHC_DAT4 SDHC_VS	IO
GPIO3_18	General Purpose Input/Output	**SPI_PCS1** SDHC_DAT5 SDHC_CMD_DIR	IO
GPIO3_19	General Purpose Input/Output	**SPI_PCS2** SDHC_DAT6 SDHC_DAT0_DIR	IO
GPIO3_20	General Purpose Input/Output	**SPI_PCS3** SDHC_DAT7 SDHC_DAT123_DIR	IO
GPIO3_21	General Purpose Input/Output	**SDHC_CMD**	IO
GPIO3_22	General Purpose Input/Output	**SDHC_DAT0**	IO
GPIO3_23	General Purpose Input/Output	**SDHC_DAT1**	IO
GPIO3_24	General Purpose Input/Output	**SDHC_DAT2**	IO
GPIO3_25	General Purpose Input/Output	**SDHC_DAT3**	IO
GPIO3_26	General Purpose Input/Output	**SDHC_CLK**	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
GPIO3_27	General Purpose Input/Output	**IRQ03** TDMB_TSYNC UC3_RTSB_TXEN	IO
GPIO3_28	General Purpose Input/Output	**IRQ04** TDMA_RXD UC1_RXD7 TDMA_TXD	IO
GPIO3_29	General Purpose Input/Output	**IRQ05** TDMA_RSYNC UC1_CTSB_RXDV	IO
GPIO3_30	General Purpose Input/Output	**RTC**	IO
GPIO4_00	General Purpose Input/Output	**EMI1_MDC**	IO
GPIO4_01	General Purpose Input/Output	**EMI1_MDIO**	IO
GPIO4_02	General Purpose Input/Output	**USB_DRVVBUS**	IO
GPIO4_03	General Purpose Input/Output	**USB_PWRFAULT**	IO
GPIO4_04	General Purpose Input/Output	**IRQ06** TDMA_RXD_EXC TDMA_TXD UC1_TXD7	IO
GPIO4_05	General Purpose Input/Output	**IRQ07** TDMA_TSYNC UC1_RTSB_TXEN	IO
GPIO4_06	General Purpose Input/Output	**IRQ08** TDMB_RXD UC3_RXD7 TDMB_TXD	IO
GPIO4_07	General Purpose Input/Output	**IRQ09** TDMB_RSYNC UC3_CTSB_RXDV	IO
GPIO4_08	General Purpose Input/Output	**IRQ10** TDMB_RXD_EXC TDMB_TXD UC3_TXD7	IO
GPIO4_09	General Purpose Input/Output	**IRQ11**	IO
GPIO4_10	General Purpose Input/Output	**EVT9_B**	IO
GPIO4_11	General Purpose Input/Output	**EC1_RXD1**	IO
GPIO4_12	General Purpose Input/Output	**EC1_RXD0**	IO
GPIO4_13	General Purpose Input/Output	**EC1_RX_CLK**	IO
GPIO4_14	General Purpose Input/Output	**EC1_RX_DV**	IO
GPIO4_15	General Purpose Input/Output	**EC2_TXD3**	IO
GPIO4_16	General Purpose Input/Output	**EC2_TXD2**	IO
GPIO4_17	General Purpose Input/Output	**EC2_TXD1**	IO
GPIO4_18	General Purpose Input/Output	**EC2_TXD0**	IO
GPIO4_19	General Purpose Input/Output	**EC2_TX_EN**	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
GPIO4_20	General Purpose Input/Output	**EC2_GTX_CLK**	IO
GPIO4_21	General Purpose Input/Output	**EC2_GTX_CLK125**	IO
GPIO4_22	General Purpose Input/Output	**EC2_RXD3**	IO
GPIO4_23	General Purpose Input/Output	**EC2_RXD2**	IO
GPIO4_24	General Purpose Input/Output	**EC2_RXD1**	IO
GPIO4_25	General Purpose Input/Output	**EC2_RXD0**	IO
GPIO4_26	General Purpose Input/Output	**EC2_RX_CLK**	IO
GPIO4_27	General Purpose Input/Output	**EC2_RX_DV**	IO
GPIO4_28	General Purpose Input/Output	**IIC3_SCL** EVT5_B USB2_DRVVBUS BRGO4 CLK11	IO
GPIO4_29	General Purpose Input/Output	**IIC3_SDA** EVT6_B USB2_PWRFAULT BRGO1 CLK12_CLK8	IO
GPIO4_30	General Purpose Input/Output	**IIC4_SCL** EVT7_B TDMA_RQ UC1_CDB_RXER	IO
GPIO4_31	General Purpose Input/Output	**IIC4_SDA** EVT8_B TDMB_RQ UC3_CDB_RXER	IO
Power-On-Reset Configuration			
cfg_eng_use0	Power-on-Reset Configuration	**IFC_WE0_B** GPIO1_22	I
cfg_eng_use1	Power-on-Reset Configuration	**IFC_OE_B** GPIO1_26	I
cfg_eng_use2	Power-on-Reset Configuration	**IFC_WP0_B** GPIO1_27	I
cfg_gpinput0	Power-on-Reset Configuration	**IFC_AD00** GPIO1_00	I
cfg_gpinput1	Power-on-Reset Configuration	**IFC_AD01** GPIO1_01	I
cfg_gpinput2	Power-on-Reset Configuration	**IFC_AD02** GPIO1_02	I
cfg_gpinput3	Power-on-Reset Configuration	**IFC_AD03** GPIO1_03	I
cfg_gpinput4	Power-on-Reset Configuration	**IFC_AD04** GPIO1_04	I
cfg_gpinput5	Power-on-Reset Configuration	**IFC_AD05** GPIO1_05	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
cfg_gpinput6	Power-on-Reset Configuration	**IFC_AD06** GPIO1_06	I
cfg_gpinput7	Power-on-Reset Configuration	**IFC_AD07** GPIO1_07	I
cfg_ifc_te	Power-on-Reset Configuration	**IFC_TE** GPIO1_23	I
cfg_rcw_src0	Power-on-Reset Configuration	**IFC_CLE** GPIO1_25	I
cfg_rcw_src1	Power-on-Reset Configuration	**IFC_AD08** GPIO1_08	I
cfg_rcw_src2	Power-on-Reset Configuration	**IFC_AD09** GPIO1_09	I
cfg_rcw_src3	Power-on-Reset Configuration	**IFC_AD10** GPIO1_10	I
cfg_rcw_src4	Power-on-Reset Configuration	**IFC_AD11** GPIO1_11	I
cfg_rcw_src5	Power-on-Reset Configuration	**IFC_AD12** GPIO1_12	I
cfg_rcw_src6	Power-on-Reset Configuration	**IFC_AD13** GPIO1_13	I
cfg_rcw_src7	Power-on-Reset Configuration	**IFC_AD14** GPIO1_14	I
cfg_rcw_src8	Power-on-Reset Configuration	**IFC_AD15** GPIO1_15	I
Quad SPI			
QSPI_A_CS0	Chip Select	**IFC_A00** GPIO1_16	O
QSPI_A_CS1	CS1	**IFC_A01** GPIO1_17	O
QSPI_A_DATA0	DATA0	**IFC_A06** GPIO2_00 IFC_WP1_B	IO
QSPI_A_DATA1	DATA1	**IFC_A07** GPIO2_01 IFC_WP2_B	IO
QSPI_A_DATA2	DATA2	**IFC_A08** GPIO2_02 IFC_WP3_B	IO
QSPI_A_DATA3	DATA3	**IFC_A09** GPIO2_03 IFC_RB2_B IFC_CS_B4	IO
QSPI_A_DATA4 / QSPI_B_DATA0	DATA4	**IFC_PAR0** GPIO2_06	IO
QSPI_A_DATA5 / QSPI_B_DATA1	DATA5	**IFC_PAR1** GPIO2_07	IO

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
QSPI_A_DATA6 / QSPI_B_DATA2	DATA6	**IFC_PERR_B** GPIO2_16	IO
QSPI_A_DATA7 / QSPI_B_DATA3	DATA7	**IFC_CS3_B** GPIO2_11	IO
QSPI_A_DQS	DQS	**IFC_A10** GPIO2_04 IFC_RB3_B IFC_CS_B5	I
QSPI_A_SCK	SCK	**IFC_A02** GPIO1_18	O
QSPI_B_CS0	Chip Select	**IFC_A03** GPIO1_19	O
QSPI_B_CS1	CS1	**IFC_A04** GPIO1_20	O
QSPI_B_DATA0 / QSPI_A_DATA4	DATA0	**IFC_PAR0** GPIO2_06	IO
QSPI_B_DATA1 / QSPI_A_DATA5	DATA1	**IFC_PAR1** GPIO2_07	IO
QSPI_B_DATA2 / QSPI_A_DATA6	DATA2	**IFC_PERR_B** GPIO2_16	IO
QSPI_B_DATA3 / QSPI_A_DATA7	DATA3	**IFC_CS3_B** GPIO2_11	IO
QSPI_B_DQS	DQS	**IFC_A11** GPIO2_05 IFC_CS_B6	I
QSPI_B_SCK	SCK	**IFC_A05** GPIO1_21 cfg_dram_type	O
QUICC Engine			
BRGO1	Baud Rate Generator Output	**IIC3_SDA** GPIO4_29 EVT6_B USB2_PWRFAULT CLK12_CLK8	O
BRGO2 / CLK9	Baud Rate Generator Output	**IIC2_SCL** GPIO3_12 SDHC_CD_B	O
BRGO3 / CLK10	Baud Rate Generator Output	**IIC2_SDA** GPIO3_13 SDHC_WP	O
BRGO4	Baud Rate Generator Output	**IIC3_SCL** GPIO4_28 EVT5_B USB2_DRVVBUS CLK11	O
CLK10 / BRGO3	Clock	**IIC2_SDA** GPIO3_13	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
		SDHC_WP	
CLK9 / BRGO2	Clock	**IIC2_SCL** GPIO3_12 SDHC_CD_B	I
TDMA_RQ / UC1_CDB_RXER	RQ	**IIC4_SCL** GPIO4_30 EVT7_B	O
TDMB_RQ / UC3_CDB_RXER	RQ	**IIC4_SDA** GPIO4_31 EVT8_B	O
UC1_CDB_RXER / TDMA_RQ	Receive Error	**IIC4_SCL** GPIO4_30 EVT7_B	I
UC1_CTSB_RXDV	Receive Data	**IRQ05** GPIO3_29 TDMA_RSYNC	I
UC1_RTSB_TXEN	Transmit Enable	**IRQ07** GPIO4_05 TDMA_TSYNC	O
UC1_RXD7	Receive Data	**IRQ04** GPIO3_28 TDMA_RXD TDMA_TXD	I
UC1_TXD7	Transmit Data	**IRQ06** GPIO4_04 TDMA_RXD_EXC TDMA_TXD	O
UC3_CDB_RXER / TDMB_RQ	Receive Error	**IIC4_SDA** GPIO4_31 EVT8_B	I
UC3_CTSB_RXDV	Receive Data	**IRQ09** GPIO4_07 TDMB_RSYNC	I
UC3_RTSB_TXEN	Transmit Enable	**IRQ03** GPIO3_27 TDMB_TSYNC	O
UC3_RXD7	Receive Data	**IRQ08** GPIO4_06 TDMB_RXD TDMB_TXD	I
UC3_TXD7	Transmit Data	**IRQ10** GPIO4_08 TDMB_RXD_EXC TDMB_TXD	O
Time Division Multiplexing			
TDMA_RSYNC	RSYNC	**IRQ05** GPIO3_29 UC1_CTSB_RXDV	I

Table continues on the next page...

Table 3-1. LS1088A Signal Reference by Functional Block (continued)

Name	Description	Alternate Function(s)	Pin type
TDMA_RXD / TDMA_TXD	RXD	**IRQ04** GPIO3_28 UC1_RXD7	I
TDMA_RXD_EXC / TDMA_TXD	Recieve Data	**IRQ06** GPIO4_04 UC1_TXD7	I
TDMA_TSYNC	TSYNC	**IRQ07** GPIO4_05 UC1_RTSB_TXEN	I
TDMA_TXD / TDMA_RXD	Transmit Data	**IRQ04** GPIO3_28 UC1_RXD7	O
TDMA_TXD / TDMA_RXD_EXC	Transmit Data	**IRQ06** GPIO4_04 UC1_TXD7	O
TDMB_RSYNC	RSYNC	**IRQ09** GPIO4_07 UC3_CTSB_RXDV	I
TDMB_RXD / TDMB_TXD	RXD	**IRQ08** GPIO4_06 UC3_RXD7	I
TDMB_RXD_EXC / TDMB_TXD	Recieve Data	**IRQ10** GPIO4_08 UC3_TXD7	I
TDMB_TSYNC	TSYNC	**IRQ03** GPIO3_27 UC3_RTSB_TXEN	I
TDMB_TXD / TDMB_RXD	Transmit Data	**IRQ08** GPIO4_06 UC3_RXD7	O
TDMB_TXD / TDMB_RXD_EXC	Transmit Data	**IRQ10** GPIO4_08 UC3_TXD7	O
TMR			
CLK11	Clock #11	**IIC3_SCL** GPIO4_28 EVT5_B USB2_DRVVBUS BRGO4	I
CLK12_CLK8	CLK8	**IIC3_SDA** GPIO4_29 EVT6_B USB2_PWRFAULT BRGO1	I

3.3 Output Signal States During Reset

When a system reset is initiated (HRESET_B or PORESET_B sampled asserted by the chip), the chip aborts all current internal and external transactions and releases the bidirectional I/O signals to a high-impedance state. However, some signals get stable values at power-on reset.

While the the chip is in reset, it drives HRESET_B asserted and ignores most input signals (except for the reset configuration signals) and drives most of the output-only signals to an inactive state.

Note that signals associated with all potential RCW source interfaces are enabled and active while the chip is in reset (that is, while HRESET_B is being driven asserted by the chip, but after PORESET_B is deasserted). This is necessary to allow the interfaces to be used for fetching configuration information from non-volatile memory devices.

Chapter 4
Reset, Clocking, and Initialization

4.1 Reset, clocking, and initialization overview

This content describes the reset, clocking, and initialization, including a definition of the reset configuration signals and the options they select.Note that other chapters in this book may describe specific aspects of initialization for individual blocks.

The reset, clocking, and control signals provide many options for operation. Additionally, many modes are selected with reset configuration signals during a power on reset (assertion of PORESET_B) and by using the reset configuration word (RCW) functionality.

4.2 External signal descriptions

The table below summarizes the external signals described in this chapter.

Table 4-3 and Table 4-4 have detailed signal descriptions, but this table contains references to additional sections that contain more information.

Table 4-1. Reset and Control Signals Summary

Signal	I/O	Description
PORESET_B	I	Power on reset input.
HRESET_B	I/O	Hard reset input. Functions as an output during initial steps in the power on reset sequence. See Power-On Reset Sequence Overview for more information. For reset assertion to the chip, use only PORESET_B.
RESET_REQ_B	O	Reset request output. An internal block requests that PORESET_B be asserted.
TA_TMP_DETECT_B	I	Tamper Detect.
TA_BB_TMP_DETECT_B	I	Low Power Tamper Detect.

Table 4-2. Clock Signals Summary

Signal	I/O	Description
SYSCLK	I	Primary clock input.
DIFF_SYSCLK	I	Differential SYSCLK (positive)
DIFF_SYSCLK_B	I	Differential SYSCLK (negative)
RTC	I	Real time clock input.
SD1_REF_CLK*n*_P/SD1_REF_CLK*n*_N SD2_REF_CLK*n*_P/SD2_REF_CLK*n*_N	I	SerDes high-speed interface reference clock *n*.
CLK_OUT	O	Diagnostic clock output.
DDRCLK	I	Reference clock for DDR controller.

The following sections describe the reset and clock signals in detail.

4.2.1 System control signals

The table below describes some of the system control signals. Power-on reset configuration describes the signals that also function as reset configuration signals.

Table 4-3. System control signals: Detailed signal descriptions

Signal	I/O	Description	
PORESET_B	I	Power on reset. Causes the chip to abort all current internal and external transactions and set all registers to their default values. PORESET_B may be asserted completely asynchronously with respect to all other signals.	
		State Meaning	Asserted/Negated-See Signal Descriptions and Power-on reset configuration for more information on the interpretation of the other signals during reset.
		Timing	Assertion/Negation-The chip data sheet gives specific timing information for this signal and the reset configuration signals.
HRESET_B	I/O	Hard reset. Causes the chip to abort all current internal and external transactions and set all registers to their default values. HRESET_B may be asserted completely asynchronously with respect to all other signals. HRESET_B is driven as an output during the first part of the power on reset sequence, after which, it becomes an input, allowing external devices to stall/hold the reset sequence. See Hard Reset Sequence (HRESET_B) for more information. For reset assertion to the chip, use only PORESET_B.	
		State Meaning	Asserted/Negated-See Signal Descriptions and Power-on reset configuration for more information on the interpretation of the other signals during reset.
		Timing	Assertion/Negation-The chip data sheet gives specific timing information for this signal.
RESET_REQ_B	O	Reset request. Indicates to the board (system in which the chip is embedded) that a condition requiring the assertion of PORESET_B has been detected.	

Table continues on the next page...

Table 4-3. System control signals: Detailed signal descriptions (continued)

Signal	I/O	Description	
		State Meaning	Asserted-An event has triggered a request for either a hard reset or a power on reset. See Reset Request Status Register (RSTRQSR1) in chapter "Reset, Clocking, and Initialization" for more information. Negated-Indicates no reset request.
		Timing	Assertion/Negation-May occur any time. Once asserted, RESET_REQ_B does not negate until PORESET_B is asserted.
ASLEEP	O	Power Management Signal. See External Signal Description	
TA_TMP_DETECT_B	I	Tamper Detect.	
TA_BB_TMP_DETECT_B	I	Low Power Tamper Detect.	

4.2.2 External Clock Signals

The table below describes some of the external clock signals of the chip.

Note that some clock signals are specific to modules within the chip, and although some of their functionality is described here, they are defined in detail in their respective chapters.

Table 4-4. Clock External Signals-Detailed Signal Descriptions

Signal	I/O	Description	
SYSCLK	I	System clock (SYSCLK). SYSCLK is the primary clock input to the chip.	
		Timing	Assertion/Negation-See the chip data sheet for specific timing information for this signal.
DIFF_SYSCLK, DIFF_SYSCLK_B	I	Differential SYSCLK (positive)/Differential SYSCLK (negative). These signals provide the clock to System Clock associated PLL's and if programmed to DDR PLL.	
		Timing	Assertion/Negation-See the chip data sheet for specific timing information for this signal.
RTC	I	Real time clock. May be used (optionally) to clock the time base of the cores. The RTC timing specifications are given in the chip data sheet . .	
		Timing	Assertion/Negation-See the chip data sheet for specific timing information for this signal.
SD*n*_REF_CLK*n*_P, SD*n*_REF_CLK*n*_N	I	SerDes high-speed interface differential reference clocks. These differential clock inputs are used to independently clock the banks/ports of high-speed differential signal lanes available on the chip. The SerDes reference clock timing specifications are given in the chip data sheet . See Valid Reference Clocks and PLL Configurations for SerDes Protocols .	
		Timing	Assertion/Negation-See the chip data sheet for specific timing information for these signals.
CLK_OUT	O	Diagnostic clock output. This output may be configured to offer one of a variety of internal system clocks to external hardware for diagnostic or debug purposes.	
DDRCLK	I	DDR controller complex clock. DDRCLK is the reference clock to the DDR PLL.	

4.3 Reset register descriptions

The table below shows the memory-mapped and register definition of the Reset module and lists the offset, name, and a cross-reference to the complete description of each register. These registers only support 32-bit accesses.

4.3.1 RST_CCSR memory map

RST base address: 1E6_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Reset Control Register (RSTCR)	32	RW	0000_0000h
4h	Service Processor Reset Control Register (RSTCRSP)	32	RW	0000_0000h
10h	Reset Request Mask Register (RSTRQMR1)	32	RW	0002_0000h
18h	Reset Request Status Register (RSTRQSR1)	32	W1C	0000_0000h
20h	Reset Request WDT Mask Register (RSTRQWDTMRL)	32	RW	0000_0000h
30h	Reset Request WDT Status Register (RSTRQWDTSRL)	32	W1C	0000_0000h
60h	Boot Release Register (BRRL)	32	RW	0000_0000h
90h	Core Enable Boot Release Status Register (BRCORENBR)	32	RO	0000_0000h
100h	RCW Request Register (RCW_REQR)	32	RO	0000_0000h
104h	RCW Completion Register (RCW_COMPLETIONR)	32	RW	0000_0000h
110h	PBI Request Register (PBI_REQR)	32	RO	0000_0000h
114h	PBI Completion Register (PBI_COMPLETIONR)	32	RW	0000_0000h
400h - 41Ch	Core Reset Status Register n (CRSTSR0 - CRSTSR7)	32	W1C	0000_0000h
420h - 42Ch	Core Reset Status Register n (CRSTSR8 - CRSTSR11)	32	W1C	0000_0000h
A00h	QMBM Warm Reset Control Register (QMBM_WARMRST)	32	RW	0000_0000h
A04h	SOC_IP Warm Reset Request Register (SOCIP_WARMRST)	32	RU	Table 4-4
BF8h	IP Block Revision 1 Register (IP_REV1)	32	RO	0281_0500h
BFCh	IP Block Revision 2 Register (IP_REV2)	32	RO	Table 4-4

4.3.2 Reset Control Register (RSTCR)

4.3.2.1 Offset

Register	Offset
RSTCR	0h

4.3.2.2 Function

The RSTCR contains bits that allows software to force assertion of RESET_REQ_B. External hardware may then decide to issue the desired reset signal (PORESET_B or HRESET_B) to the device.

4.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											Reserved	Reserved		RESET_REQ	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.2.4 Fields

Field	Function
31-5 —	Reserved
4 —	Reserved
3-2 —	Reserved
1 RESET_REQ	Hardware reset request

Table continues on the next page...

Field	Function
	External hardware may then decide to issue the desired reset signal (PORESET_B or HRESET_B) to the device. 0b - No reset request initiated. 1b - Hardware reset request initiated by software.
0 —	Reserved

4.3.3 Service Processor Reset Control Register (RSTCRSP)

4.3.3.1 Offset

Register	Offset
RSTCRSP	4h

4.3.3.2 Function

The RSTCRSP contains bits that allows Service Processor to force assertion of RESET_REQ_B. External hardware may then decide to issue the desired reset signal (PORESET_B or HRESET_B) to the device.

4.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														RESET_REQ	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.3.4 Fields

Field	Function
31-2 —	Reserved
1 RESET_REQ	Hardware reset request External hardware may then decide to issue the desired reset signal (PORESET_B or HRESET_B) to the device. 0b - No reset request initiated. 1b - Hardware reset request initiated by software.
0 —	Reserved

4.3.4 Reset Request Mask Register (RSTRQMR1)

4.3.4.1 Offset

Register	Offset
RSTRQMR1	10h

4.3.4.2 Function

The RSTRQMR1 contains mask bits for optional masking of RESET_REQ_B sources to prevent generation of such a reset request. Excludes core watchdog timer sources.

4.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						Reserved	Reserved	Reserved	Reserved	Reserved	SRDS_RST_MSK	RPTOE_MSK	PBSTE_MSK	MBEE_MSK	SDC_MSK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	SFP_MSK	SP_MSK	Reserved									Reserved			MC_MSK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.4.4 Fields

Field	Function
31-26 —	Reserved
25 —	Reserved
24 —	Reserved
23 —	Reserved
22 —	Reserved
21 —	Reserved
20 SRDS_RST_MSK	SerDes reset request event mask 0b - SerDes reset event can cause a reset request 1b - SerDes reset event cannot cause a reset request
19 RPTOE_MSK	RCPM Time Out reset request event mask 0b - RCPM Time Out event can cause a reset request 1b - RCPM Time Out event cannot cause a reset request
18 PBSTE_MSK	Reserved 0b - POR BIST error event can cause a reset request 1b - POR BIST error event cannot cause a reset request

Table continues on the next page...

Field	Function
17 MBEE_MSK	Multi-bit ECC error reset request mask 0b - Multi-bit ECC error event can cause a reset request 1b - Multi-bit ECC error event cannot cause a reset request. MBEE_MSK field must be preset to 1 to not cause a reset request due to multi-bit ECC error on boot due to some IP reading memories before they are initialized.
16 SDC_MSK	Security Debug Controller error reset request mask 0b - SDC error event can cause a reset request 1b - SDC error event cannot cause a reset request
15 —	Reserved
14 SFP_MSK	Security Fuse Processor error during POR fuse process reset mask. 0b - Security Fuse Processor error event can cause a reset request 1b - Security Fuse Processor error event cannot cause a reset request
13 SP_MSK	Service Processor error reset mask. 0b - SP error event can cause a reset request 1b - SP error event cannot cause a reset request
12-4 —	Reserved
3-1 —	Reserved
0 MC_MSK	Management Complex Debug Reset Request Mask 0b - Management Complex Debug error event can cause a reset request 1b - Management Complex Debug error event cannot cause a reset request

4.3.5 Reset Request Status Register (RSTRQSR1)

4.3.5.1 Offset

Register	Offset
RSTRQSR1	18h

4.3.5.2 Function

The RSTRQSR contains status bits to record the reasons for RESET_REQ_B assertion. The bits here are set independent of the RSTRQMR. This means if a reset occurs and is masked by RSTRQMR, it will still be recoreded in RSTRQSR. Excludes core watchdog timer sources).

NOTE

For the different sources captured in this register, some of these can be serviced with either PORESET_B or HRESET_B and some of these must be serviced with PORESET_B only.

4.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						Reserved	Reserved	MC_WDT_RR	TZONE_WDT_RR	SP_WDT_R R	SRDS_RST_R R	RPTOE_R R	PBSTE_R R	MBEE_R R	SDC_R R
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SECM_R R	SFP_R R	SP_R R	Reserved	SW_R R	WDT_RR	IFC_R R	Reserved					Reserved			MC_RR
W	W1C	W1C	W1C		W1C	W1C	W1C									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.5.4 Fields

Field	Function
31-26 —	Reserved
25 —	Reserved
24 —	Reserved
23 MC_WDT_RR	MC Processor reset request requires device level PORESET_B or HRESET_B. 0b - MC WDOG has not caused a reset request 1b - Reset request caused by MC WDOG
22 TZONE_WDT_RR	TrustZone WDOG reset request requires device level PORESET_B or HRESET_B. 0b - TrustZone WDOG reset request event not active 1b - SerDes reset request event active

Table continues on the next page...

Field	Function
21 SP_WDT_RR	Service Processor WDOG reset request . 0b - SP reset request event not active 1b - SP reset request event active
20 SRDS_RST_RR	SerDes reset event. Occurs if any enabled SerDes PLL does not lock. 0b - SerDes reset request event not active 1b - SerDes reset request event active
19 RPTOE_RR	RCPM Time Out reset request event RCPM Time Out event (for core halt, core stop, or core reset request) requires device level PORESET_B or HRESET_B. 0b - RCPM Time Out event reset request event not active 1b - RCPM Time Out event reset request event active
18 PBSTE_RR	POR BIST Error reset request POR BIST error (either for memory/array initialization or for memory/array diagnostics during POR) requires device level PORESET_B or HRESET_B. 0b - POR BIST error reset request event not active 1b - POR BIST error reset request event active
17 MBEE_RR	Multi-bit ECC reset request Platform internal memory multi-bit ECC error requires device level PORESET_B or HRESET_B. 0b - Multi-bit ECC error reset request event not active 1b - Multi-bit ECC error reset request event active
16 SDC_RR	Security Debug Controller reset request Security Debug Controller requires device level PORESET_B . 0b - SDC reset request event not active 1b - SDC reset request event active
15 SECM_RR	Security Monitor violation during POR fuse process caused reset request Security Monitor reached Hard Fail state and requires device level PORESET_B . 0b - Security Monitor reset request event not active 1b - Security Monitor reset request event active
14 SFP_RR	Security Fuse Processor error during POR fuse process caused reset request Security Fuse Processor error requires device level PORESET._B 0b - Security Fuse Processor reset request event not active 1b - Security Fuse Processor reset request event active
13 SP_RR	Service Processor reset request requires device level PORESET_B or HRESET_B. Note that the 'w1c" to this bit only takes effect when RSCTRSP[RESET_REQ] bit is clear 0b - SP has not caused a reset request 1b - Reset request caused by SP
12 —	Reserved
11 SW_RR	Software settable reset requested (See Reset Control Register (RSTCR).) 0b - Software reset request event not active 1b - Software reset request event active
10 WDT_RR	IFC reset request requires device level PORESET_B or HRESET_B. 0b - WDT reset request event not active 1b - WDT reset request event active

Table continues on the next page...

Field	Function
9 IFC_RR	IFC reset request requires device level PORESET_B or HRESET_B. 0b - IFC reset request event not active 1b - IFC reset request event active
8-4 —	Reserved
3-1 —	Reserved
0 MC_RR	Management Complex Debug Reset Request 0b - Management Complex Debug Reset Request event not active 1b - Management Complex Debug Reset Request event active

4.3.6 Reset Request WDT Mask Register (RSTRQWDTMRL)

4.3.6.1 Offset

Register	Offset
RSTRQWDTMRL	20h

4.3.6.2 Function

The RSTRQWDTMRL contains mask bits for optional masking of RESET_REQ_B generation on core thread watchdog timer expiration from the individual core threads. Excludes device source.

.

4.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								WDT_MSK7	WDT_MSK6	WDT_MSK5	WDT_MSK4	WDT_MSK3	WDT_MSK2	WDT_MSK1	WDT_MSK0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.6.4 Fields

Field	Function
31-8 —	Reserved
7 WDT_MSK7	Core 7 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
6 WDT_MSK6	Core 6 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
5 WDT_MSK5	Core 5 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
4 WDT_MSK4	Core 4 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
3 WDT_MSK3	Core 3 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
2 WDT_MSK2	Core 2 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
1 WDT_MSK1	Core 1 WDT Reset Request Mask. 0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request
0	Core 0 WDT Reset Request Mask.

Field	Function
WDT_MSK0	0b - Associated WDT can cause a reset request 1b - Associated WDT cannot cause a reset request

4.3.7 Reset Request WDT Status Register (RSTRQWDTSRL)

4.3.7.1 Offset

Register	Offset
RSTRQWDTSRL	30h

4.3.7.2 Function

The RSTRQWDTSRL contains status bits to record the reasons for RESET_REQ_B assertion due to a Core thread watchdog timer expired.

4.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								WDT_RR7	WDT_RR6	WDT_RR5	WDT_RR4	WDT_RR3	WDT_RR2	WDT_RR1	WDT_RR0
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.7.4 Fields

Field	Function
31-8 —	Reserved
7 WDT_RR7	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
6 WDT_RR6	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
5 WDT_RR5	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
4 WDT_RR4	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
3 WDT_RR3	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
2 WDT_RR2	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
1 WDT_RR1	WDT Reset Request Status. 0b - Associated WDT configured for WRC/WRS=10 has not caused a reset request 1b - Associated WDT configured for WRC/WRS=10 has caused a reset request
0 WDT_RR0	WDT Reset Request Status. 0b - Associated WDT configured for has not caused a reset request 1b - Associated WDT configured for has caused a reset request

4.3.8 Boot Release Register (BRRL)

4.3.8.1 Offset

Register	Offset
BRRL	60h

4.3.8.2 Function

The BRR contains control bits for enabling boot for each core. On existing HRESET or PORESET, the RCW BOOT_HO field optionally allows for logical core 0 to be released for booting or to remain in boot hold off. All other cores remain in boot hold off until their corresponding bit is set.

NOTE

If you change a bit from 1 to 0 outside of warm reset, results are boundly undefined for that core.

4.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.8.4 Fields

Field	Function
31-12 —	Reserved
11 CR11	Core 11 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
10 CR10	Core 10 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
9 CR9	Core 9 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
8 CR8	Core 8 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting

Table continues on the next page...

Field	Function
7 CR7	Core 7 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
6 CR6	Core 6 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
5 CR5	Core 5 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
4 CR4	Core 4 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
3 CR3	Core 3 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
2 CR2	Core 2 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting
1 CR1	Core 1 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released folr Booting
0 CR0	Core 0 Release. 0b - Core is in Boot Holdoff and not released for Booting 1b - Core released for Booting

4.3.9 Core Enable Boot Release Status Register (BRCORENBR)

4.3.9.1 Offset

Register	Offset
BRCORENBR	90h

4.3.9.2 Function

The BRCORENBR contains the status bits to indicate the RCW Boot Release location.

4.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										CORE_NBR					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.9.4 Fields

Field	Function
31-6 —	Reserved
5-0 CORE_NBR	Core Number This field designates which core will begin booting out of reset if enabled using RCW[BOOT_HO] (which refers to this core as 'core X' 000000b - Core 0 can be release for booting on existing reset(BRR is loaded at reset with 32'h0000_0001) 000001b - Core 1 can be release for booting on existing reset(BRR is loaded at reset with 32'h0000_0002) 000010b - Core 2 can be release for booting on existing reset(BRR is loaded at reset with 32'h0000_0004) 000011b - Core 3 can be release for booting on existing reset(BRR is loaded at reset with 32'h0000_0008) 011110b - Core 30 can be release for booting on existing reset(BRR is loaded at reset with 32'h4000_0000) 011111b - Core 31 can be release for booting on existing reset(BRR is loaded at reset with 32'h8000_0000)

4.3.10 RCW Request Register (RCW_REQR)

4.3.10.1 Offset

Register	Offset
RCW_REQR	100h

4.3.10.2 Function

The RCW_REQR contains the status bit. The Service Processor polls this bit to know when it can begin the RCW loading.

4.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															RQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.10.4 Fields

Field	Function
31-1 —	Reserved
0 RQ	Read only bit set by the RST logic. 0b - no request to the Service Processor for RCW loading. 1b - request to the Service Processor for RCW loading

4.3.11 RCW Completion Register (RCW_COMPLETIONR)

4.3.11.1 Offset

Register	Offset
RCW_COMPLETIONR	104h

4.3.11.2 Function

The RCW_COMPLETIONR contains the control bits written by the Service Processor to indicate that it finished the RCW loading.

4.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ERR_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DONE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.11.4 Fields

Field	Function
31-24 —	Reserved
23-16 ERR_CODE	RCW error code.
15-1 —	Reserved
0 DONE	RCW done bit. 0b - Service Processor has not yet completed the RCW loading 1b - Service Processor has completed the RCW loading

4.3.12 PBI Request Register (PBI_REQR)

4.3.12.1 Offset

Register	Offset
PBI_REQR	110h

4.3.12.2 Function

The PBI_REQR contains the status bit. The Service Processor can poll this bit to know when it can begin the PBI phase.

4.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															RQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.12.4 Fields

Field	Function
31-1 —	Reserved
0 RQ	Read only bit set by the RST logic. This bit will automatically be cleared by hardware after the PBI phase is done. 0b - no request to the Service Processor for PBI phase. 1b - request to the Service Processor for PBI phase

4.3.13 PBI Completion Register (PBI_COMPLETIONR)

4.3.13.1 Offset

Register	Offset
PBI_COMPLETIONR	114h

4.3.13.2 Function

The PBI_COMPLETIONR contains the control bits written by the Service Processor to indicate that it finished the PBI phase.

4.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								ERR_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DONE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.13.4 Fields

Field	Function
31-24 —	Reserved
23-16 ERR_CODE	PBI error code. 00000000b - Service Processor is in Boot Holdoff and not released for Booting 00000001b - Service Processor released for Booting
15-1 —	Reserved
0 DONE	PBI Completion Register (PBI_COMPLETIONR) PBI done bit. 0b - Service Processor has not yet completed the PBI phase 1b - Service Processor has complete the PBI phase

4.3.14 Core Reset Status Register n (CRSTSR0 - CRSTSR7)

4.3.14.1 Offset

For a = 0 to 7:

Register	Offset
CRSTSRa	400h + (a × 4h)

4.3.14.2 Function

The CRSTSRn contains the reset status bits for each core

NOTE

The number of CRSTSRn registers is determined by the number of physical cores.

Reset of this Register

A power-on reset of the device causes the following to occur:

- RST_PORST is set
- All other bits are cleared

A hard reset of the device causes the following to occur:

- RST_PORESET remains unchanged (PORST resources are not affected by Hard Reset)
- RST_HRESET is set
- All other bits are cleared

If an application prefers that RST_PORST is not set after hard reset, then a write-1-clear must be done to CRSTSRn[RST_PORST] upon exiting power-on reset.

Ready Bit Functionality

This bit is cleared on a device power-on or hard reset. Upon completion of power-on or hard reset processing, this bit may be automatically set for a core if none of the conditions specified in the READY bit definition are true.

4.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												Reserved	READY	RST_HRST	RST_PORST
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.14.4 Fields

Field	Function
31-4 —	Reserved
3 —	Reserved
2 READY	Core ready pin Core is in the 'ready' state after device has successfully passed through the "System Ready" point and the core is currently not in any of the following states: • Core PH10 state • Core PH15 state 0b - Core is not ready 1b - Core is ready
1 RST_HRST	Core was reset due to an HRESET (note a PORESET causes an HRESET, but a PORESET will not cause this bit to be set)
0 RST_PORST	Core was reset due to a PORESET

4.3.15 Core Reset Status Register n (CRSTSR8 - CRSTSR11)

4.3.15.1 Offset

Register	Offset
CRSTSR8	420h
CRSTSR9	424h
CRSTSR10	428h
CRSTSR11	42Ch

4.3.15.2 Function

The CRSTSRn contains the reset status bits for each core

NOTE

The number of CRSTSRn registers is determined by the number of physical cores.

Reset of this Register

A power-on reset of the device causes the following to occur:

- RST_PORST is set
- All other bits are cleared

A hard reset of the device causes the following to occur:

- RST_PORESET remains unchanged (PORST resources are not affected by Hard Reset)
- RST_HRESET is set
- All other bits are cleared

If an application prefers that RST_PORST is not set after hard reset, then a write-1-clear must be done to CRSTSRn[RST_PORST] upon exiting power-on reset.

Ready Bit Functionality

This bit is cleared on a device power-on or hard reset. Upon completion of power-on or hard reset processing, this bit may be automatically set for a core if none of the conditions specified in the READY bit definition are true.

4.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				RST_WDT				Reserved				RST_MPIC			
W					W1C								W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				RST_CORE				Reserved				Reserved	READY	RST_HRST	RST_PORST
W					W1C										W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.15.4 Fields

Field	Function
31-28 —	Reserved
27-24 RST_WDT	Threaded mpic 0000b - threaded mpic 0001b - threaded wdt
23-20 —	Reserved
19-16 RST_MPIC	Threaded MPICr 0000b - threaded mpic 0001b - threaded mpic
15-12 —	Reserved
11-8 RST_CORE	Core was reset in response to internal core request to reset itself by setting bit DBCR0[RST] register 0000b - No reset 0001b - Reset occurred
7-4 —	Reserved
3 —	Reserved
2 READY	Core ready pin Core is in the 'ready' state after device has successfully passed through the "System Ready" point and the core is currently not in any of the following states:

Table continues on the next page...

Field	Function
	• Core PH10 state • Core PH15 state 0b - Core is not ready 1b - Core is ready
1 RST_HRST	Core was reset due to an HRESET (note a PORESET causes an HRESET, but a PORESET will not cause this bit to be set)
0 RST_PORST	Core was reset due to a PORESET

4.3.16 QMBM Warm Reset Control Register (QMBM_WARMRST)

4.3.16.1 Offset

Register	Offset
QMBM_WARMRST	A00h

4.3.16.2 Function

Software writes bit location to initiate warm reset to the QBMan.

Software polls bit location to know when warm reset has completed.

NOTE

Before initiating warm reset software must:

Shut down software portal interfaces (stop de-queueing new work from QMAN and stops enqueueing new work to QMAN).

NOTE

After setting the warm reset bits software must:

Poll QMBM_WARMRST to know when warm reset has completed.

Re-initialize QMAN/BMAN. This may include clearing out any history it has of QMAN portal ring state as well as forgetting about buffers it may not have previously released.

For more details please refer the DPAA2 Reference Manual.

4.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															BMQM_WARMRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3.16.4 Fields

Field	Function
31-1 —	Reserved
0 BMQM_WARMRST	BMQM Warm Reset Write 1 to initiate BMan warm reset. Write 0 has no effect. Read of 1 indicates BMan warm reset in progress. 0b - No reset request 1b - Reset request occurs

4.3.17 SOC_IP Warm Reset Request Register (SOCIP_WARMRST)

4.3.17.1 Offset

Register	Offset
SOCIP_WARMRST	A04h

4.3.17.2 Function

Software writes bit location to initiate warm reset to the QBMan.

Software polls bit location to know when warm reset has completed.

NOTE

Before initiating warm reset software must:

Shut down software portal interfaces (stop de-queueing new work from IP and stops enqueueing new work to IP).

Quiesce IP

NOTE

After setting the warm reset bits software must:

Poll SOCIP_WARMRST to know when warm reset has completed.

Re-initialize IP. This may include clearing out any history it has of QMAN portal ring state as well as forgetting about buffers it may not have previously released.

Restart IP.

4.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

4.3.17.4 Fields

Field	Function
31-0 —	Reserved

4.3.18 IP Block Revision 1 Register (IP_REV1)

4.3.18.1 Offset

Register	Offset
IP_REV1	BF8h

4.3.18.2 Function

The IP_REV1 Register contains the block revision fields.

4.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IP_MJ								IP_MN							
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

4.3.18.4 Fields

Field	Function
31-16	Block ID

Table continues on the next page...

Field	Function
IP_ID	0001.
15-8 IP_MJ	Major revision 04.
7-0 IP_MN	Minor revision 01.

4.3.19 IP Block Revision 2 Register (IP_REV2)

4.3.19.1 Offset

Register	Offset
IP_REV2	BFCh

4.3.19.2 Function

The IP_REV2 Register contains the block revision fields.

4.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								IP_INT							
W																
Reset	u	u	u	u	u	u	u	u	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								IP_CFG							
W																
Reset	u	u	u	u	u	u	u	u	0	0	0	0	0	0	0	0

4.3.19.4 Fields

Field	Function
31-24 —	Reserved
23-16 IP_INT	Block ID 0001.
15-8 —	Reserved
7-0 IP_CFG	IP block configuration ID (option).

4.4 CGU core cluster register descriptions

4.4.1 CGU Core to Hardware Accelerator (HWA) Block Cluster Group memory map

CGU_CGA base address: 130_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	HWA1CSR (HWA1CSR)	32	RW	0000_0000h
30h	HWA2CSR (HWA2CSR)	32	RW	0000_0000h
50h	HWA3CSR (HWA3CSR)	32	RW	0000_0000h

4.4.2 HWAaCSR (HWA1CSR - HWA3CSR)

4.4.2.1 Offset

Register	Offset
HWA1CSR	10h
HWA2CSR	30h

Table continues on the next page...

Register	Offset
HWA3CSR	50h

4.4.2.2 Function

The HWA1-3 CSR registers control the clock frequency selection for each clock generator's hardware accelerators clock muxes.

4.4.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	HWACLKSEL				0										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.4.2.4 Fields

Field	Function
31 —	Reserved
30-27 HWACLKSEL	HWACLKSEL HWA1CSR 0000 - Sync mode 0001 - Cluster A/B PLL1 divide-by-1 0010 - ClusterA/B PLL1 divide-by-2 0011 - Cluster A/B PLL1 divide-by-3 0100 - Cluster A/B PLL1 divide-by-4 0101 - HWAMUX1 External Clock Source 0110 - Cluster PLL2 divide-by-2 0110 - Cluster PLL2 divide-by-3 0111 - 1111 - reserved

Table continues on the next page...

Field	Function
	HWA2CSR 0000 - Sync mode 0001 - Cluster A/B PLL2 divide-by-1 0010 - Cluster A/B PLL2 divide-by-2 0011 - Cluster A/B PLL2 divide-by-3 0100 - Cluster A/B PLL2 divide-by-4 0101 - HWAMUX2 External Clock Source 0110 - Cluster PLL1 divide-by-2 0110 - Cluster PLL1 divide-by-3 0111 - 1111 - reserved HWA3CSR 0000 - Sync mode 0101 - HWAMUX3 External Clock Source 0110 - Cluster PLL2 divide-by-2 0110 - Cluster PLL2 divide-by-3 0111 - 1111 - reserved
26-0 —	Reserved

4.5 CGU Platform register descriptions

4.5.1 CGU Platform CCSR memory map

CGU Platform base address: 136_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Platform Clock Control/Status Register (CLKCSR)	32	RW	0000_0000h

4.5.2 Platform Clock Control/Status Register (CLKCSR)

4.5.2.1 Offset

Register	Offset
CLKCSR	0h

4.5.2.2 Function

CLKCSR selects which signal to observe on the CLKOUT1 pad.

4.5.2.3 Diagram

4.5.2.4 Fields

Field	Function
31-17 —	Reserved
16 CLKOEN1	CLKOEN1 CLKOEN1. CLKOUT1 pad enable. 0b - Tristate CLKOUT1 pad 1b - Enable CLKOUT1 pad
15-11 CLKOSEL1	CLKOSEL1[4:0] Select signal for observation 00000-10111b - Reserved

Table continues on the next page...

Field	Function
	11000b - pll_vco_out_div8 11001b - pll_vco_out_div6 11010b - pll_clk_out_div4 11011b - pll_clk_out_div2 11100b - pll_clk_out (same as raw_clk) 11101b - pll_vco_bufout 11110b - pll_fdbk_bufout 11111b - pll_sys_bufout
10-9 CLKODIV1	CLKODIV1[1:0] Select division setting for clockout mux output to reduce the frequency of the signal to a more observable/measurable range. 01b - divide-by-2 10b - divide-by-4 11b - divide-by-8
8-0 —	Reserved

4.6 CGU DDR register descriptions

4.6.1 CGU DDR CCSR memory map

CGU_DDR base address: 138_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	DDR Clock Control/Status Register (CLKCSR)	32	RW	0000_0000h

4.6.2 DDR Clock Control/Status Register (CLKCSR)

4.6.2.1 Offset

Register	Offset
CLKCSR	0h

4.6.2.2 Function

CLKCSR controls the DDR PLL's phase divider option.

4.6.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	CLKSEL				0										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.6.2.4 Fields

Field	Function
31 —	Reserved
30-27 CLKSEL	CLKSEL[3:0] CGU feed forward divider select. The DDR PLL phase clocks may select from different divider options depending on the DDR target frequency. 0000b - no divide 0001b - divide-by-2 0010-1111b - reserved
26-0 —	Reserved

4.7 CORE CLUSTER UNIT CCU register descriptions

4.7.1 CORE CLUSTER UNIT CCU memory map

CCU base address: 137_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Core Cluster 1 Clock Control/Status Register (CLKC1CSR)	32	RW	0000_0000h
20h	Core Cluster 2 Clock Control/Status Register (CLKC2CSR)	32	RW	0000_0000h

4.7.2 Core Cluster a Clock Control/Status Register (CLKC1CSR - CLKC2CSR)

4.7.2.1 Offset

Register	Offset
CLKC1CSR	0h
CLKC2CSR	20h

4.7.2.2 Function

The CLK1-2 CSR registers control the clock frequency selection for each clock generator's core cluster clock mux. CLK1CSR corresponds to core cluster 1.

4.7.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	CLKSEL				0										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.7.2.4 Fields

Field	Function
31 —	Reserved
30-27 CLKSEL	CLKSEL[3:0] Selects the clock source for this clock domain. Default value set via RCW. 0000 - Cluster PLL1 output 0001 - Cluster PLL1 divide-by-2 0010 - Cluster PLL1 divide-by-4 0011 - Reserved 0100 - Cluster PLL2 output 0101 - Cluster PLL2 divide-by-2 0110 - Cluster PLL2 divide-by-4 0111 - Reserved For two PLLs per cluster: 0111 - 1111 - Reserved 0000b - Cluster PLL1 output 0001b - Cluster PLL1 divide-by-2 0010b - Cluster PLL1 divide-by-2 0011b - Reserved 0100b - Cluster PLL2 output 0101b - Cluster PLL2 divide-by-2 0110b - Cluster PLL2 divide-by-4 0111b - Reserved 0111-1111b - Reserved
26-0 —	Reserved

4.8 Functional description

This section describes the various ways to reset the chip, the POR sequence, the hard reset sequence, the POR configurations, the reset configuration word (RCW), and the clocking on the device.

4.8.1 Reset Operations

The platform supports several inputs to reset the platform logic and/or core(s):

- Power-on reset (PORESET_B) of entire device

- External hard reset (HRESET_B) of entire device
- Core watchdog timer expiration from each core with specific WRS encoding
- Individual core reset requests from the respective cores
- Interrupt controller reset request of a core or cores initiated by software write to PIC PIR register

4.8.2 Reset Actions

The reset control logic determines the cause of the reset, synchronizes it if necessary, and resets the appropriate internal hardware. Each type of reset has a different impact on the device logic:

- Power-On Reset has the greatest impact, sampling the POR configuration pins , resetting the entire device, including all clocking logic PLLs and fuse-related logic.
- Hard Reset resets the entire device, with the exception of:
  - PLLs in some cases
  - fuse-related logic in all cases
- Warm Device Reset is very similar to Hard Reset but reapplies the repair information across the chip since portions have had power removed
- Other sources of reset mentioned simply cause a reset of the state of a given core. Registers that store fuse related information for that core are not reset.

The table below identifies the reset actions for each reset source.

Table 4-5. Reset Actions

Action Taken	Reset Source			
	POERESET	External device asserts HRESET (w/o POERESET asserted)	Internal signal requests Warm Device Reset w/ Fuse Restore	Second Watchdog timer expiration resulting in either an interrupt request or a software issued reset request
POR config signals sampled (includes RCW source configuration re-sampled)	Y	N		N
Fuses read	Y	N	N	N
PLLs reset and locked/re-locked	Y	N	Y	N
Platform logic reset including all configuration registers	Y	Y (though some Device Configuration block status registers and DCSR registers not reset)		N
Reset Config Word (RCW) data loaded/reloaded	Y	Y		N
HRESET_B driven	Y			N
Reset to core	Y			Y

4.8.3 Sources of Reset Requests

Reasons that the RESET_REQ_B signal may be asserted include:

- Service Processor - RCW Phase failure while loading reset configuration word data (due to checksum error over the RCW or some other error while interacting with the memory interface that is the source of the data). The Service Processor will write an encoded error message to RCW_COMPLETION[ERRCODE] if needed to indicate one of up to N possible different errors that it encountered interacting with the memory interface or with the data structures themselves that it consumes. Such an error from the Service Processor forces the device reset sequence to halt indefinitely. A device Hard Reset is required, at a minimum.
- TZ Watchdog Timer expiration while it is configured to request device reset. A device Hard Reset is required, at a minimum.
- Software reset request of device from the Service Processor by writing the Reset block RSTCRSP register--note that Secure Boot failure causes one of these software reset requests (Internal Secure Boot Code writes this register). A device Hard Reset is required, at a minimum.
- Software reset request of device from GPPs by writing the Device Configuration block RSTCR register--note that Secure Boot failure causes one of these software reset requests (Internal Secure Boot Code writes this register). A device Hard Reset is required, at a minimum.
- Security Monitor detects a potential security breach and requests device or system level reset. A device Power-On Reset is required.
- Security Fuse Processor requests device or system level reset. A device Power-On Reset is required.
- Security Debug Controller detects an attempted security breach and requests device or system level reset. A device Power-On Reset is required.
- Multi-Bit ECC Error reported by platform compiled SRAM requests device or system level reset. If such an error is detected during the device reset sequence, the sequence is indefinitely halted. A device Hard Reset is required, at a minimum.
- Reset Controller reset request when a SerDes PLL does not lock when it is expected to lock during the device reset sequence (this is conveyed by the setting of the SRDS_PLL_PD_PLL*n* RCW field for the PLL of the given SerDes solution of the Chip). The cause for this is not supplying a required SerDes reference clock. A device Hard Reset is required.

4.8.4 Power-On Reset Sequence Overview

Assertion of the PORESET_B external signal initiates the Power-On Reset flow. PORESET_B should be asserted for at least 32 SYSCLKs after stable external power to the device is applied. Refer to the device hardware specification for more information regarding power sequencing and PORESET_B input requirements.

After the negation of PORESET_B, the reset control logic begins cycling the device through its full reset and RCW configuration process. The device asserts HRESET_B throughout the power-on reset process, including RCW configuration. Reset and RCW configuration time varies according to the configuration source and SYSCLK frequency. Initially, the RCW source POR configuration input pins are sampled to determine the configuration source. Next, the device starts loading the actual RCW data. The system PLL begins to lock according to the clock ratio/mode values communicated in the RCW data. Once the PLL locks and the RCW data is all loaded, HRESET_B is released by the device and the clocking unit starts distributing PLL outputs throughout the device. Pre-Boot Initialization is then optionally performed, and the cores are permitted to boot.

4.8.5 Power-On Reset Sequence Detailed Description

The table below contains the detailed POR sequence. In the "Resets" column:

- "P - -" indicates operations performed for Power-on Reset only
- "P - W" indicates operations performed for Power-on and Warm Device Reset only
- "PHW" indicates operations performed for Power-on, Hard, and Warm Device Reset

Table 4-6. Power-On Reset Sequence

Stage	Resets	Step
Initial Stages up to PORESET_B De-assertion		
1	P - -	Power Applied. Power is applied to comply to the *Chip Datasheet*.
2	P - -	PORESET_B Pin Assertion by External System Logic. The external system logic asserts PORESET_B (JTAG compliance pin) (assertion time requires 32 SYSCLK clocks) causing all registers to be initialized to their default states (note that the actual reset of the platform is later, just abstracting this unnecessary detail from the customer) and most I/O drivers to be tri-stated (some clock, clock enables, and system control signals are active).
		Note that both fuse processors and the regs receiving fuse values should only be reset with PORESET_B or signal generated in response to a PORESET_B assertion, as opposed to during the reset process of a Hard Reset (initiated by external device asserting HRESET).
		All JTAG state machine logic is only reset with TRST_B assertion, and not on a PORESET_B assertion. TRST_B should typically be driven asserted concurrently with PORESET_B, though it is recommended that customers control TRST_B separately from PORESET_B if boundary scan testing/operation is required.
3	P - -	SYSCLK and Configuration Inputs Applied.

Table continues on the next page...

Table 4-6. Power-On Reset Sequence (continued)

Stage	Resets	Step
		The system applies a toggling SYSCLK signal and stable POR reset configuration inputs. At this point SYSCLK is propagated throughout the device. Platform PLL is running in Bypass mode.
4	P - -	HRESET_B Pin driven by the chip. The device begins driving the bidirectional HRESET pin asserted after sampling the assertion of PORESET_B. This can be used to reset other components on the board.
5	P - -	PORESET_B Pin De-assertion by External System Logic. External system logic negates PORESET_B after its required hold time (32 SYSCLK cycles) and after POR configuration inputs have been valid for at least 4 SYSCLK cycles.
Fuse Processing		
6	P - -	Configuration Pins Sampled. The device samples the RCW source POR configuration input pins (CFG_RCW_SRC[8:0]) on de-assertion of PORESET_B to determine the RCW source. Note that these POR config pins and all others like it are only sampled on a PORESET_B.
7	P - -	Secure Fuse Processing. The fuse controller for security-related fuses reads its fuse box. The SFP (Security Fuse Processor) should only be reset during a PORESET_B and not during the reset process of a Hard Reset. Debug and DFT scan access is blocked up to this point by the scan control logic and after this point the value broadcast from the Debug Mode SFP fuse enables or disables Debug and DFT scan access.
7	PHW	The secure fuses are available for use by the Chip.
8	P - -	Repair Fuse Processing.
	P - W	The fuse controller for repair (and other non-security related items) pushes the data into the appropriate platform/core registers.
Reset Stages		
9	PHW	Reset Platform Logic. The Reset block initiates and completes the reset of the rest of the platform logic. Note that this platform reset step is the point at which the device hard reset process (HRESET) begins if an external device asserts HRESET, assuming the device is not already sequencing through the power-on reset process.
10	PHW	Enable I/O Drivers - Selected Primary RCW Interface and DDR I/Os. Some of the I/O drivers are enabled, specifically those pins associated only with the specific RCW interface selected via CFG_RCW_SRC[8:0]. Also, the DDR I/Os become enabled at this point as well (exceptions: MCKE, MCK, MODT are on from the beginning).
		The ASLEEP pin is also enabled at this point if the Reset Sequence Pause is enabled (CFG_RSP_DIS=0); otherwise ASLEEP pin is enabled later in the reset sequence.
11	PHW	Array Initialization (Optional). Array Initialization runs on the platform if enabled by the POR configuration input pin. .
12	PHW	Service Processor Released from Reset. The Service Processor is released from reset in this state (must be after array initialization) and begins executing the Boot 0 Reset Handler. Note that the Service Processor is reset for Power-on reset, Hard Reset, and Warm Device Reset.
13	PHW	RCW Loading (Optional).

Table continues on the next page...

Table 4-6. Power-On Reset Sequence (continued)

Stage	Resets	Step
		The Service Processor polls Reset RCW_REQ[REQ]. When the request is detected, it starts loading the RCW data stored in record format from the interface specified by the CFG_RCW_SRC[8:0] POR configuration pins and stores the 128 bytes of data to the RCWSR registers within the Device Configuration block. Loading time varies and may be non-deterministic, and depends on the source of the RCW. Note that if any of the hard-coded RCW options are used, this RCW loading process is effectively bypassed and the device is automatically configured according to the specific RCW field encodings pre-assigned for the given hard-coded RCW option.
14	PHW	RCW Completion - PLLs Begin to Lock. Once all 1024 bits of RCW data is loaded, the PLLs begin to lock. The Service Processor writes Reset RCW_COMPLETION[DONE] to indicate completion. For a Hard Reset that occurs without sequencing through a Power-On Reset process, the individual PLLs are not re-locked.
15	PHW	Halt the Reset Sequence on RCW Error Detection. If the Service Processor reports an error during its process of loading the RCW data the following occurs: • The device reset sequence is halted, remaining in this state. • An error code is reported by the SP in RCW_COMPLETION[ERR_CODE] • A request for a reset of the SoC is captured in RSTRQSR1[SP_RR], which generates a reset request if not masked by RSTRQMR1[SP_MSK] This state can only be exited with a PORESET or Hard Reset.
PLL Locking and Reset Pause		
16	PHW	Platform PLL Drives Platform Clock Tree. The platform clock tree is then switched over to be driven by the output of the platform PLL. ipg_syncs are driven as determined by the RCW configuration fields. The platform PLL locks using H-Tree feedback if the Test Port is enabled or synchronous core or synchronous DDR operation is configured, or if RCW explicitly configures the platform PLL for HFM. The reset sequence waits for the rising edge of SYSCLK together with ipg_syncs being phase aligned in a known, reproducible state before moving to the next state.
		Once the platform clock is driving the clock tree, the Reset block now asserts the PERIPHCLKEN pin used by the Service Processor.
Pre-Boot Initialization Stages and HRESET_B Signalling		
17	PHW	HRESET_B De-assertion. The device stops driving HRESET_B.
18	PHW	Pre-Boot Initialization. If enabled by RCW, the Service Processor polls Reset PBI_REQ[REQ]. When the request is detected, it starts Pre-Boot Initialization, reading data from either the SD/MMC,SPI, Serial NOR (QSPI), or IFC NOR interface and writing to CCSR space or local memory space (OCRAM, SRAM, DDR). If the Service Processor reports an error during its Pre-Boot Initialization process, the device reset sequence is halted. This state can only be cleared with a PORESET or Hard Reset. Note that non-CCSR-mapped memory space of a block on HSIO must not be the target of any Pre-Boot Initialization (since SERDES block is not running). If the Service Processor reports an error during its process of Preboot Initialization, the following occurs: • The device reset sequence (Reset FSM) is halted, remaining in this state.

Table continues on the next page...

Table 4-6. Power-On Reset Sequence (continued)

Stage	Resets	Step
		• An error code is reported by the SP in PBI_COMPLETION[ERR_CODE]. • A request for a request of the SoC is captured in RSTRQSR1[SP_RR], which generates a reset request if not masked by RSTRQMR1[SP_MSK]. This state can only be exited with a PORESET or Hard Reset.
19	PHW	Wait for HRESET_B De-assertion from External Sources. Any external device optionally driving HRESET_B negates it if not done earlier. If other external devices do not release HRESET_B, the device reset sequence stalls at this point until HRESET_B is released.
SERDES, TimeZero, Further Initialization, and System Ready		
20	PHW	SERDES Released from Reset. All SERDES blocks are released from reset and the SERDES PLLs are locked provided the SRDS_EN RCW bit is a logic 1. Note that once the cores are permitted to boot, software should check the RST_DONE bit within the SERDES Control block SRDSBnRSTCLT registers to ensure that the SERDES PLLs have successfully locked as expected. Transactions or packet data cannot be transferred through the targeted lane(s) of the SERDES interface if the PLL associated with the lane(s) does not lock properly. Note that a SERDES PLL will not lock if the corresponding reference clock is not provided.
21	PHW	TimeZero Event. The TimeZero event is established at rising edge of SYSCLK together with ipg_syncs being phase aligned in a known, reproducible state, and the Reset block performs initialization on the Cores (by releasing core "warm" reset), HSIO, and other IP blocks. Note that a PORESET must issue a "cold" or "POR" reset of the cores, while a device Hard Reset or specific warm reset of a given core must issue a "warm" reset.
22	PHW	Completion of IP Initialization. The Reset block waits for all IPs to complete their initialization processes.
24	PHW	System Ready State. The device enters the System Ready state when all initialization processes have completed. The PCI-Express interfaces are released to accept external requests, and the GPP cores are allowed to exit reset if allowed by the Reset block Boot Release Register (BRR). The ASLEEP signal negates synchronized to a rising edge of SYSCLK, indicating the ready state of the system. This ASLEEP negation also indicates externally when the Test Port or JTAG/SAP can access and initialize main memory (this assumes the Reset Sequence Pause point wasn't used to initialize main memory and the cores are all still in holdoff). Also note, the default configuration of the EVT[0] pin allows for an "event" on that signal prior to System Ready to place all cores in debug halt state . Moving forward from this "system ready" point, the ASLEEP pin transitions to the asserted state when the device enters sleep mode.

4.8.6 Power-On Reset Sequence Timing

The following figure shows a timing diagram of the POR sequence.

Figure 4-1. Power-On Reset Sequence

4.8.7 Hard Reset Sequence (HRESET_B)

The hard reset sequence is initiated externally by asserting HRESET_B. The device continues asserting HRESET_B throughout the HRESET_B sequence. Reset and RCW configuration time varies according to the configuration source and SYSCLK frequency. The reset configuration input signals (CFG_RCW_SRC[8:0]) are not sampled on hard reset (only at Power-On Reset) and PLLs are not re-initialized or re-locked (in some configurations), so the device immediately starts loading the RCW data, and configures the device. After the configuration sequence completes, the device releases the HRESET_B signal and exits the HRESET_B state. After negation is detected, a 16-cycle period is taken before testing for the presence of an external reset. The hard reset sequence begins with reset of the device at step 9.

4.8.7.1 Selecting the RCW Source and "Flash Mode"

This table shows the strapping value required on the CFG_RCW_SRC[8:0] for selecting the desired RCW source. CFG_RCW_SRC[8:7] must always be set to 2'b00.

Table 4-7. RCW Source and Flash Mode Encodings

CFG_RCW_SRC[6:0], FLASH_MODE[6:0][1]		RCW Source, IFC Mode	Valid for:	
			RCW_SRC	FLASH_MODE
RCW Loading - Reserved Area				
7'b000_xxxx	7'b000_0000	Reserved	-	-
	7'b000_0001 - 7'b000_1111	Reserved	-	-
7'b001_xxxx	7'b001_xxxx	Hard-Coded RCW Options	Y	N
RCW Loading - from NOR Flash • All NOR Flash options require IFC_GRP_FGHI_BASE=2'b00, IFC_GRP_F_EXT=3'b000, IFC_GRP_G_EXT=3'b000, and IFC_GRP_H_EXT=3'b000.				
7'b01x_xxxx (IFC NOR Flash options)	7'b010_xxxx	NOR Flash: 16-bit Port Size Fields in IFC CSOR0 and CSPR0 registers are loaded at reset based on these settings. CFG_RCW_SRC[3] / FLASH_MODE[3] decoded: • 1'b0 - Conventional NOR (flash w/ ext latch): - IFC_AD[n-1:0] contains most signification portion of Address • 1'b1 - ADM NOR (flash w/ internal latch): - IFC_AD[n-1:0] contains least signification portion of Address CFG_RCW_SRC[2:1] / FLASH_MODE[2:1] is configured as follows: • 2'b00 - 22-bit address bus selected (not available for SoCs with 32-bit addr bits) • 2'b01 - 25-bit address bus selected (requires IFC_A_8_6=1'b0) • 2'b10 - 28-bit address bus selected (requires IFC_A_8_6=1'b0, IFC_GRP_A_BASE=2'b00) • 2'b11 - Reserved CFG_RCW_SRC[0] / FLASH_MODE[0] decoded: • Should be set the to the inverted value of CFG_RCW_SRC[3] / FLASH_MODE[3] CFG_RCW_SRC[0] / FLASH_MODE[0] will become a Reserved bit. requires this is the inverted value of CFG_RCW_SRC[3]	Y	Y
7'b01x_xxxx	Reserved			
RCW Loading - from Serial Flash (SD/MMC at offset 0x1000, QSPI at offset 0x0, I2C, Serial NOR at offset 0x0)				

Table continues on the next page...

Table 4-7. RCW Source and Flash Mode Encodings (continued)

CFG_RCW_SRC[6:0], FLASH_MODE[6:0][1]			RCW Source, IFC Mode	Valid for:	
				RCW_SRC	FLASH_MODE
7'b10x_xxxx (Group 1: Non-IFC options)	7'b100_0000		SD 2.0 Card (uses write-protect and card detect) • Requires SDHC_BASE=1'b0 and SDHC_EXT=3'b0 • Data transfers are 1-bit • Does not support voltage select control for dynamically switchable supplies or direction controls for external SD level shifters.	Y	N
	7'b100_0001		eMMC (no write-protect and no card detect) • Requires SDHC_BASE=1'b0 and SDHC_EXT=3'b0 • Data transfers are 1-bit	Y	N
	7'b100_0010		Reserved	-	-
	7'b100_0011 - 7'b100_1000		Reserved	-	-
	7'b100_1010 - 7'b101_1111		Reserved		
	7'b01x_xxxx		Reserved		
	7'b100_1001		I2C1 Extended Addressing (2 byte addressing supporting up to 64 KB) "Extended Addressing" refers to EEPROMs with a calling address of 7'b1010aaa (ROM code only supports one calling address with 'aaa' = 3'h0) which require two full bytes of data word address after the calling address.	Y	N
	7'b11x_xxxx (Group 2: Non-IFC options, QSPI)	7'b110_000x	Reserved	-	-
		7'b110_0010	Serial NOR (QSPI) 24-bit Addressing • QSPI Flash A is used. • Requires IFC_GRP_A_BASE=2'b11 • Data transfers are 1-bit	Y	Y
		7'b110_0011 - 7'b111_1111	Reserved	-	-

1. This field specifies the mode for the IFC or Serial NOR which contains the boot code. It is always used in conjunction with RCW[BOOT_LOC].

4.8.7.2 NOR Flash Configuration for RCW and Boot

The chip supports the following IFC pins: IFC_AD[15:0] and IFC_A[11:0] which support a maximum configuration of 28 address and 16 data pins.

Based on the number of IFC_AD and IFC_A pins the table below defines which modes are available . Any row containing the value "Yes" means that the mode located under "User Selection" is available for use.

Table 4-8. User NOR Flash Options (Conventional NOR)

Pinout	User Selection	Proper Configuration / Use			
28A /16 D	Mode	cfg_rcw_src / rcw[flash_mode]	ADM_SHFT	Required RCW Settings	Address - IFC Pins (left is MS portion)
Parallel NOR w/ 16-bit Data Bus Options					
Yes	22A/16D, Conv	0_0010_0001	6	(none)	AD[15:0], A[5:0]
Yes	25A/16D, Conv	0_0010_0011	9	See Reqd_RCW.25A	AD[15:0], A[8:0]
Yes	28A/16D, Conv	0_0010_0101	12	See Reqd_RCW.28A	AD[15:0], A[11:0]

The table below summarizes options and configuration for ADM NOR configurations, which internally latch the least significant portion of the address.

Table 4-9. User NOR Flash Options (ADM NOR)

Pinout	User Selection	Proper Configuration / Use			
28A / 16D	Mode	cfg_rcw_src / rcw[flash_mode]	ADM_SHFT	Required RCW Settings	Address - IFC Pins (left is MS portion)
Parallel NOR w/ 16-bit Data Bus Options					
Yes	22A/16D, ADM	0_0010_1000	16	(none)	A[5:0], AD[15:0]
Yes	25A/16D, ADM	0_0010_1010	16	See Reqd_RCW.25A	A[8:0], AD[15:0]
Yes	28A/16D, ADM	0_0010_1100	16	See Reqd_RCW.28A	A[11:0], AD[15:0]

Required RCW Configuration for User Selections

The required RCW settings supporting the above tables are found in the following table.

Table 4-10. Required RCW Configuration based on User's Selection

Class	Required RCW Settings	Comments
Reqd_RCW.25A	RCW[IFC_A_8_6] = 1'b0	Required RCW settings when configured for a 25-bit address
Reqd_RCW.28A	RCW[IFC_A_8_6] = 1'b0 RCW[IFC_GRP_A_BASE] = 2'b00 RCW[IFC_GRP_A_EXT] = 3'b000	Required RCW settings when configured for a 28-bit address

4.8.7.3 RCW, PBI, and Boot Options

The table below summarizes the options supported for specifying the interfaces used for RCW loading, optional Preboot-Initialization (PBI), and Boot operations.

Table 4-11. Supported Interface Options for RCW, PBI, and Boot

Reset RCW Phase		Reset PBI Phase		Boot Phase after Reset
Source of the RCW Data	**cfg_rcw_src[]**	**Source of the PBI Data**	**Source of the Boot Image**	**Source of the Boot Code Image from which the Cores fetch instructions**
Low Speed Serial: • SD/MMC Card	9'b_0_0100_0000	If RCW's PBI_LENGTH field is zero, PBI is not performed. Otherwise, the PBI source is specified by cfg_rcw_src .	When PBI is performed, the source of the boot image for all records except the "Block Copy Command" is specified by cfg_rcw_src. Alternatively, a boot image can be loaded from the source specified by the Block Copy Command [SRC] field .	Instructions are fetched from a valid "execute in place" memory interface from the following address: • If BOOTLOCPTR != null, then this register provides the address • Else the address is calculated using the value in RCW[BOOT_LOC]. If [BOOT_LOC] specifies IFC (5'h18) or Serial NOR interface (5'h1A): • If RCW was also from the same interface, its configuration is specified by cfg_rcw_src. • Else the IFC configuration is specified by RCW[FLASH_MODE].
Low Speed Serial: • eMMC	9'b_0_0100_0001			
Low Speed Serial: • I2C1	9'b_0_0100_1001			
Low Speed Serial: • Serial NOR QSPI	9'b_0_0110_0xxx			
IFC Flash : • NOR flash	9'b0_00xx_xxxx			

4.8.7.4 Power-on reset configuration

Various device functions are initialized by sampling certain signals during power on reset.

The values of all these signals are sampled into registers when PORESET_B is deasserted. These inputs are to be pulled high or low by external resistors. During PORESET_B, all other signal drivers connected to these signals must be in the high-impedance state.

All POR configuration signals have internal pull-up resistors so that if the desired setting is high, there is no need for a pull-up resistor on the board. See the chip data sheet for proper pull-down resistor values for POR configuration signals, when necessary.

This section describes the functions and modes configured by the POR configuration signals.

NOTE

In the following tables, the binary value 0 represents a signal pulled down to GND and a value of 1 represents a signal pulled up to that signal's corresponding V_{DD} voltage level, regardless of the sense of the functional signal name.

4.8.7.5 General-purpose input

The general-purpose inputs listed in this table are available for application-specific use.

The encoded values latched on these signals during POR are accessible in GPPORCR, as described in General-Purpose POR Configuration Register (GPPORCR1).

Table 4-12. General-purpose input

Functional signals	Reset configuration name	Value (binary)	General-purpose input
IFC_AD[7:0] Default (1111_1111)	cfg_gpinput[7:0]	all	Application-defined

4.8.7.6 IFC external transceiver enable polarity select

The IFC external transceiver enable polarity select input, described in this table, specifies the polarity of the IFC_TE output signal to accommodate external transceivers with either active-high or active-low enable inputs.

The encoded value latched on this signal during POR is accessible in PORSR1[IFC_TE]. See External transceiver enable (TE) for more information.

Table 4-13. IFC external transceiver enable polarity select

Functional signals	Reset configuration name	Value (binary)	IFC external transmitter polarity
IFC_TE Default (1)	cfg_ifc_te	0	IFC drives logic 1 for TE assertion
		1	IFC drives logic 0 for TE assertion

4.8.7.7 RCW Loading Phase - Service Processor Brings in RCW Data

The Service Processor executes program code that directly accesses the CCSR space of the IFC, I2C, SPI, QSPI, or eSDHC blocks, and/or to the IFC memory space directly. If a non-hard-coded RCW option is selected, then the Service Processor reads in RCW data from one of these interfaces and then pushes the data to the Device Configuration (DCFG) block's RCWSR registers to configure the device.

The Service Processor has direct access to the chip's memory space through the chip interconnect for reading PBI records. The Service Processor also has bypass access to the CCSR and DCSR spaces.

In the PBI phase, the Service Processor's read transactions are steered to either CCSR space or the IFC, while write transactions are steered to the RCWSR registers that receive RCW information in the Device Configuration block's CCSR space. .

4.8.7.8 Reset Control Word (RCW) Field Descriptions

4.8.7.8.1 RCW

Function

Fields

Bit(s)	Field name	Description	Notes/comments
1-0	SYS_PLL_CFG	System PLL Configuration.	Generic field to allow options for analog/digital control of PLL. 0b00 - Setting A 0b01 - Reserved 0b10 - Reserved 0b11 - Reserved
6-2	SYS_PLL_RAT	System PLL Multiplier/Ratio	Configures the ratio between the System Clock and the SoC's coherency domain. which specifies how the Platform clock relates to the SoC's Coherency Domain clock. **NOTE:** Odd ratios are not supported when the Coherency domain runs at double the rate of the Platform clock. 0b00000 - Reserved 0b00001 - Reserved 0b00010 - Reserved 0b00011 - Reserved 0b00100 - 4:1 0b00101 - 5:1 0b00110 - 6:1 0b00111 - 7:1

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b01000 - 8:1 0b01001 - 9:1 0b01010 - 10:1 0b01011 - 11:1 0b01100 - 12:1 0b01101 - Reserved 0b01110 - Reserved 0b01111 - Reserved 0b10000 - Reserved 0b10001 - Reserved 0b10010 - Reserved 0b10011 - Reserved 0b10100 - Reserved 0b10101 - Reserved 0b10110 - Reserved 0b10111 - Reserved 0b11000 - Reserved 0b11001 - Reserved 0b11010 - Reserved 0b11011 - Reserved 0b11100 - Reserved 0b11101 - Reserved 0b11110 - Reserved 0b11111 - Reserved
7	RESERVED_7	Reserved	
9-8	MEM_PLL_CFG	Memory Controller Complex PLL Configuration (applies to all Class 1 DDR controllers).	0b00 - Setting A 0b01 - Reserved 0b10 - Reserved 0b11 - Reserved
15-10	MEM_PLL_RAT	Memory Controller Complex PLL Multiplier/Ratio (applies to all Class 1 DDR controllers).	Configures the ratio between the DDR Reference Clock and the DDR data rate. See RCW[DDR_REFCLK_SEL] which selects the DDR Reference clock. **NOTE:** Only async mode of operation is supported for the memory controller complex. 0b000000 - Reserved 0b000001 - Reserved 0b000010 - Reserved 0b000011 - Reserved 0b000100 - Reserved 0b000101 - Reserved 0b000110 - Reserved 0b000111 - Reserved 0b001000 - Reserved 0b001001 - Reserved 0b001010 - 10:1 (Async Mode) 0b001011 - 11:1 (Async Mode) 0b001100 - 12:1 (Async Mode) 0b001101 - 13:1 (Async Mode) 0b001110 - 14:1 (Async Mode) 0b001111 - 15:1 (Async Mode) 0b010000 - 16:1 (Async Mode) 0b010001 - 17:1 (Async Mode) 0b010010 - 18:1 (Async Mode) 0b010011 - 19:1 (Async Mode)

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b010100 - 20:1 (Async Mode) 0b010101 - 21:1 (Async Mode) 0b010110 - Reserved 0b010111 - Reserved 0b011000 - Reserved 0b011001-0b111111 - Reserved
17-16	—		
23-18	—		
25-24	CGA_PLL1_CFG	Cluster Group A PLL 1 Configuration	0b00 - Setting A 0b01 - Reserved 0b10 - Reserved 0b11 - Reserved
31-26	CGA_PLL1_RAT	Cluster Group A PLL 1 Multiplier/ Ratio	Configures the ratio between the System Clock and the CGAn PLL output. 0b000000 - Reserved 0b000001 - Reserved 0b000010 - Reserved 0b000011 - Reserved 0b000100 - Reserved 0b000101 - Reserved 0b000110 - 6:1 (Async Mode) 0b000111 - 7:1 (Async Mode) 0b001000 - 8:1 (Async Mode) 0b001001 - 9:1 (Async Mode) 0b001010 - 10:1 (Async Mode) 0b001011 - 11:1 (Async Mode) 0b001100 - 12:1 (Async Mode) 0b001101 - 13:1 (Async Mode) 0b001110 - 14:1 (Async Mode) 0b001111 - 15:1 (Async Mode) 0b010000 - 16:1 (Async Mode) 0b010001 - Reserved 0b010010 - Reserved 0b010011 - Reserved 0b010100 - Reserved 0b010101 - Reserved 0b010110 - Reserved 0b010111 - Reserved 0b011000 - Reserved 0b011001 - Reserved 0b011010 - Reserved 0b011011 - Reserved 0b011100 - Reserved 0b011101 - Reserved 0b011110 - Reserved 0b011111 - Reserved 0b100000 - Reserved 0b100001-0b111111 - Reserved
33-32	CGA_PLL2_CFG	Cluster Group A PLL 2 Configuration	See CGA_PLL1_CFG
39-34	CGA_PLL2_RAT	Cluster Group A PLL 2 Multiplier/ Ratio	See CGA_PLL1_RAT
41-40	—		

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
47-42	—		
49-48	—		
55-50	—		
57-56	—		
63-58	—		
65-64	—		
71-66	—		
73-72	—		
79-74	—		
81-80	—		
87-82	—		
89-88	—		
95-90	—		
97-96	—		
103-98	—		
105-104	—		
111-106	—		
113-112	—		
119-114	—		
121-120	RESERVED_121_120	Reserved	
127-122	RESERVED_127_122	Reserved	
128	SYS_PLL_SPD	System PLL Speed Select	This field must always be set to zero. 0b0 - High speed operation 0b1 - Low speed operation
129	MEM_PLL_SPD	Memory Controller Complex PLL Speed Select	This field must always be set to zero.
130	—		
131	RESERVED_131	Reserved	
132	CGA_PLL1_SPD	Cluster Group A PLL 1 Speed Select	See SYS_PLL_SPD
133	CGA_PLL2_SPD	Cluster Group A PLL 2 Speed Select	See SYS_PLL_SPD
134	—		
135	—		
136	—		
137	—		
138	—		
139	—		
140	—		
141	—		

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
142	—		
143	—		
147-144	C1_PLL_SEL	Cluster 1 PLL Select	Selects a clock from one of up to two or three available Cluster Group 1 PLLs 0b0000 - CGA_PLL1 / 1 0b0001 - CGA_PLL1 / 2 0b0010 - CGA_PLL1 / 4 0b0011 - Reserved 0b0100 - CGA_PLL2 / 1 0b0101 - CGA_PLL2 / 2 0b0110 - CGA_PLL2 / 4 0b0111 - Reserved 0b1000 -Reserved 0b1001 - Reserved 0b1010 - Reserved 0b1011 - Reserved 0b1100 - Reserved 0b1101 - Reserved 0b1110 - Reserved 0b1111 - Reserved **NOTE:** When CGA_PLLn_SPD=1, the associated CGA_PLLn/N clock frequency becomes 2*CGA_PLLn/N, for N=2,4."
151-148	C2_PLL_SEL	Cluster 2 PLL Select	Selects a clock from one of up to two or three available Cluster Group A PLLs 0b0000 - CGA_PLL1 / 1 0b0001 - CGA_PLL1 / 2 0b0010 - CGA_PLL1 / 4 0b0011 - Reserved 0b0100 - CGA_PLL2 / 1 0b0101 - CGA_PLL2 / 2 0b0110 - CGA_PLL2 / 4 0b0111 - Reserved 0b1000 -Reserved 0b1001 - Reserved 0b1010 - Reserved 0b1011 - Reserved 0b1100 - Reserved 0b1101 - Reserved 0b1110 - Reserved 0b1111 - Reserved **NOTE:** When CGA_PLLn_SPD=1, the associated CGA_PLLn/N clock frequency becomes 2*CGA_PLLn/N, for N=2,4."
155-152	—		
159-156	—		
163-160	—		
167-164	—		
171-168	—		
175-172	—		

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
178-176	HWA_CGA_M1_CLK_SEL	Hardware Accelerator Block Cluster Group A Mux 1 Clock Select	Allows for a platform accelerator block's (or multiple blocks') frequency to be maximized (using async clock) by leveraging Cluster Group A PLL 1 or 2 as the source clock. This bit selects the QuadSPI input clock. 0b000 - Reserved 0b001 - Async mode -- Cluster Group A PLL 1 /1 is clock 0b010 - Async mode -- Cluster Group A PLL 1 /2 is clock 0b011 - Async mode -- Cluster Group A PLL 1 /3 is clock 0b100 - Async mode -- Cluster Group A PLL 1 /4 is clock 0b101 - Reserved 0b110 - Async mode -- Cluster Group A PLL 2 /2 is clock 0b111 - Async mode -- Cluster Group A PLL 2 /3 is clock **NOTE:** When CGA_PLLn_SPD=1, the associated CGA_PLLn/N clock frequency becomes 2*CGA_PLLn/N, for N=2,4."
181-179	HWA_CGA_M2_CLK_SEL	Hardware Accelerator Block Cluster Group A Mux 2 Clock Select	Allows for a platform accelerator block's (or multiple blocks') frequency to be maximized (using async clock) by leveraging Cluster Group A PLL 1 or 2 as the source clock. This bit selects the eSDHC input clock. 0b000 - Reserved 0b001 - Async mode -- Cluster Group A PLL 2 /1 is clock 0b010 - Async mode -- Cluster Group A PLL 2 /2 is clock 0b011 - Async mode -- Cluster Group A PLL 2 /3 is clock 0b100 - Async mode -- Cluster Group A PLL 2 /4 is clock 0b101 - Reserved 0b110 - Async mode -- Cluster Group A PLL 1 /2 is clock 0b111 - Async mode -- Cluster Group A PLL 1 /3 is clock **NOTE:** When CGA_PLLn_SPD=1, the associated CGA_PLLn/N clock frequency becomes 2*CGA_PLLn/N, for N=2,4."
184-182	—		
187-185	—		
190-188	—		
193-191	—		
196-194	—		

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
199-197	RESERVED_199_197	Reserved	
202-200	RESERVED_202_200	Reserved	
205-203	—		
208-206	RESERVED_208_206	Reserved	
209	RESERVED_209	Reserved	
210	RESERVED_210	Reserved	
212-211	DDR_REFCLK_SEL	DDR Reference Clock Selection	Selects the Reference Clock for the DDR PLL 0b00 - The DDRCLK pin provides the reference clock to the DDR PLL. 0b01 - Reserved 0b10 - System Clock (Power-on-reset cfg_eng_use[0] (RCLK_CFG on pin IFC_WE0_B) selects between the differential system clock (DIFF_SYSCLK/DIFF_SYSCLK_B) and the single-ended system clock (SYSCLK). 0b11 - Reserved
214-213	DRAM_LAT	DDR Latency	**NOTE:** if DDR latency is not known at the time the RCW is loaded, it is acceptable to conservatively configure this field as 2'b01. This field is used for optimizing the DDRinterface. No functional issues result if this field does not match the latency of the actual DRAM's used. 0b00 - (6-6-6 or 7-7-7 DRAMs) 0b01 - (8-8-8, 9-9-9, 10-10-10, 11-11-11 or higher latency DRAMs) 0b10 - reserved 0b11 - (5-5-5 DRAMs)
215	DDR_RATE	DDR Data Rate	This setting is defined by the factory. 0b0 - Required setting for this SoC 0b1 - Reserved
216	RESERVED_216	Reserved	
217	RESERVED_217	Reserved	
218	RESERVED_218	Reserved	
220-219	RESERVED_220_219	Reserved	
222-221	RESERVED_222_221	Reserved	
224-223	RESERVED_224_223	Reserved	
226-225	RESERVED_226_225	Reserved	
228-227	RESERVED_228_227	Reserved	
230-229	REQD_CUST_CONFIG_230_229	Required Customer Configuration	

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
255-231	RESERVED_255_231	Reserved	
259-256	RESERVED_259_256	Reserved	
264-260	BOOT_LOC	Boot Location	This field indicates the default boot interface from which the core fetches instructions. **NOTE:** This field is only used when the DCFG boot location pointer registers are null. Using the boot location pointer register is the preferred method when there is a pre-boot initialization image. In simpler cases with no PBI, RCW[BOOT_LOC] is the means for locating the boot image. The following restrictions apply: - Only one IFC mode can be used for CFG_RCW_SRC and BOOT_LOC - Similarly, only one Serial NOR mode can be used for CFG_RCW_SRC and BOOT_LOC See RCW, PBI, and Boot Options for a summary of allowed options. If the IFC encoding is selected RCW[FLASH_MODE] configures the IFC, provided CFG_RCW_SRC[] has not configured IFC already. If the Serial NOR encoding is selected: RCW[FLASH_MODE] configures the Serial NOR, provided CFG_RCW_SRC[] has not configured it already. 0b00000 - PCI-Express 1 0b00001 - PCI-Express 2 0b00010 - PCI-Express 3 0b00011 - Reserved 0b00100 - Reserved 0b00101 - Reserved 0b00110 - Reserved 0b00111 - Reserved 0b01000 - Reserved 0b01001 - Reserved 0b01010 - Reserved 0b01011 - Reserved 0b01100 - Reserved 0b01101 - Reserved 0b01110 - Reserved 0b01111 - Reserved 0b10000 - Reserved 0b10001 - Reserved 0b10010 - Reserved 0b10011 - Reserved 0b10100 - Memory Complex 0b10101 - OCRAM 0b10110 - Reserved 0b10111 - Reserved

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b11000 - IFC - See description for more information 0b11001 - Reserved 0b11010 - Serial NOR (QSPI) - See description for more information 0b11011 - Reserved 0b11100 - Reserved 0b11101 - Reserved 0b11110 - Reserved 0b11111 - Reserved
265	BOOT_HO	Boot Holdoff	This field is used to specify which cores remain in boot holdoff on exiting reset. Note that the core released for boot, referred to as "core 0", can be identified using the read-only BRCORENBR[CORE_NBR] value provided by the Reset block. 0b0 - All cores except core 0 in holdoff 0b1 - All cores in holdoff
266	SB_EN	Secure Boot Enable	Secure boot is enabled if either this RCW bit is set or the 'Intent to Secure' fuse value is set. The intended use of the SB_EN bit is to emulate the effect of the ITS fuse. SB_EN itself cannot replace the use of the ITS fuse for production systems. 0b0 - Secure boot not enabled 0b1 - Secure boot enabled--boot from internal Boot ROM
275-267	FLASH_MODE	Flash Mode	This field specifies the mode for the IFC or Serial NOR which contains the boot code. It is always used in conjunction with RCW[BOOT_LOC]. Specifically, It is used for configuring the flash interface (IFC or Serial NOR) for cases where the RCW (and optionally PBI) source was something other than one of these interfaces. The field is only valid when BOOT_LOC specifies either "IFC" or "Serial NOR". Note that CFG_RCW_SRC may have precedence over configuring the IFC or Serial NOR--see the definition fhte BOOT_LOC field. For the case where this is used for Serial NOR configuration as described in RCW[BOOT_LOC], see options below. 0b000000000 - Reserved 0b000000001 - Reserved 0b000000010 - Serial NOR (QSPI) 1 channel 24-bit Addressing 0b000000011 - Reserved 0b000000100 - Reserved 0b000000101 - Reserved 0b000000110 - Reserved 0b000000111 - Reserved 0b000001000-0b111111111 - Reserved

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
287-276	PBI_LENGTH	Pre-Boot Initialization Length	Size in words of the PBI image up until the final requires STOP or CRC and STOP command. The length specified here does not include the Preamble or RCW records. If there is no PBI image, this value must be set to zero. **NOTE:** A value of zero means that there are no PBI commands to execute as described in Section 24.5.3.3, "Boot 0 Pre-Boot Loader Routine". This is the means for skipping the reset sequence's PBI phase.
288	SDBGEN	Secure Debug Enable	Secure Debug (CoreSight SPIDEN) is enabled after RCW is loaded if this RCW bit is set and the 'Intent to Secure' fuse value is cleared. See Table 7-8 for exact behavior. 0b0 - Secure debug is not enabled 0b1 - Secure debug is enabled if the ITS fuse is not burned to asserted
291-289	RESERVED_291_289	Reserved	
301-292	SYSCLK_FREQ	System Clock Frequency	This field is provided so that software can determine the System Clock Frequency of the SoC. In addition, this field is used by hardware for proper configuration of the Arm Generic Timer. The encoded value is multiplied by 166.667 KHz to determine the frequency of the System Clock . The allowable range of values of this field is SoC-specific, as it depends on the range of SYSCLK frequencies supported. 0b0110000000 - Reserved 0b0110010000 - Reserved 0b0111110100 - Reserved 0b1001011000 - 100 MHz 0b1011101110 - 125 MHz 0b1100100000 - 133.333 MHz
303-302	RESERVED_303_302	Reserved	
310-304	GPIO_LED_NUM	GPIO LED Number	Selected GPIO Output bit for indicating Secure Boot Failure. This field selects the GPIO output pin to indicate a secure boot failure when the feature is enabled by RCW[GPIO_LED_EN]. This field is ignored if RCW[GPIO_LED_EN] is 1'b0. 0b0000000 - GPIO_1[0] 0b0000001 - GPIO_1[1] 0b0000010 - GPIO_1[2] 0b0000011 - GPIO_1[3] 0b0000100 - GPIO_1[4] 0b0000101 - GPIO_1[5] 0b0000110 - GPIO_1[6] 0b0000111 - GPIO_1[7] 0b0001000 - GPIO_1[8] 0b0001001 - GPIO_1[9]

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b0001010 - GPIO_1[10] 0b0001011 - GPIO_1[11] 0b0001100 - GPIO_1[12] 0b0001101 - GPIO_1[13] 0b0001110 - GPIO_1[14] 0b0001111 - GPIO_1[15] 0b0010000 - GPIO_1[16] 0b0010001 - GPIO_1[17] 0b0010010 - GPIO_1[18] 0b0010011 - GPIO_1[19] 0b0010100 - GPIO_1[20] 0b0010101 - GPIO_1[21] 0b0010110 - GPIO_1[22] 0b0010111 - GPIO_1[23] 0b0011000 - GPIO_1[24] 0b0011001 - GPIO_1[25] 0b0011010 - GPIO_1[26] 0b0011011 - GPIO_1[27] 0b0011100 - GPIO_1[28] 0b0011101 - GPIO_1[29] 0b0011110 - GPIO_1[30] 0b0011111 - GPIO_1[31] 0b0100000 - GPIO_2[0] 0b0100001 - GPIO_2[1] 0b0100010 - GPIO_2[2] 0b0100011 - GPIO_2[3] 0b0100100 - GPIO_2[4] 0b0100101 - GPIO_2[5] 0b0100110 - GPIO_2[6] 0b0100111 - GPIO_2[7] 0b0101000 - GPIO_2[8] 0b0101001 - GPIO_2[9] 0b0101010 - GPIO_2[10] 0b0101011 - GPIO_2[11] 0b0101100 - GPIO_2[12] 0b0101101 - GPIO_2[13] 0b0101110 - GPIO_2[14] 0b0101111 - GPIO_2[15] 0b0110000 - GPIO_2[16] 0b0110001 - GPIO_2[17] 0b0110010 - GPIO_2[18] 0b0110011 - GPIO_2[19] 0b0110100 - GPIO_2[20] 0b0110101 - GPIO_2[21] 0b0110110 - GPIO_2[22] 0b0110111 - GPIO_2[23] 0b0111000 - GPIO_2[24] 0b0111001 - GPIO_2[25] 0b0111010 - GPIO_2[26] 0b0111011 - GPIO_2[27] 0b0111100 - GPIO_2[28] 0b0111101 - GPIO_2[29] 0b0111110 - GPIO_2[30] 0b0111111 - GPIO_2[31] 0b1000000 - GPIO_3[0] 0b1000001 - GPIO_3[1] 0b1000010 - GPIO_3[2]

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b1000011 - GPIO_3[3] 0b1000100 - GPIO_3[4] 0b1000101 - GPIO_3[5] 0b1000110 - GPIO_3[6] 0b1000111 - GPIO_3[7] 0b1001000 - GPIO_3[8] 0b1001001 - GPIO_3[9] 0b1001010 - GPIO_3[10] 0b1001011 - GPIO_3[11] 0b1001100 - GPIO_3[12] 0b1001101 - GPIO_3[13] 0b1001110 - GPIO_3[14] 0b1001111 - GPIO_3[15] 0b1010000 - GPIO_3[16] 0b1010001 - GPIO_3[17] 0b1010010 - GPIO_3[18] 0b1010011 - GPIO_3[19] 0b1010100 - GPIO_3[20] 0b1010101 - GPIO_3[21] 0b1010110 - GPIO_3[22] 0b1010111 - GPIO_3[23] 0b1011000 - GPIO_3[24] 0b1011001 - GPIO_3[25] 0b1011010 - GPIO_3[26] 0b1011011 - GPIO_3[27] 0b1011100 - GPIO_3[28] 0b1011101 - GPIO_3[29] 0b1011110 - GPIO_3[30] 0b1011111 - GPIO_3[31] 0b1100000 - GPIO_4[0] 0b1100001 - GPIO_4[1] 0b1100010 - GPIO_4[2] 0b1100011 - GPIO_4[3] 0b1100100 - GPIO_4[4] 0b1100101 - GPIO_4[5] 0b1100110 - GPIO_4[6] 0b1100111 - GPIO_4[7] 0b1101000 - GPIO_4[8] 0b1101001 - GPIO_4[9] 0b1101010 - GPIO_4[10] 0b1101011 - GPIO_4[11] 0b1101100 - GPIO_4[12] 0b1101101 - GPIO_4[13] 0b1101110 - GPIO_4[14] 0b1101111 - GPIO_4[15] 0b1110000 - GPIO_4[16] 0b1110001 - GPIO_4[17] 0b1110010 - GPIO_4[18] 0b1110011 - GPIO_4[19] 0b1110100 - GPIO_4[20] 0b1110101 - GPIO_4[21] 0b1110110 - GPIO_4[22] 0b1110111 - GPIO_4[23] 0b1111000 - GPIO_4[24] 0b1111001 - GPIO_4[25] 0b1111010 - GPIO_4[26] 0b1111011 - GPIO_4[27]

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b1111100 - GPIO_4[28] 0b1111101 - GPIO_4[29] 0b1111110 - GPIO_4[30] 0b1111111 - GPIO_4[31]
311	GPIO_LED_EN	GPIO LED Enable	Enable for GPIO output assertion on Secure Boot Failure. This bit enables the indication of a Secure Boot failure by driving a user selected GPIO output as asserted. The selected GPIO output is used to drive a LED. . 0b0 - No GPIO bits are driven on detection of secure boot failure 0b1 - The GPIO bit selected by RCW[GPIO_LED_NUM] is driven with a value of 1'b1 on detection of a secure boot failure; otherwise it is driven with a value of 1'b0.
319-312	RESERVED_319_312	Reserved	
351-320	RESERVED_351_320	Reserved	
354-352	UART_EXT	UART Pin Configuration ("Extension" portion)	Together with UART_BASE field, this field configures functionality of UART pins: {UART1_SOUT, UART1_SIN, UART1_RTS_B, UART1_CTS_B, UART2_SOUT, UART2_SIN, UART2_RTS_B, UART2_CTS_B} 0b000 - See UART_BASE field definition 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
357-355	IIC2_EXT	IIC2 Pin Configuration ("Extension" portion)	Together with IIC2_BASE field, this field configures functionality of IIC2_* pins. 0b000 - See IIC2_BASE field definition 0b001 - Reserved 0b010 - {IIC2_SCL, IIC2_SDA} = uQE {CLK9, CLK10} 0b011 - {IIC2_SCL, IIC2_SDA} = uQE {BRGO2, BRGO3} 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
360-358	IIC3_EXT	IIC3 Pin Configuration ("Extension" portion)	Together with IIC3_BASE field, this field configures functionality of IIC3_* pins. 0b000 - See IIC3_BASE field definition 0b001 - {IIC3_SCL, IIC3_SDA} = {USB2_DRVVBUS,USB2_PWRFAULT} 0b010 - {IIC3_SCL, IIC3_SDA} = uQE {BRGO4, BRGO1} 0b011 - {IIC3_SCL, IIC3_SDA} = uQE {BRGO4, BRGO1} 0b100 - {IIC3_SCL, IIC3_SDA} = uQE {CLK11, CLK12_CLK8} 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
363-361	IIC4_EXT	IIC4 Pin Configuration ("Extension" portion)	Together with IIC4_BASE field, this field configures functionality of IIC4_* pins. 0b000 - See IIC4_BASE field definition 0b001 - Reserved 0b010 - {IIC4_SCL, IIC4_SDA} = uQE {TDMA_RQ,TDMB_RQ} 0b011 - {IIC4_SCL, IIC4_SDA} = uQE {UC1_CDB_RXER,UC3_CDB_RXER} 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
366-364	SPI_EXT	SPI Pin Configuration ("Extension" portion)	Together with SPI_BASE_BASE and SPI_PCS_BASE fields, this field configures functionality of SPI_* pins. 0b000 - See SPI_BASE_BASE and SPI_PCS_BASE field definition 0b001 - SPI_SCK=USB_REF_CLK, Others are based on SPI_BASE_BASE and SPI_PCS_BASE. 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
369-367	SDHC_EXT	SDHC Pin Configuration ("Extension" portion)	Together with SDHC_BASE_BASE and SDHC_PCS_BASE fields, this field configures functionality of SDHC_* pins. 0b000 - See SDH_BASE_BASE and SDH_PCS_BASE field definition 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
372-370	IRQ1_EXT	IRQ1 Pin Configuration ("Extension" portion)	This field configures functionality of IRQ[1] pin 0b000 - IRQ[1]

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
375-373	IRQ2_EXT	IRQ2 Pin Configuration ("Extension" portion)	This field configures functionality of IRQ[2] pin 0b000 - IRQ[2] 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
378-376	IRQ_EXT	IRQ3-11 Pin Configuration ("Extension" portion)	Together with IRQ_BASE field, this field configures functionality of IRQ[3:11] pins. 0b000 - See IRQ_BASE field definition 0b001 - uQE pins: IRQ[3] = TDMB_TSYNC, IRQ[4] = TDMA_TXD/TDMA_RXD, IRQ[5] = TDMA_RSYNC, IRQ[6] = TDMA_TXD/ TDMA_RXD_EXC, IRQ[7] = TDMA_TSYNC, IRQ[8] = TDMB_TXD/TDMB_RXD, IRQ[9] = TDMB_RSYNC, IRQ[10]=TDMB_TXD/ TDMB_RXD_EXC, IRQ[11]=f(RCW[IRQ_BASE]) 0b010 - uQE pins: IRQ[3] = UC3_RTSB_TXEN, IRQ[4] = UC1_RXD[7], IRQ[5] = UC1_CTSB_RXDV, IRQ[6] = UC1_TXD[7], IRQ[7] = UC1_RTSB_TXEN, IRQ[8] = UC3_RXD[7], IRQ[9] = UC3_CTSB_RXDV, IRQ[10]=UC3_TXD[7], IRQ[11]=f(RCW[IRQ_BASE]) 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
383-379	RESERVED_383_379	Reserved	
385-384	UART_BASE	UART Pin Configuration ("Base" portion)	This field configures functionality of 6 UART pins { UART1_RTS_B, UART1_CTS_B, UART2_SOUT, UART2_SIN, UART2_RTS_B, UART2_CTS_B} It does not control UART1_SOUT, UART1_SIN which have no functional multiplexed modes.

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			NOTE: for SoC's which have UART_EXT field as valid: UART_EXT field must be all 0's for UART_BASE to take effect. 0b00 - 6 GPIO pins (no UART flow control). 0b01 - UART1 with flow control (UART1_RTS_B, UART1_CTS_B), 4 GPIO pins. 0b10 - UART1 with flow control, UART2 with flow control. 0b11 - UART1-4 with no flow control
387-386	IIC2_BASE	IIC2 Pin Configuration ("Base" portion)	This field configures functionality of IIC2_* pins. 0b00 - IIC2 0b01 - GPIO 0b10 - SDHC_CD, SDHC_WP 0b11 - Reserved
389-388	IIC3_BASE	IIC3 Pin Configuration ("Base" portion)	This field configures functionality of IIC3_* pins. 0b00 - IIC3 0b01 - GPIO 0b10 - EVT_B[5:6] 0b11 - SPI_PCS[4:5] (only valid if SPI_BASE_BASE is configured for SPI)
391-390	IIC4_BASE	IIC4 Pin Configuration ("Base" portion)	This field configures functionality of IIC4_* pins. 0b00 - IIC4 0b01 - GPIO 0b10 - EVT_B[7:8] 0b11 - Reserved
393-392	SPI_BASE_BASE	SPI Pin Configuration ("Base" portion)	This field configures functionality of the SPI_SOUT, SPI_SIN, SPI_SCK pins. **NOTE:** for SoC's which have SPI_EXT field as valid: SPI_EXT field must be all 0's for SPI_BASE_BASE to take effect. 0b00 - SPI_SOUT, SPI_SIN, SPI_SCK 0b01 - 3 GPIO pins 0b10 - {SDHC_CLK_SYNC_OUT, SDHC_CLK_SYNC_IN, 1 GPIO pin} 0b11 - Reserved
395-394	SPI_PCS_BASE	SPI_PCS Pin Configuration ("Base" portion)	This field configures functionality of the SPI_PCS[0:3] pins. **NOTE:** for SoC's which have SPI_EXT field as valid: SPI_EXT field must be all 0's for SPI_PCS_BASE to take effect. 0b00 - SPI_PCS[0:3] 0b01 - 4 GPIO pins 0b10 - SDHC_DAT[4:7] for 8bit MMC card support 0b11 - {SDHC_VS, SDHC_CMD_DIR, SDHC_DAT0_DIR, SDHC_DAT123_DIR}
396	SDHC_BASE	SDHC Pin Configuration ("Base" portion)	This field configures functionality of SDHC pins {SDHC_CMD, SDHC_DAT[0:3], SDHC_CLK}.

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			NOTE: for SoC's which have SDHC_EXT field as valid: SDHC_EXT field must be all 0's for SDHC_BASE to take effect. **NOTE:** the SDHC pins are driven with SDHC functionality regardless of the setting of this field or the SDHC extension field if CFG_RCW_SRC selects SD/MMC. 0b0 - {SDHC_CMD, SDHC_DAT[0:3], SDHC_CLK} 0b1 - 6 GPIO pins
397	IRQ11_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ11 0b1 - GPIO
398	IRQ10_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ10 0b1 - GPIO4_08
399	IRQ09_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ09 0b1 - GPIO4_07
400	IRQ08_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ08 0b1 - GPIO4_06
401	IRQ07_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ07 0b1 - GPIO4_05
402	IRQ06_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ06 0b1 - GPIO4_04
403	IRQ05_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ05

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b1 - GPIO3_29
404	IRQ04_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ04 0b1 - GPIO3_28
405	IRQ03_BASE	IRQ Pin Configuration	This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect. 0b0 - IRQ03 0b1 - GPIO3_27
406	EVT_9	EVT_9 Pin Configuration	This field configures functionality of the EVT[9] pin. **NOTE:** EVT[9] functionality, if selected by this field, defaults to input functionality until reconfigured by software to output functionality. 0b0 - EVT[9] 0b1 - GPIO
407	RTC	RTC Pin Configuration	This field configures functionality of RTC pin. 0b0 - RTC 0b1 - GPIO
408	ASLEEP	ASLEEP Pin Configuration	This field configures functionality of ASLEEP pin. 0b0 - ASLEEP 0b1 - GPIO (output only)
415-409	RESERVED_415_409	Reserved	
418-416	IFC_GRP_A_EXT	IFC_GRP_A_EXT Pin Configuration ("Extension" portion)	Together with IFC_GRP_A_BASE field, this field configures functionality of the IFC pins {IFC_A[25:27]} for 16b data bus pinouts which pinout 28b of addressability). 0b000 - See IFC_GRP_A_BASE field definition. 0b001 - IFC_CS_B[4:6] 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
421-419	RESERVED_421_419	Reserved	
424-422	RESERVED_424_422	Reserved	
427-425	IFC_GRP_D_EXT	IFC_GRP_D_EXT Pin Configuration ("Extension" portion)	Together with IFC_GRP_D_BASE field, this field configures functionality of "Group D" of the IFC pins {IFC_PAR[0:1], IFC_PERR_B}. 0b000 - See IFC_GRP_D_BASE field definition. 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
430-428	IFC_GRP_E_EXT	IFC_GRP_E_EXT Pin Configuration ("Extension" portion)	Together with IFC_GRP_E_BASE field, this field configures functionality of "Group E1, E2 and E3" of the IFC pins {IFC_CS_B[1:3]}.
433-431	RESERVED_433_431	Reserved	
436-434	IFC_GRP_G_EXT	IFC_GRP_G_EXT Pin Configuration ("Extension" portion)	This field configures functionality of "Group G" of the IFC pins {IFC_AD[8:15]}. 0b000 - See IFC_GRP_FGHI_BASE field definition. 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
439-437	IFC_GRP_H_EXT	IFC_GRP_H_EXT Pin Configuration ("Extension" portion)	This field configures functionality of "Group H" of the IFC pins {IFC_AD[0:7], IFC_WE0_B, IFC_TE, IFC_AVD, IFC_CLE, IFC_OE_B, IFC_WP_B[0]}. **NOTE:** extension use of these pins are output-only. 0b000 - See IFC_GRP_FGHI_BASE field definition. 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
442-440	IFC_GRP_I_EXT	IFC_GRP_I_EXT Pin Configuration ("Extension" portion)	**NOTE:** this field configures functionality of "Group I" of the IFC pins {IFC_WE2_B, IFC_WE3_B, IFC_BCTL, IFC_NDDQS, IFC_RB_B[0], IFC_CLK[0:1], IFC_NDWE_B]}. 0b000 - See IFC_GRP_FGHI_BASE field definition. 0b001 - Reserved 0b010 - Reserved 0b011 - Reserved 0b100 - Reserved 0b101 - Reserved 0b110 - Reserved 0b111 - Reserved
447-443	RESERVED_447_443	Reserved	
449-448	IFC_GRP_A_BASE	IFC_GRP_A_BASE Pin Configuration ("Base" portion)	IFC's 28-bit addressing is made of concatenation of {IFC_AD[15:0], IFC_A[11:0]}. This field configures functionality of IFC pins {IFC_A[11:9]} for 16b data bus pinouts which pinout 28b of addressability).

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			This field is ignored if RCW_SRC or PBI_SRC are configured to QSPI, as in such modes the boot source defines the functionality of these pins. 0b00 - IFC_A[11:9] 0b01 - 3 pins GPIO 0b10 - {IFC_RB_B[2:4]} for 16b data bus pinouts with 25b of addressability 0b11 - QuadSPI enabled.
451-450	RESERVED_451_450	Reserved	
452	RESERVED_452	Reserved	
453	RESERVED_453	Reserved	
455-454	IFC_GRP_D_BASE	IFC_GRP_D_BASE Pin Configuration ("Base" portion)	Together with IFC_GRP_D_EXT field, This field configures functionality of "Group D" of the IFC pins {IFC_PAR[0:1], IFC_PERR_B} This field is ignored if RCW_SRC or PBI_SRC or IFC_GRP_A_BASE are configured to QuadSPI, as any of these events configured to QuadSPI forces such pins to QuadSPI operation, rather than whateveris programmed in this field. 0b00 - {IFC_PAR[0:1], IFC_PERR_B} 0b01 - {IFC_NDDQS_B, IFC_NDRE_B, 3 pins GPIO} 0b10 - 5 pins GPIO 0b11 - Reserved
457-456	IFC_GRP_E_BASE	IFC_GRP_E_BASE Pin Configuration ("Base" portion)	Together with IFC_GRP_E_EXT field, this field configures functionality of "Group E1" of the IFC pins {IFC_RB_B[1],IFC_CS_B[1:7]} 0b00 - {IFC_RB_B[1], IFC_CS_B[1:7]} 0b01 - {IFC_RB_B[1], IFC_CS_B[1:3], 4 pins GPIO} 0b10 - 8 pins GPIO 0b11 - Reserved
459-458	IFC_GRP_FGHI_BASE	IFC_GRP_FGHI_BASE Pin Configuration ("Base" portion)	This field configures functionality of "Group F, G, H and I" of the IFC pins {IFC_AD[15:0]/IFC_A[8:0]} This field is ignored for any pins that have multiplexed QuadSPI functionality if RCW_SRC or PBI_SRC or IFC_GRP_A_BASE are configured to QuadSPI, as any of these events configured to QuadSPI forces such pins to QuadSPI operation, rather than whatever is programmed in this field. Pins IFC_CS_B[0], IFC_WE*_B, IFC_BCTL, IFC_TE, IFC_NDDQS, IFC_AVD, IFC_CLE,IFC_OE_B, IFC_WP_B[0], IFC_RB_B[0], IFC_CLK[0:n], IFC_NDWE_B all have the same functionality (either IFC or GPIO) as IFC_AD[7:0] as defined by this field. 0b00 - IFC_AD[15:0]/IFC_A[5:0] pins function as IFC. If IFC_GRP_FGHI_BASE equals 00 and IFC_A_8_6 equals 0, IFC_A[8:6] pins function as IFC.

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b01 - IFC_AD[15:0] pins function as IFC. IFC_A[8:0] pins function as GPIO. 0b10 - IFC_AD[7:0] pins function as IFC. IFC_AD[15:8]/IFC_A[8:0] pins function as GPIO. 0b11 - IFC_AD[15:0]/IFC_A[8:0] pins function as GPIO. IFC disabled.
460	IFC_A_8_6	IFC_A_8_6 Pin Configuration	This field configures functionality of the IFC pins IFC_A[8:6] for 16b data bus pinouts which pinout 22b of addressability. This field is ignored if IFC_GRP_FGHI_BASE is non-zero. 0b0 - {IFC_A[8:6]} 0b1 - {IFC_WP_B[3:1]} for 16b data bus pinouts with 22b of addressability
461	QSPI_OCT_EN	QSPI Pin Configuration	This field configures the functionality of QuadSPI (multiplexed on IFC pins) This field is only used if RCW_SRC, PBI_SRC, or IFC_GRP_A_BASE is configured to QuadSPI, as it only affects pinout when QuadSPI operation is selected. 0b0 - 4-bit QuadSPI "A" and 4-bit QuadSPI "B" interfaces supported. No high speed CK or CK2 clocks supported. 0b1 - 8-bit QuadSPI "A" interface supported, together with high speed differential CK and CK2 clocks.
479-462	RESERVED_479_462	Reserved	
482-480	—		
483	—		
484	—		
485	HOST_AGT_PEX1	Host/Agent PEX1	Configures Host/Agent mode for PEX 1 Interface if present on an SoC 0b0 - Host mode 0b1 - Agent mode
486	HOST_AGT_PEX2	Host/Agent PEX2	Configures Host/Agent mode for PEX 2 Interface if present on an SoC 0b0 - Host mode 0b1 - Agent mode
487	HOST_AGT_PEX3	Host/Agent PEX3	Configures Host/Agent mode for PEX 3 Interface if present on an SoC 0b0 - Host mode 0b1 - Agent mode
488	—		
511-489	RESERVED_511_489	Reserved	
543-512	RESERVED_543_512	Reserved	

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
575-544	RESERVED_575_544	Reserved	
607-576	RESERVED_607_576	Reserved	
639-608	RESERVED_639_608	Reserved	
671-640	RESERVED_671_640	Reserved	
703-672	RESERVED_703_672	Reserved	
735-704	RESERVED_735_704	Reserved	
767-736	RESERVED_767_736	Reserved	
799-768	GP_INFO	General Purpose Information	This data doesn't configure the SoC hardware in any way. It is available to pass information to software.
831-800	Reserved_831_800	Reserved	
834-832	EC1	EC1 Pin Configuration	Configures the functionality of the EC1 pins 0b000 - WRIOP MAC4 RGMII 0b001 - GPIO2,GPIO4
837-835	EC2	EC2 Pin Configuration	Configures the functionality of the EC2 pins 0b000 - WRIOP MAC5 RGMII 0b001 - GPIO4 0b010 - 1588 and GPIO for the remaining
838	EM1	EM1 Pin Configuration	Configures the functionality of the External Media Interface1 pins 0b0 - WRIOP MDIO1 0b1 - GPIO4
839	EM2	EM2 Pin Configuration	Configures the functionality of the External Media Interface2 pins 0b0 - WRIOP MDIO2 0b1 - GPIO2
855-840	RESERVED_855_840	Reserved	
857-856	USB	USB Pin Configuration	Configures the functionality of the USB_DRVBUS and USB_PWRFAULT pins 0b00 - USB_DRVVBUS, USB_PWRFAULT 0b01 - GPIO_4[2],GPIO_4[3]
859-858	USB1_REFCLK_SEL	USB1 Reference Clock Select	Selects reference clock for USB1 PHY PLL. 0b00 - PLL reference clock is SYSCLK or DIFF_SYSCLK/_B (selected based on Power-On-Reset Config RCLK_CFG) 0b01 - PLL reference clock is USB_REF_CLK (selected via SPI_EXT, SPI_BASE RCWs) 0b10-0b11 - Reserved
861-860	USB2_REFCLK_SEL	USB2 Reference Clock Select	Selects reference clock for USB2 PHY PLL.

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			0b00 - PLL reference clock is SYSCLK or DIFF_SYSCLK/_B (selected based on Power-On-Reset Config RCLK_CFG) 0b01 - PLL reference clock is USB_REF_CLK (selected via SPI_EXT, SPI_BASE RCWs) 0b10-0b11 - Reserved
863-862	RESERVED_863_862	Reserved	
869-864	USB1_CLK_FSEL	USB1 PHY reference clock frequency select.	Drives the USB1 PHY fsel[5:0] input. This bit should be set to 0b100111. See USB PHY documentation for details.
875-870	USB2_CLK_FSEL	USB2 PHY reference clock frequency select.	Drives the USB2 PHY fsel[5:0] input. This bit should be set to 0b100111. See USB PHY documentation for details.
895-876	RESERVED_895_876	Reserved	
896	SRDS_PLL_PD_PLL1	SERDES1 PLL 1 Power Down	SerDes PLLs are powered up in the POR Reset state titled "SERDES Released from Reset". If this field is set to 0, it informs RESET_REQ logic that it should assert RESET_REQ if a proper reference clock isn't applied or the SERDES PLL does not lock. 0b0 - PLL is not powered down. 0b1 - PLL is powered down.
897	SRDS_PLL_PD_PLL2	SERDES1 PLL 2 Power Down	See SRDS_PLL_PD_PLL1
898	SRDS_PLL_PD_PLL3	SERDES2 PLL 1 Power Down	See SRDS_PLL_PD_PLL1
899	SRDS_PLL_PD_PLL4	SERDES2 PLL 2 Power Down	See SRDS_PLL_PD_PLL1
911-900	RESERVED_911_900	Reserved	
915-912	SRDS_PRTCL_S1_LN0	SerDes Protocol Select - SerDes 1 Lane0	Proper use of this field is described in "Configuration of Networking SerDes" within the SerDes Module chapter. **NOTE:** Power down for any module can be configured by programming the DEVDISRn register. If any module is powered down, its data access as well as register access are not allowed. There are extra requirements specifically for PCI Express to address the following two usage scenarios: • A PCI Express controller is selected by a specific SRDS_PRTCL_Sn_LNn option in RCW. However the end product application has no usage of the controller and thus wants to disable it. • A PCI Express controller is not selected by a specific SRDS_PRTCL_Sn_LNn option in RCW. For both the above

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			cases, the unused or unselected PCI Express controller must be disabled by setting the corresponding bit of the DEVDISR3. Once disabled, data access and register access to that PCI Express controller must be avoided to prevent undefined behavior. It is recommended to disable unused or unselected PCI Express controllers in PBI.
919-916	SRDS_PRTCL_S1_LN1	SerDes Protocol Select - SerDes 1 Lane1	Proper use of this field is described in "Configuration of Networking SerDes" within the SerDes Module chapter.
923-920	SRDS_PRTCL_S1_LN2	SerDes Protocol Select - SerDes 1 Lane2	Proper use of this field is described in "Configuration of Networking SerDes" within the SerDes Module chapter.
927-924	SRDS_PRTCL_S1_LN3	SerDes Protocol Select - SerDes 1 Lane3	Proper use of this field is described in "Configuration of Networking SerDes" within the SerDes Module chapter.
931-928	SRDS_PRTCL_S2_LN0	SerDes Protocol Select - SerDes 2 Lane0	Proper use of this field is described in "Configuration of Non-Networking SerDes" within the SerDes Module chapter.
935-932	SRDS_PRTCL_S2_LN1	SerDes Protocol Select - SerDes 2 Lane1	Proper use of this field is described in "Configuration of Non-Networking SerDes" within the SerDes Module chapter.
939-936	SRDS_PRTCL_S2_LN2	SerDes Protocol Select - SerDes 2 Lane2	Proper use of this field is described in "Configuration of Non-Networking SerDes" within the SerDes Module chapter.
943-940	SRDS_PRTCL_S2_LN3	SerDes Protocol Select - SerDes 2 Lane3	Proper use of this field is described in "Configuration of Non-Networking SerDes" within the SerDes Module chapter.
945-944	SRDS_PLL_REF_CLK_SEL_S1	SerDes PLL Reference Clock Select - SerDes 1	This field selects the PLL reference clock frequency for this SerDes. - RCW[944]: SerDes1, PLL1 - RCW[945]: SerDes1, PLL2 1'b0: 100MHz for protocols operating at 1.25 or 2.5 or 5.0GHz 100MHz for protocols operating at 3.0 or 6.0GHz 125MHz for protocols operating at 3.125 or 6.25GHz 156.25MHz for protocols operating at 10.3125GHz 1'b1: 125MHz for protocols operating at 1.25 or 2.5 or 5.0GHz 125MHz for protocols operating at 3.0 or 6.0GHz 156.25MHz for protocols operating at 3.125 or 6.25GHz
947-946	SRDS_PLL_REF_CLK_SEL_S2	SerDes PLL Reference Clock Select - SerDes 2	See SRDS_PLL_REF_CLK_SEL_S1.

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
949-948	SRDS_DIV_PEX_S1	SerDes Frequency Divider - SerDes 1	This field configures the frequency of PCI Express protocols on this SerDes. Lanes that are supporting other protocols and frequencies are unaffected by this field. 0b00-0b01 - Can train up to a max rate of 5G 0b10 - Can train up to a max rate of 2.5G 0b11 - Reserved
951-950	SRDS_DIV_PEX_S2	SerDes Frequency Divider - SerDes 2	This field configures the frequency of PCI Express protocols on this SerDes. Lanes that are supporting other protocols and frequencies are unaffected by this field. 0b00 - Can train up to a max rate of 8G (or 5G if that is the maximum for the corresponding protocol selection) 0b01 - Can train up to a max rate of 5G 0b10 - Can train up to a max rate of 2.5G 0b11 - Reserved
953-952	SRDS_S1_REFCLK_SRC_SEL	SERDES1 PLL1,2 Reference Clock Source Select	This field selects the reference clock for SerDes1 PLL1 and PLL2. 952: SERDES1, PLL1 953: SERDES1, PLL2 • 0b00 - PLL1 reference clock is SD1_REF_CLK1_P/N PLL2 reference clock is SD1_REF_CLK2_P/N • 0b01 - PLL1 reference clock is DIFF_SYSCLK/_B PLL2 reference clock is SD1_REF_CLK2_P/N • 0b10 - Reserved • 0b11 - PLL1 reference clock is DIFF_SYSCLK/_B PLL2 reference clock is DIFF_SYSCLK/_B Note that RCW[953] may be 0b1 only if RCW[952] is 0b1.
955-954	SRDS_S2_REFCLK_SRC_SEL	SERDES2 PLL1,2 Reference Clock Source Select	This field selects the reference clock for SerDes2 PLL1 and PLL2. 954: SERDES2, PLL1 955: SERDES2, PLL2 • 0b00 - PLL1 reference clock is SD2_REF_CLK1_P/N PLL2 reference clock is SD2_REF_CLK2_P/N • 0b01 - PLL1 reference clock is DIFF_SYSCLK/_B PLL2 reference clock is SD2_REF_CLK2_P/N • 0b10 - Reserved • 0b11 - PLL1 reference clock is DIFF_SYSCLK/_B

Table continues on the next page...

Bit(s)	Field name	Description	Notes/comments
			PLL2 reference clock is DIFF_SYSCLK/_B Note that RCW[955] may be 0b1 only if RCW[954] is 0b1.
959-956	RESERVED_959_948	Reserved	
991-960	RESERVED_991_960	Reserved	
1023-992	RESERVED_1023_992	Reserved	

4.8.7.9 Hard Coded RCW Options

If any of the hardcoded RCW options are used, the PBL RCW loading process is bypassed and the device is automatically configured according to the specific RCW field encodings that are pre-assigned for the given hardcoded RCW option. The hardcoded RCW options can be useful if the board has no valid RCW present or if the customer is unable to load in the RCW from a non-volatile memory. Using a hardcoded RCW option allows the part to be initialized so that the desired RCW can be programmed or restored.

The hardcoded RCW option should be selected based on the system SYSCLK and DDRCLK frequency combination.

Table 4-14. Hard-Coded RCW Options

<table>
<tr><th>cfg_rcw_src[0:8] Encoding</th><th>Comments</th></tr>
<tr><td>0_1001_101x</td><td>Hardcoded RCW, for use with SYSCLK and DDRCLK combinations:
<table><tr><th>SYSCLK</th><th>DDRCLK</th></tr><tr><td>100-133 MHz</td><td>125-133.3 MHz</td></tr></table></td></tr>
<tr><td>0_1001_110x</td><td><table><tr><th>SYSCLK</th><th>DDRCLK</th></tr><tr><td>100-133 MHz</td><td>100 MHz</td></tr></table></td></tr>
</table>

4.8.7.9.1 RCW settings for hard coded options

Table 4-15. RCW settings for hard coded options

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
1:0	SYS_PLL_CFG	Setting A (2'h0)	Setting A (2'h0)
6:2	SYS_PLL_RAT	4:1 (5'h4)	4:1 (5'h4)
7:7	RESERVED_7	Reserved (1'h0)	Reserved (1'h0)
9:8	MEM_PLL_CFG	Setting A (2'h0)	Setting A (2'h0)
15:10	MEM_PLL_RAT	12:1 (Async Mode) (6'hc)	13:1 (Async Mode) (6'hd)
17:16	Unimplemented	Reserved	Reserved
23:18	MEM2_PLL_RAT	12:1 (Async Mode) (6'hc)	13:1 (Async Mode) (6'hd)
25:24	CGA_PLL1_CFG	Setting A (2'h0)	Setting A (2'h0)
31:26	CGA_PLL1_RAT	6:1 (Async Mode) (6'h6)	6:1 (Async Mode) (6'h6)
33:32	CGA_PLL2_CFG	Setting A (2'h0)	Setting A (2'h0)
39:34	CGA_PLL2_RAT	6:1 (Async Mode) (6'h6)	6:1 (Async Mode) (6'h6)
41:40	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
47:42	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
49:48	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
55:50	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
57:56	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
63:58	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
65:64	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
71:66	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
73:72	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
79:74	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
81:80	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
87:82	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
89:88	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
95:90	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
97:96	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
103:98	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
105:104	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
111:106	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
113:112	Unimplemented	Reserved (2'h0)	Reserved (2'h0)
119:114	Unimplemented	Reserved (6'h0)	Reserved (6'h0)
121:120	RESERVED_121_120	Reserved (2'h0)	Reserved (2'h0)

Table continues on the next page...

Table 4-15. RCW settings for hard coded options (continued)

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
127:122	RESERVED_127_122	Reserved (6'h0)	Reserved (6'h0)
128:128	SYS_PLL_SPD	High speed operation (1'h0) Low speed operation (1'h0)	High speed operation (1'h0) Low speed operation (1'h0)
129:129	MEM_PLL_SPD	High speed operation (1'h0)	High speed operation (1'h0)
130:130	Unimplemented	Reserved (1'h0)	Reserved (1'h0)
131:131	RESERVED_131	Reserved (1'h0)	Reserved (1'h0)
132:132	CGA_PLL1_SPD	Low speed operation (1'b1)	Low speed operation (1'b1)
133:133	CGA_PLL2_SPD	Low speed operation (1'b1)	Low speed operation (1'b1)
134:134	Unimplemented	Reserved	Reserved
135:135	Unimplemented	Reserved	Reserved
136:136	Unimplemented	Reserved	Reserved
137:137	Unimplemented	Reserved	Reserved
138:138	Unimplemented	Reserved	Reserved
139:139	Unimplemented	Reserved	Reserved
140:140	Unimplemented	Reserved	Reserved
141:141	Unimplemented	Reserved	Reserved
142:142	Unimplemented	Reserved	Reserved
143:143	Unimplemented	Reserved	Reserved
147:144	C1_PLL_SEL	CGA_PLL1 / 1 (4'h0)	CGA_PLL1 / 1 (4'h0)
151:148	C2_PLL_SEL	CGn_PLL1 / 1 (4'h0)	CGn_PLL1 / 1 (4'h0)
155:152	Unimplemented	Reserved	Reserved
159:156	Unimplemented	Reserved	Reserved
163:160	Unimplemented	Reserved	Reserved
167:164	Unimplemented	Reserved	Reserved
171:168	Unimplemented	Reserved	Reserved
175:172	Unimplemented	Reserved	Reserved
178:176	HWA_CGA_M1_CLK_SEL	Async mode -- Cluster Group A PLL 1 /2 is clock (3'h2)	Async mode -- Cluster Group A PLL 1 /2 is clock (3'h2)
181:179	HWA_CGA_M2_CLK_SEL	Async mode -- Cluster Group A PLL 1 /2 is clock (3'h6)	Async mode -- Cluster Group A PLL 1 /2 is clock (3'h6)
184:182	Unimplemented	Reserved	Reserved
187:185	Unimplemented	Reserved	Reserved
190:188	Unimplemented	Reserved	Reserved
193:191	Unimplemented	Reserved	Reserved
196:194	Unimplemented	Reserved	Reserved

Table continues on the next page...

Table 4-15. RCW settings for hard coded options (continued)

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
199:197	RESERVED_199_197	Reserved (3'h0)	Reserved (3'h0)
202:200	RESERVED_202_200	Reserved (3'h0)	Reserved (3'h0)
205:203	Unimplemented	Reserved	Reserved
208:206	RESERVED_208_206	Reserved (3'h0)	Reserved (3'h0)
209:209	RESERVED_209	Reserved (1'h0)	Reserved (1'h0)
210:210	RESERVED_210	Reserved (1'h0)	Reserved (1'h0)
212:211	DDR_REFCLK_SEL	The DDRCLK pin provides the reference clock to the DDR PLL. (2'h0)	The DDRCLK pin provides the reference clock to the DDR PLL. (2'h0)
214:213	DRAM_LAT	(8-8-8, 9-9-9, 10-10-10, 11-11-11 or higher latency DRAMs) (2'h1)	(8-8-8, 9-9-9, 10-10-10, 11-11-11 or higher latency DRAMs) (2'h1)
215:215	DDR_RATE	Always zero (1'h0)	Always zero (1'h0)
216:216	RESERVED_216	Reserved (1'h0)	Reserved (1'h0)
217:217	RESERVED_217	Reserved (1'h0)	Reserved (1'h0)
218:218	RESERVED_218	Reserved (1'h0)	Reserved (1'h0)
220:219	RESERVED_220_219	Reserved (2'h0)	Reserved (2'h0)
222:221	RESERVED_222_221	Reserved (2'h0)	Reserved (2'h0)
224:223	RESERVED_224_223	Reserved (2'h0)	Reserved (2'h0)
226:225	RESERVED_226_225	Reserved (2'h0)	Reserved (2'h0)
228:227	RESERVED_228_227	Reserved (2'h0)	Reserved (2'h0)
230:229	DDR_FDBK_MULT	(2'h0)	(2'h0)
255:231	RESERVED_255_231	Reserved (25'h0)	Reserved (25'h0)
259:256	RESERVED_259_256	Reserved (4'h0)	Reserved (4'h0)
264:260	BOOT_LOC	Reserved (5'h17)	Reserved (5'h17)
265:265	BOOT_HO	All cores in holdoff (1'h1)	All cores in holdoff (1'h1)
266:266	SB_EN	Secure boot not enabled (1'h0)	Secure boot not enabled (1'h0)
275:267	FLASH_MODE	Always zero (9'h0)	Always zero (9'h0)
287:276	PBI_LENGTH	Always zero (12'h0)	Always zero (12'h0)
288:288	SDBGEN	Secure debug is not enabled (1'h0)	Secure debug is not enabled (1'h0)
291:289	RESERVED_291_289	Reserved (3'h0)	Reserved (3'h0)

Table continues on the next page...

Table 4-15. RCW settings for hard coded options (continued)

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
301:292	SYSCLK_FREQ	(10'h258)	(10'h258)
303:302	RESERVED_303_302	Reserved (2'h0)	Reserved (2'h0)
310:304	GPIO_LED_NUM	Always zero (7'h0)	Always zero (7'h0)
311:311	GPIO_LED_EN	No GPIO bits are driven on detection of secure boot failure (1'h0)	No GPIO bits are driven on detection of secure boot failure (1'h0)
319:312	RESERVED_319_312	Reserved (8'h0)	Reserved (8'h0)
351:320	RESERVED_351_320	Reserved (32'h0)	Reserved (32'h0)
354:352	Unimplemented	Reserved	Reserved
357:355	IIC2_EXT	See IIC2_BASE field definition (3'h0)	See IIC2_BASE field definition (3'h0)
360:358	IIC3_EXT	See IIC3_BASE field definition (3'h0)	See IIC3_BASE field definition (3'h0)
363:361	IIC4_EXT	See IIC4_BASE field definition (3'h0)	See IIC4_BASE field definition (3'h0)
366:364	SPI_EXT	See SPI_BASE_BASE and SPI_PCS_BASE field definition (3'h0)	See SPI_BASE_BASE and SPI_PCS_BASE field definition (3'h0)
369:367	Unimplemented	Reserved	Reserved
372:370	Unimplemented	Reserved	Reserved
375:373	Unimplemented	Reserved	Reserved
378:376	IRQ_EXT	See IRQ_BASE field definition (3'h0)	See IRQ_BASE field definition (3'h0)
383:379	RESERVED_383_379	Reserved (5'h0)	Reserved (5'h0)
385:384	UART_BASE	6 GPIO pins (no UART flow control). (2'h0)	6 GPIO pins (no UART flow control). (2'h0)
387:386	IIC2_BASE	GPIO (2'h1)	GPIO (2'h1)
389:388	IIC3_BASE	GPIO (2'h1)	GPIO (2'h1)
391:390	IIC4_BASE	GPIO (2'h1)	GPIO (2'h1)
393:392	SPI_BASE_BASE	3 GPIO pins (2'h1)	3 GPIO pins (2'h1)
395:394	SPI_PCS_BASE	4 GPIO pins (2'h1)	4 GPIO pins (2'h1)
396:396	SDHC_BASE	{SDHC_CMD, SDHC_DAT[0:3], SDHC_CLK} (1'h0)	{SDHC_CMD, SDHC_DAT[0:3], SDHC_CLK} (1'h0)
397:397	IRQ03_BASE	GPIO (1'h1)	GPIO (1'h1)
398:398	IRQ04_BASE	GPIO (1'h1)	GPIO (1'h1)
399:399	IRQ05_BASE	GPIO (1'h1)	GPIO (1'h1)
400:400	IRQ06_BASE	GPIO (1'h1)	GPIO (1'h1)
401:401	IRQ07_BASE	GPIO (1'h1)	GPIO (1'h1)
402:402	IRQ08_BASE	GPIO (1'h1)	GPIO (1'h1)
403:403	IRQ09_BASE	GPIO (1'h1)	GPIO (1'h1)
404:404	IRQ10_BASE	GPIO (1'h1)	GPIO (1'h1)

Table continues on the next page...

Table 4-15. RCW settings for hard coded options (continued)

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
405:405	IRQ11_BASE	GPIO (1'h1)	GPIO (1'h1)
406:406	EVT_9	EVT[9] (1'h0)	EVT[9] (1'h0)
407:407	RTC	RTC (1'h0)	RTC (1'h0)
408:408	ASLEEP	ASLEEP (1'h0)	ASLEEP (1'h0)
415:409	RESERVED_415_409	Reserved (7'h0)	Reserved (7'h0)
418:416	IFC_GRP_A_EXT	See IFC_GRP_A_BASE field definition. (3'h0)	See IFC_GRP_A_BASE field definition. (3'h0)
421:419	Unimplemented	Reserved	Reserved
424:422	Unimplemented	Reserved	Reserved
427:425	Unimplemented	Reserved	Reserved
430:428	Unimplemented	Reserved	Reserved
433:431	Unimplemented	Reserved	Reserved
436:434	Unimplemented	Reserved	Reserved
439:437	Unimplemented	Reserved	Reserved
442:440	Unimplemented	Reserved	Reserved
447:443	RESERVED_447_443	Reserved (5'h0)	Reserved (5'h0)
449:448	IFC_GRP_A_BASE	{ IFC_A[25:27]} (2'h0)	{IFC_A[25:27]} (2'h0)
451:450	Unimplemented	Reserved	Reserved
452:452	Unimplemented	Reserved	Reserved
453:453	RESERVED_453	Reserved (1'h0)	Reserved (1'h0)
455:454	IFC_GRP_D_BASE	{IFC_PAR[0:1], IFC_PERR_B} (2'h0)	{IFC_PAR[0:1], IFC_PERR_B} (2'h0)
457:456	IFC_GRP_E_BASE	8 pins GPIO (2'h2)	8 pins GPIO (2'h2)
459:458	IFC_GRP_FGHI_BASE	IFC_AD/IFC_A[0:24] pins function as IFC (2'h0)	IFC_AD/IFC_A[0:24] pins function as IFC (2'h0)
460:460	IFC_A_8_6	{IFC_A[22:24] } (1'h0)	{IFC_A[22:24]} (1'h0)
461:461	QSPI_OCT_EN	4-bit QuadSPI "A" and 4-bit QuadSPI "B" interfaces supported. No high speed CK or CK2 clocks supported. (1'h0)	4-bit QuadSPI "A" and 4-bit QuadSPI "B" interfaces supported. No high speed CK or CK2 clocks supported. (1'h0)
479:462	RESERVED_479_462	Reserved (18'h0)	Reserved (18'h0)
482:480	Unimplemented	Reserved	Reserved
483:483	Unimplemented	Reserved	Reserved
484:484	Unimplemented	Reserved	Reserved
485:485	HOST_AGT_PEX1	Always zero (1'h0)	Always zero (1'h0)
486:486	HOST_AGT_PEX2	Always zero (1'h0)	Always zero (1'h0)
487:487	HOST_AGT_PEX3	Always zero (1'h0)	Always zero (1'h0)

Table continues on the next page...

Table 4-15. RCW settings for hard coded options (continued)

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
488:488	Unimplemented	Reserved (1'h0)	Reserved (1'h0)
511:489	RESERVED_511_489	Reserved (23'h0)	Reserved (23'h0)
543:512	RESERVED_543_512	Reserved (32'h0)	Reserved (32'h0)
575:544	RESERVED_575_544	Reserved (32'h0)	Reserved (32'h0)
607:576	RESERVED_607_576	Reserved (32'h0)	Reserved (32'h0)
639:608	RESERVED_639_608	Reserved (32'h0)	Reserved (32'h0)
671:640	RESERVED_671_640	Reserved (32'h0)	Reserved (32'h0)
703:672	RESERVED_703_672	Reserved (32'h0)	Reserved (32'h0)
735:704	RESERVED_735_704	Reserved (32'h0)	Reserved (32'h0)
767:736	RESERVED_767_736	Reserved (32'h0)	Reserved (32'h0)
799:768	GP_INFO	Always zero (32'h0)	Always zero (32'h0)
831:800	Reserved_831_800	Reserved (32'h0)	Reserved (32'h0)
834:832	EC1	Always zero (3'h0)	Always zero (3'h0)
837:835	EC2	Always zero (3'h0)	Always zero (3'h0)
838:838	EM1	Always zero (1'h0)	Always zero (1'h0)
839:839	EM2	Always zero (1'h0)	Always zero (1'h0)
855:840	RESERVED_855_840	Reserved (16'h0)	Reserved (16'h0)
857:856	USB	Always zero (2'h0)	Always zero (2'h0)
859:858	USB1_REFCLK_SEL	Always zero (2'h0)	Always zero (2'h0)
861:860	USB2_REFCLK_SEL	Always zero (2'h0)	Always zero (2'h0)
863:862	USB3_REFCLK_SEL	Always zero (2'h0)	Always zero (2'h0)
869:864	USB1_CLK_FSEL	Always zero (6'h0)	Always zero (6'h0)
875:870	USB2_CLK_FSEL	Always zero (6'h0)	Always zero (6'h0)
881:876	USB3_CLK_FSEL	Always zero (6'h0)	Always zero (6'h0)
895:882	RESERVED_895_882	Reserved (14'h0)	Reserved (14'h0)
896:896	SRDS_PLL_PD_PLL1	Always one (1'b1)	Always one (1'b1)
897:897	SRDS_PLL_PD_PLL2	Always one (1'b1)	Always one (1'b1)

Table continues on the next page...

Table 4-15. RCW settings for hard coded options (continued)

		0_1001_101x	0_1001_110x
Bits	**RCW Field**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **highest freq DDRCLK (125-133)**	**"Simple RCW Loading Bypass" for tester / debug probe usage.** • **SYSCLK freq (100-133)** • **lowest freq DDRCLK (100)**
898:898	SRDS_PLL_PD_PLL3	Always one (1'b1)	Always one (1'b1)
899:899	SRDS_PLL_PD_PLL4	Always one (1'b1)	Always one (1'b1)
911:900	RESERVED_911_900	Reserved (12'h0)	Reserved (12'h0)
915:912	SRDS_PRTCL_S1_LN0	Always zero (4'h0)	Always zero (4'h0)
919:916	SRDS_PRTCL_S1_LN1	Always zero (4'h0)	Always zero (4'h0)
923:920	SRDS_PRTCL_S1_LN2	Always zero (4'h0)	Always zero (4'h0)
927:924	SRDS_PRTCL_S1_LN3	Always zero (4'h0)	Always zero (4'h0)
931:928	SRDS_PRTCL_S2_LN0	Always zero (4'h0)	Always zero (4'h0)
935:932	SRDS_PRTCL_S2_LN1	Always zero (4'h0)	Always zero (4'h0)
939:936	SRDS_PRTCL_S2_LN2	Always zero (4'h0)	Always zero (4'h0)
943:940	SRDS_PRTCL_S2_LN3	Always zero (4'h0)	Always zero (4'h0)
945:944	SRDS_PLL_REF_CLK_SEL_S1	Always zero (2'h0)	Always zero (2'h0)
947:946	SRDS_PLL_REF_CLK_SEL_S2	Always zero (2'h0)	Always zero (2'h0)
949:948	SRDS_DIV_PEX_S1	Always zero (2'h0)	Always zero (2'h0)
951:950	SRDS_DIV_PEX_S2	Always zero (2'h0)	Always zero (2'h0)
953:952	SRDS_S1_REFCLK_SRC_SEL	Always zero (2'h0)	Always zero (2'h0)
955:954	SRDS_S2_REFCLK_SRC_SEL	Always zero (2'h0)	Always zero (2'h0)
959:956	RESERVED_959_948	Reserved (4'h0)	Reserved (4'h0)
991:960	RESERVED_991_960	Reserved (32'h0)	Reserved (32'h0)
1023:992	RESERVED_1023_992	Reserved (32'h0)	Reserved (32'h0)

4.8.8 IP Logic Clock Distribution and Configuration

The chip takes primary clocking input from the external DIFF_SYSCLK signal. The DIFF_SYSCLK input (frequency) is multiplied using multiple phase locked loops (PLL) to create a variety of frequencies which can then be passed to a variety of internal logic, including cores and peripheral IP modules.

The DDR PLL is used to provide clocking to the DDR memory controller complexes. The DDR PLL may use either the DIFF_SYSCLK or the DDRCLK input clock as a reference to create a unique DDR memory controller complex clock. In this case, the DDR complex operates asynchronously with respect to the platform clock.

Note that many of the IP modules contain logic allowing software to further modify their external interface clocks within the IP module. See the applicable IP module chapter for details.

The following figures describes internal logic clock distribution along with the means to configure the various ratios and clock sources.

RCW[C1_PLL_SEL]
RCW[C2_PLL_SEL]
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/4
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/4
Cluster A to A53 cluster 1
Cluster A to A53 cluster 2
MUX
CGA_PLL1
CGA_PLL2
1/2
1/4
1/3
RCW[HWA_CGA_M1_CLK_SEL]
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/3
CGA_PLL1/4
CGA_PLL2/2
CGA_PLL2/3
QSPI
RCW[HWA_CGA_M2_CLK_SEL]
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/3
CGA_PLL2/4
CGA_PLL1/2
CGA_PLL1/3
Platform Clock
eSDHC
Platform PLL
Platform Clock
SYSCLK
32KHz CLK GEN
CLK_32K (to RTC input clock mux, LP SECMON)
Clock Control Sync Generation
Sync:HSIO Complex(1:2)
Sync:SerDes(1:2)
Sync:GIO Initiator(1:2)
Sync:GIO Target(1:2)
Sync:GIO_DIV2(1:4)
Sync:Debug(1:1)
Sync:SYSCLK(1:1)
Sync:CFP(1:N)
Sync:DPAA(1:1)
Sync:DPAA Div 2(1:2)
USB3PHY1
USB3PHY2
SS CLK IO
DIFF_SYSCLK
DIFF_SYSCLK_B
SerDes 1
SD1_REF_CLK1_P
SD1_REF_CLK1_N
SD1_REF_CLK2_P
SD1_REF_CLK2_N
SerDes1 PLL1
SerDes1 PLL2
SerDes 1
SerDes 2
SD2_REF_CLK1_P
SD2_REF_CLK1_N
SD2_REF_CLK2_P
SD2_REF_CLK2_N
SerDes2 PLL1
SerDes2 PLL2
SerDes 2
DDR CGU

Figure 4-2. Clock subsystem block diagram

Chapter 5
Interrupt Assignments

5.1 Introduction

This chapter describes the GIC interrupt assignments for the chip. These are the on-chip interrupt sources from peripheral logic within the integrated device.

5.2 Internal Interrupt Sources

The table below shows the assignments of the internal interrupt sources from the Arm GIC500 to the Arm cores.

Interrupt Identifier	Arm interrupt source	Notes
SGIs - Inter-processor interrupts		
0	SGI 1	Software-generated interrupts These interrupts do not have a physical connection as a source
1	SGI 2	
…	SGI 3 - 15	
15	SGI 16	
PPIs - Per-processor interrupts - active low or rising edge		
16	reserved	
17	reserved	
18	reserved	
19	reserved	
20	reserved	
21	reserved	
22	Debug communications channel	(cluster-internal) COMMIRQ - Debug communications channel
23	PMU overflow	(cluster-internal) PMUIRQ - Perfmon[1, 1]
24	CTI	(cluster-internal) CTIIRQ - Cross-trigger interface[1]
25	Virtual maintenance interrupt	(cluster-internal) VCPUMNTIRQ - Virtual maintenance interface[1]

Table continues on the next page...

26	Hypervisor timer	(cluster-internal) CNTHPIRQ - EL2 Physical timer event[1]
27	Virtual timer	(cluster-internal) CNTVIRQ - Virtual timer event[1]
28	Watchdog timer	
29	Secure physical timer	(cluster-internal) CNTPSIRQ - EL1 Secure physical timer event[1]
30	Non-secure physical timer	(cluster-internal) CNTPNSIRQ - EL1 Non-secure physical timer event[1]
31	reserved	
SPIs - General legacy-style discrete pin interrupts; can map any source to any destination - active high or rising edge		
32	External Interrupt 1	
33	External Interrupt 2	
34	External Interrupt 3	
35	External Interrupt 4	
36	External Interrupt 5	
37	External Interrupt 6	
38	External Interrupt 7	
39	External Interrupt 8	
40	External Interrupt 9	
41	External Interrupt 10	
42	External Interrupt 11	
43	External Interrupt 12	
44	Coherency manager error	Includes CCI-400: nERRORIRQ signal
45	SMMU Global Secure Fault	MMU-500 interrupts. There are a potentially large number of discrete perfmon and per-context interrupts that have SoC-specific connectivity
46	SMMU Combined Secure Interrupt	
47	SMMU Global Non-secure Fault	
48	SMMU Combined Non-secure Interrupt	
49	System DRAM controller 1 error	Memory Complex 1 memory controller
50	Reserved	
51	Reserved	
52	Reserved	
53	IFC run-time	These may remain ORed as before; if so, #54 is reserved
54	IFC error	
55	Thermal Monitor Unit (alarm)	
56	Thermal Monitor Unit (critical alarm)	
57	QSPI	
58	DSPI	OR of all DSPI interrupt signals
59	EPU Performance Monitor	SoC EPU PerfMon counter interrupts (from EPISR)
60	eSDHC/(SD/MMC)	
61	Trust Island/Security Monitor (SNVS) Security	

Table continues on the next page...

62	Trust Island/Security Monitor (SNVS) General	
63	MBEE	ORed multi-bit ECC errors on SoC internal RAM
64	DUART 1	
65	DUART 2	
66	I2C 1, 2	I2C1 and I2C2 ORed
67	I2C 3, 4	I2C3 and I2C4 ORed
68	GPIO 1, 2	GPIO1 and GPIO2 ORed
69	GPIO 3, 4	GPIO3 and GPIO4 ORed
70	EPU 1	SoC Event processing unit (debug facility) interrupt 1
71	EPU 2	SoC Event processing unit (debug facility) interrupt 2
72	EPU 3	SoC Event processing unit (debug facility) interrupt 3
73	EPU 4	SoC Event processing unit (debug facility) interrupt 4
74	qDMA	For direct mode only
75	qDMA error	For direct mode only
76	FlexTimer 1	OR of all Flextimer 1 interrupt signals
77	FlexTimer 2	OR of all Flextimer 2 interrupt signals
78	FlexTimer 3	OR of all Flextimer 3 interrupt signals
79	FlexTimer 4	OR of all Flextimer 4 interrupt signals
80	MC TZASC 1	Memory Complex 1 security block
81	Reserved	
82		
83	Reserved	
84	Initiator NoC error	ex. address decode errors
85	Target NoC error	ex. address decode errors
86	HSIO NoC error	ex. address decode errors
87	Memory Complex NoC error	ex. address decode errors
88	Reserved	
89	TrustZone Watchdog Timer	"first" time-out signal, the "second" time-out signal goes to HRESET_REQ
90	GIC-500 (GPP) error	"
91	Bank 1 GPP Watchdog Timers	OR of all even-numbered GPP watchdog "second" time-out interrupts; send to an odd core
92	Bank 2 GPP Watchdog Timers	OR of all odd-numbered GPP watchdog "second" time-out interrupts; send to an even core
93	Reserved	
94	CCSR security error	Illegal access to TrustZone peripherals
95	Reserved	

Table continues on the next page...

96	Processor 1 Errors	ORed error interrupts from processor cluster 1
97	Processor 2 Errors	ORed error interrupts from processor cluster 2
98	Reserved	Reserved
99	Reserved	Reserved
100	Reserved	
101	Reserved	
102	Reserved	
103	Reserved	
104	Reserved	
105	Reserved	
106	Reserved	
107	Reserved	
108	Reserved	
109	SoC Wakeup Event 1	Power management event connection
110	Reserved	
111	Reserved	
SoC definition space		
112	USB3 1	USB1 ORED INT
113	USB3 2	USB2 ORED INT
114	Reserved	
115	Reserved	
116	Reserved	
117	DPAA NoC error	
118	ACP NoC error	
119	Reserved	
120	Reserved	
121	Reserved	
122	MDIO 1GLT	MDIO 1Gb external PHYs.
123	MDIO 10GLT	MDIO 10Gb external PHYs.
124	Reserved	
125	Reserved	
126	Reserved	
127	Reserved	
128	uQE	Quicc Engine
129	Reserved	
130	Reserved	
131	Reserved	
132	Reserved	
133	Reserved	
134	Reserved	

Table continues on the next page...

135	Reserved	
136	Reserved	
137	Reserved	
138	Reserved	
139	Reserved	
140	PEX 1	
141	PEX INTA 1	
142	PEX INTB 1	
143	PEX INTC 1	
144	PEX INTD 1	
145	PEX 2	
146	PEX INTA 2	
147	PEX INTB 2	
148	PEX INTC 2	
149	PEX INTD 2	
150	PEX 3	
151	PEX INTA 3	
152	PEX INTB 3	
153	PEX INTC 3	
154	PEX INTD 3	
155	Reserved	
156	Reserved	
157	Reserved	
158	Reserved	
159	Reserved	
160	Reserved	
161	Reserved	
162	Reserved	
163	Reserved	
164	Reserved	
165	SATA1 int 0	
166	SATA1 int 1	
167	SATA1 int 2	
168	Reserved	
169	Reserved	
170	Reserved	
171	SEC general	
172	SEC Job Queue 1	
173	SEC Job Queue 2	
174	SEC Job Queue 3	
175	SEC Job Queue 4	
176	MCC catastrophic error	MCC catastrophic error

Table continues on the next page...

177	MTR error	MTR error
178	SMMU non-secure context interrupt 0	SMMU non-secure context interrupt 0
179	SMMU non-secure context interrupt 1	SMMU non-secure context interrupt 1
180	SMMU non-secure context interrupt 2	SMMU non-secure context interrupt 2
181	SMMU non-secure context interrupt 3	SMMU non-secure context interrupt 3
182	SMMU non-secure context interrupt 4	SMMU non-secure context interrupt 4
183	SMMU non-secure context interrupt 5	SMMU non-secure context interrupt 5
184	SMMU non-secure context interrupt 6	SMMU non-secure context interrupt 6
185	SMMU non-secure context interrupt 7	SMMU non-secure context interrupt 7
186	SMMU non-secure context interrupt 8	SMMU non-secure context interrupt 8
187	SMMU non-secure context interrupt 9	SMMU non-secure context interrupt 9
188	SMMU non-secure context interrupt 10	SMMU non-secure context interrupt 10
189	SMMU non-secure context interrupt 11	SMMU non-secure context interrupt 11
190	SMMU non-secure context interrupt 12	SMMU non-secure context interrupt 12
191	SMMU non-secure context interrupt 13	SMMU non-secure context interrupt 13
192	SMMU non-secure context interrupt 14	SMMU non-secure context interrupt 14
193	SMMU non-secure context interrupt 15	SMMU non-secure context interrupt 15
194	SMMU non-secure context interrupt 16	SMMU non-secure context interrupt 16
195	SMMU non-secure context interrupt 17	SMMU non-secure context interrupt 17
196	SMMU non-secure context interrupt 18	SMMU non-secure context interrupt 18
197	SMMU non-secure context interrupt 19	SMMU non-secure context interrupt 19
198	SMMU non-secure context interrupt 20	SMMU non-secure context interrupt 20
199	SMMU non-secure context interrupt 21	SMMU non-secure context interrupt 21
200	SMMU non-secure context interrupt 22	SMMU non-secure context interrupt 22
201	SMMU non-secure context interrupt 23	SMMU non-secure context interrupt 23
202	SMMU non-secure context interrupt 24	SMMU non-secure context interrupt 24
203	SMMU non-secure context interrupt 25	SMMU non-secure context interrupt 25
204	SMMU non-secure context interrupt 26	SMMU non-secure context interrupt 26
205	SMMU non-secure context interrupt 27	SMMU non-secure context interrupt 27
206	SMMU non-secure context interrupt 28	SMMU non-secure context interrupt 28
207	SMMU non-secure context interrupt 29	SMMU non-secure context interrupt 29
208	SMMU non-secure context interrupt 30	SMMU non-secure context interrupt 30
209	SMMU non-secure context interrupt 31	SMMU non-secure context interrupt 31
210	SMMU non-secure context interrupt 32	SMMU non-secure context interrupt 32
211	SMMU non-secure context interrupt 33	SMMU non-secure context interrupt 33
212	SMMU non-secure context interrupt 34	SMMU non-secure context interrupt 34
213	SMMU non-secure context interrupt 35	SMMU non-secure context interrupt 35
214	SMMU non-secure context interrupt 36	SMMU non-secure context interrupt 36
215	SMMU non-secure context interrupt 37	SMMU non-secure context interrupt 37
216	SMMU non-secure context interrupt 38	SMMU non-secure context interrupt 38
217	SMMU non-secure context interrupt 39	SMMU non-secure context interrupt 39
218	SMMU non-secure context interrupt 40	SMMU non-secure context interrupt 40

Table continues on the next page...

219	SMMU non-secure context interrupt 41	SMMU non-secure context interrupt 41
220	SMMU non-secure context interrupt 42	SMMU non-secure context interrupt 42
221	SMMU non-secure context interrupt 43	SMMU non-secure context interrupt 43
222	SMMU non-secure context interrupt 44	SMMU non-secure context interrupt 44
223	SMMU non-secure context interrupt 45	SMMU non-secure context interrupt 45
224	SMMU non-secure context interrupt 46	SMMU non-secure context interrupt 46
225	SMMU non-secure context interrupt 47	SMMU non-secure context interrupt 47
226	SMMU non-secure context interrupt 48	SMMU non-secure context interrupt 48
227	SMMU non-secure context interrupt 49	SMMU non-secure context interrupt 49
228	SMMU non-secure context interrupt 50	SMMU non-secure context interrupt 50
229	SMMU non-secure context interrupt 51	SMMU non-secure context interrupt 51
230	SMMU non-secure context interrupt 52	SMMU non-secure context interrupt 52
231	SMMU non-secure context interrupt 53	SMMU non-secure context interrupt 53
232	SMMU non-secure context interrupt 54	SMMU non-secure context interrupt 54
233	SMMU non-secure context interrupt 55	SMMU non-secure context interrupt 55
234	SMMU non-secure context interrupt 56	SMMU non-secure context interrupt 56
235	SMMU non-secure context interrupt 57	SMMU non-secure context interrupt 57
236	SMMU non-secure context interrupt 58	SMMU non-secure context interrupt 58
237	SMMU non-secure context interrupt 59	SMMU non-secure context interrupt 59
238	SMMU non-secure context interrupt 60	SMMU non-secure context interrupt 60
239	SMMU non-secure context interrupt 61	SMMU non-secure context interrupt 61
240	SMMU non-secure context interrupt 62	SMMU non-secure context interrupt 62
241	SMMU non-secure context interrupt 63	SMMU non-secure context interrupt 63
242		
243	SMMU Performance counter interrupt 0	SMMU Performance counter interrupt 0
244	SMMU Performance counter interrupt 1	SMMU Performance counter interrupt 1
245	SMMU Performance counter interrupt 2	SMMU Performance counter interrupt 2
246	SMMU Performance counter interrupt 3	SMMU Performance counter interrupt 3
247	SMMU Performance counter interrupt 4	SMMU Performance counter interrupt 4
248	SMMU Performance counter interrupt 5	SMMU Performance counter interrupt 5
249	Reserved	
250	Reserved	
251	Reserved	
252	Reserved	
253	Reserved	
254	Reserved	
255	Reserved	
256	Reserved	
…		
1019		
Reserved space - unused		
1020	GIC internal use	

Table continues on the next page...

1021	GIC internal use	
1022	GIC internal use	
1023	GIC internal use	
1024-8191	unused	
LPIs - Locality-specific Peripheral Interrupts - Message-based interrupts; defined by software model		
8192		
…		
65535		

1. These are all driven from the Arm Generic Timer.

5.3 ISC register descriptions

This section identifies the Interrupts Sampling management resources that are not included as part of a processor core or platform IP.

5.3.1 ISC memory map

ISPPT base address: 1F7_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	External Interrupt Pins Status Register (IRQPSR)	32	RO	0000_0000h
14h	External Interrupt Control Register (IRQCR)	32	RW	0000_0000h
20h	Thread WatchDog Reset Timer Status Register (TWDTSR0)	32	RO	0000_0000h
40h	Thread WatchDog Reset Timer Control Register (TWDTCR)	32	RW	0000_0000h
60h	SoC WatchDog Reset Timer Status Register (SWDTSR)	32	RO	0000_0000h

5.3.2 External Interrupt Pins Status Register (IRQPSR)

5.3.2.1 Offset

Register	Offset
IRQPSR	10h

5.3.2.2 Function

IRQPSR provides status for the external interrupt pin.

5.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IRQSTAT31	IRQSTAT30	IRQSTAT29	IRQSTAT28	IRQSTAT27	IRQSTAT26	IRQSTAT25	IRQSTAT24	IRQSTAT23	IRQSTAT22	IRQSTAT21	IRQSTAT20	IRQSTAT19	IRQSTAT18	IRQSTAT17	IRQSTAT16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQSTAT15	IRQSTAT14	IRQSTAT13	IRQSTAT12	IRQSTAT11	IRQSTAT10	IRQSTAT9	IRQSTAT8	IRQSTAT7	IRQSTAT6	IRQSTAT5	IRQSTAT4	IRQSTAT3	IRQSTAT2	IRQSTAT1	IRQSTAT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3.2.4 Fields

Field	Function
31 IRQSTAT31	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
30 IRQSTAT30	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
29 IRQSTAT29	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
28 IRQSTAT28	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
27 IRQSTAT27	External Interrupt Status 0b - External Interrupt does not occur.

Table continues on the next page...

Field	Function
	1b - External Interrupt occurs.
26 IRQSTAT26	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
25 IRQSTAT25	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
24 IRQSTAT24	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
23 IRQSTAT23	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
22 IRQSTAT22	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
21 IRQSTAT21	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
20 IRQSTAT20	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
19 IRQSTAT19	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
18 IRQSTAT18	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
17 IRQSTAT17	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
16 IRQSTAT16	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
15 IRQSTAT15	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
14 IRQSTAT14	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
13 IRQSTAT13	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
12	External Interrupt Status

Table continues on the next page...

Field	Function
IRQSTAT12	0b - External Interrupt does not occur. 1b - External Interrupt occurs.
11 IRQSTAT11	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
10 IRQSTAT10	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
9 IRQSTAT9	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
8 IRQSTAT8	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
7 IRQSTAT7	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
6 IRQSTAT6	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
5 IRQSTAT5	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
4 IRQSTAT4	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
3 IRQSTAT3	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
2 IRQSTAT2	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
1 IRQSTAT1	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.
0 IRQSTAT0	External Interrupt Status 0b - External Interrupt does not occur. 1b - External Interrupt occurs.

5.3.3 External Interrupt Control Register (IRQCR)

5.3.3.1 Offset

Register	Offset
IRQCR	14h

5.3.3.2 Function

IRQPSR provides programmability for inverting the external interrupt pin before forwarding to the interrupt generation logic.

The assignments of device pins to the bits in this register is provided in the Device's Microarchitecture Definition.

5.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IRQINV31	IRQINV30	IRQINV29	IRQINV28	IRQINV27	IRQINV26	IRQINV25	IRQINV24	IRQINV23	IRQINV22	IRQINV21	IRQINV20	IRQINV19	IRQINV18	IRQINV17	IRQINV16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IRQINV15	IRQINV14	IRQINV13	IRQINV12	IRQINV11	IRQINV10	IRQINV9	IRQINV8	IRQINV7	IRQINV6	IRQINV5	IRQINV4	IRQINV3	IRQINV2	IRQINV1	IRQINV0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3.3.4 Fields

Field	Function
31 IRQINV31	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
30 IRQINV30	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.

Table continues on the next page...

Field	Function
29 IRQINV29	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
28 IRQINV28	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
27 IRQINV27	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
26 IRQINV26	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
25 IRQINV25	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
24 IRQINV24	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
23 IRQINV23	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
22 IRQINV22	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
21 IRQINV21	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
20 IRQINV20	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
19 IRQINV19	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
18 IRQINV18	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
17 IRQINV17	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
16 IRQINV16	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
15 IRQINV15	External Interrupt Inversion 0b - External Interrupt pin is not inverted.

Table continues on the next page...

Field	Function
	1b - External Interrupt pin is inverted.
14 IRQINV14	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
13 IRQINV13	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
12 IRQINV12	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
11 IRQINV11	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
10 IRQINV10	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
9 IRQINV9	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
8 IRQINV8	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
7 IRQINV7	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
6 IRQINV6	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
5 IRQINV5	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
4 IRQINV4	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
3 IRQINV3	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
2 IRQINV2	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
1 IRQINV1	External Interrupt Inversion 0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.
0	External Interrupt Inversion

Field	Function
IRQINV0	0b - External Interrupt pin is not inverted. 1b - External Interrupt pin is inverted.

5.3.4 Thread WatchDog Reset Timer Status Register (TWDTSR0)

5.3.4.1 Offset

Register	Offset
TWDTSR0	20h

5.3.4.2 Function

TWDTSRn provides status for the thread WatchDog timer.

5.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TWDTSTAT31	TWDTSTAT30	TWDTSTAT29	TWDTSTAT28	TWDTSTAT27	TWDTSTAT26	TWDTSTAT25	TWDTSTAT24	TWDTSTAT23	TWDTSTAT22	TWDTSTAT21	TWDTSTAT20	TWDTSTAT19	TWDTSTAT18	TWDTSTAT17	TWDTSTAT16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TWDTSTAT15	TWDTSTAT14	TWDTSTAT13	TWDTSTAT12	TWDTSTAT11	TWDTSTAT10	TWDTSTAT9	TWDTSTAT8	TWDTSTAT7	TWDTSTAT6	TWDTSTAT5	TWDTSTAT4	TWDTSTAT3	TWDTSTAT2	TWDTSTAT1	TWDTSTAT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3.4.4 Fields

Field	Function
31 TWDTSTAT31	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
30 TWDTSTAT30	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
29 TWDTSTAT29	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
28 TWDTSTAT28	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
27 TWDTSTAT27	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
26 TWDTSTAT26	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
25 TWDTSTAT25	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
24 TWDTSTAT24	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
23 TWDTSTAT23	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
22 TWDTSTAT22	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
21 TWDTSTAT21	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
20 TWDTSTAT20	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
19 TWDTSTAT19	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
18 TWDTSTAT18	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur.

Table continues on the next page...

Field	Function
	1b - Thread WatchDog Timer occurs.
17 TWDTSTAT17	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
16 TWDTSTAT16	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
15 TWDTSTAT15	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
14 TWDTSTAT14	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
13 TWDTSTAT13	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
12 TWDTSTAT12	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
11 TWDTSTAT11	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
10 TWDTSTAT10	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
9 TWDTSTAT9	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
8 TWDTSTAT8	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
7 TWDTSTAT7	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
6 TWDTSTAT6	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
5 TWDTSTAT5	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
4 TWDTSTAT4	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
3	Thread WatchDog Timer

Table continues on the next page...

Field	Function
TWDTSTAT3	0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
2 TWDTSTAT2	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
1 TWDTSTAT1	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.
0 TWDTSTAT0	Thread WatchDog Timer 0b - Thread WatchDog Timer does not occur. 1b - Thread WatchDog Timer occurs.

5.3.5 Thread WatchDog Reset Timer Control Register (TWDTCR)

5.3.5.1 Offset

Register	Offset
TWDTCR	40h

5.3.5.2 Function

TWDTCR provides general debug control for interrupt sampling function

5.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															INT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3.5.4 Fields

Field	Function
31-1 —	Reserved, should be cleared.
0 INT_EN	Interrupt enable 0b - Thread WDT events are not delivered to interruptcontroller. 1b - Thread WDT events are delivered to interruptcontroller.

5.3.6 SoC WatchDog Reset Timer Status Register (SWDTSR)

5.3.6.1 Offset

Register	Offset
SWDTSR	60h

5.3.6.2 Function

SWDTSRn provides status for the thread WatchDog timer.

5.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SWDTSTAT31	SWDTSTAT30	SWDTSTAT29	SWDTSTAT28	SWDTSTAT27	SWDTSTAT26	SWDTSTAT25	SWDTSTAT24	SWDTSTAT23	SWDTSTAT22	SWDTSTAT21	SWDTSTAT20	SWDTSTAT19	SWDTSTAT18	SWDTSTAT17	SWDTSTAT16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SWDTSTAT15	SWDTSTAT14	SWDTSTAT13	SWDTSTAT12	SWDTSTAT11	SWDTSTAT10	SWDTSTAT9	SWDTSTAT8	SWDTSTAT7	SWDTSTAT6	SWDTSTAT5	SWDTSTAT4	SWDTSTAT3	SWDTSTAT2	SWDTSTAT1	SWDTSTAT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.3.6.4 Fields

Field	Function
31 SWDTSTAT31	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
30 SWDTSTAT30	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
29 SWDTSTAT29	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
28 SWDTSTAT28	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
27 SWDTSTAT27	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
26 SWDTSTAT26	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
25	SoC WatchDog Timer

Table continues on the next page...

Field	Function
SWDTSTAT25	0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
24 SWDTSTAT24	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
23 SWDTSTAT23	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
22 SWDTSTAT22	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
21 SWDTSTAT21	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
20 SWDTSTAT20	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
19 SWDTSTAT19	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
18 SWDTSTAT18	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
17 SWDTSTAT17	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
16 SWDTSTAT16	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
15 SWDTSTAT15	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
14 SWDTSTAT14	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
13 SWDTSTAT13	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
12 SWDTSTAT12	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
11 SWDTSTAT11	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.

Table continues on the next page...

Field	Function
10 SWDTSTAT10	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
9 SWDTSTAT9	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
8 SWDTSTAT8	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
7 SWDTSTAT7	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
6 SWDTSTAT6	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
5 SWDTSTAT5	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
4 SWDTSTAT4	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
3 SWDTSTAT3	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
2 SWDTSTAT2	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
1 SWDTSTAT1	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.
0 SWDTSTAT0	SoC WatchDog Timer 0b - SoC WatchDog Timer does not occur. 1b - SoC WatchDog Timer occurs.

Chapter 6
Arm Modules

6.1 Introduction

The chip includes the following Arm modules:

- Arm® Cortex®-A53 core
- Arm generic interrupt controller (GIC-500)
- System memory management unit (MMU-500)
- Arm Generic Timer
- Cache Coherent Interconnect (CCI-400)
- Arm CoreLink™ TrustZone Address Space Controller TZC-400
- Arm AMBA® 3 APB TrustZone Protection Controller (TZPC)
- PrimeCell® Infrastructure AMBA 3 AXI TrustZone Memory Adapter (TZMA)
- Watchdog Timer Unit

This chapter provides a brief overview of the core, interrupt controller, memory management unit and generic timer. For more information on these modules, see the Arm documentation that accompanies this reference manual. For the latest released documentation for the possible updates, visit Arm information center.

Table 6-1. Related resources from Arm

Resource	IP Revision
Arm® Cortex®-A53 MPCore Processor	r0p4
Arm® CoreLink™ GIC-500 Generic Interrupt Controller	r1p1
Arm® CoreLink™ MMU-500 System Memory Management Unit	r2p4
Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile	v8.0

For information on other Arm modules, see the following:

- Cache Coherent Interconnect (CCI-400)
- TrustZone Address Space Controller (TZC-400)
- TrustZone Protection Controller (TZPC)
- TrustZone Memory Adapter (TZMA)

6.2 Arm® Cortex®-A53

The Arm® Cortex®-A53 processor is an extremely power efficient Armv8 processor capable of supporting 32-bit and 64-bit code seamlessly. It makes use of a highly efficient 8-stage in-order pipeline balanced with advanced fetch and data access techniques for performance. Cortex-A53 Core processor has a L1 cache subsystem and an L2 cache controller.

- NEON™ Advanced SIMD - Advanced single instruction multiple data version with NEON instructions to perform "Packed SIMD" processing
  - Registers are considered as vectors of elements of the same data type.
  - Data types can be: signed/unsigned 8-bit, 16-bit, 32-bit, 64-bit, single precision floating point.
  - Instructions perform the same operation in all lanes.
  - NEON technology is a 128-bit SIMD (Single Instruction, Multiple Data) architecture extension designed to provide flexible and powerful acceleration for consumer multimedia applications, delivering a significantly enhanced user experience. It has 32 registers, 64-bits wide (dual view as 16 registers, 128-bits wide.
- Vector floating-point version 4 (VFPv4) architecture extension for floating-point computation that is fully compliant with the IEEE 754 standard
- TrustZone security technology
- Virtualization extensions for the development of virtualized systems that enable the switching of guest operating systems
- Enables multiple software environments and their applications to simultaneously access the system capabilities
- Large physical address (LPA) reach for address translation of up to 40 bit physical addresses
- Multiprocessing extensions for multiprocessing functionality
- In-order pipeline with direct and indirect branch prediction
- Harvard level 1 (L1) memory system with a memory management unit (MMU)
- Level 2 (L2) memory system
  - 16-way set-associative 1 MB L2 cache with 64 byte line size
- CoreSight™ debug interface
- Trace support through an embedded trace macrocell (ETM) interface
- 2-way set-associative 32 KB L1 instruction cache with 64 byte line size
- 4-way set-associative 32 KB L1 data cache with 64 byte line size
- ECC on L1 and L2 caches
- Snoop Control Unit: The SCU is responsible for managing the interconnect, arbitration, communication, cache-2-cache and system memory transfers, cache

coherence and other capabilities for the processor. The Cortex-A53 processor also exposes these capabilities to other system accelerators and non-cached DMA driven peripherals to increase performance and reduce system wide power consumption. This system coherence also reduces software complexity involved maintaining software coherence within each OS driver.

6.3 Arm generic interrupt controller (GIC)

Generic interrupt controller (GIC) is a centralized resource for supporting and managing interrupts in a system that includes at least one processor. It provides:

- A rich and flexible approach to inter-processor communication and the routing and prioritization of system interrupts. Supporting up to 256 independent interrupts, under software control, each interrupt can be distributed across CPU, hardware prioritized, and routed between the operating system and TrustZone software management layer. This routing flexibility and the support for virtualization of interrupts into the operating system, provides one of the key features required to enhance the capabilities of a solution utilizing a hypervisor.
- Support for the following interrupt types:
  - System Error Interrupts (SEIs)
  - Locality-specific Peripheral Interrupts (LPI). These are interrupts generated by a peripheral writing to a memory-mapped register in the GIC-500.
- Memory-mapped access to all registers
- Interrupt masking and prioritization
- Registers for managing interrupt sources, interrupt behavior, and interrupt routing to one or more processors
- Support for the following:
  - Arm architecture security extensions
  - Arm architecture virtualization extensions
  - Enabling, disabling, and generating processor interrupts from hardware (peripheral) interrupt sources
  - Software-generated interrupts (SGIs)
  - Interrupt masking and prioritization
  - Uniprocessor and multiprocessor environments
  - Wakeup events in power-management environments

6.4 System memory management unit (MMU-500)

The MMU-500 is a system-level Memory Management Unit (MMU) that translates an input address to an output address, by performing one or more translation table walks.

It supports the translation table formats defined by the Arm architecture, Armv8, and can perform

- Stage 1 translations that translate an input Virtual Address (VA) to an output Physical Address (PA) or Intermediate Physical Address (IPA).
- Stage 2 translations that translate an input IPA to an output PA.
- Combined stage 1 and stage 2 translations that translate an input VA to an output IPA, and then translate that IPA to a PA. The MMU-500 performs a translation table walk for each stage of the translation.
- The purpose of separating the process into 2 stages is to allow the guest OS to control the address translation between VA and what it “thinks” is the PA, while the Hypervisor controls the translation from IPA to PA. This split process allows the Hypervisor to separate the resources of different Virtual Machines (VMs). Address translation is performed both in the cores and, for IO devices, using the System Memory Management Unit (SMMU).

The MMU-500 is a distributed SMMU, which makes-use of one central controller (TCU) and up to 32 translation units (TBUs). In this chip there is a single TCU supporting 6 TBUs. The SMMU supports 10 stream ID bits and Security State Determination (SSD). The security attributes and other security features depend on the SSD.

The SMMU uses a hit/miss mechanism for the following concatenation: {tbu number,stream_id}. This concatenation (and not just the stream ID) is then assigned (if exist) to a context bank that determines the translation type and form. Another parameter that affects the translation is the SSD value. Non-secure masters can only be subjected to a stage 2 translation, so this value also plays a part in the translation process.

Each master has a unique stream ID assigned to it. The stream ID is an identifier attached to each transaction to determine the translation context.

The Access Management Qualifier (AMQ) is the key to enforcing system partitioning and virtualization for all non-GPP transactions that are serviced by the chip. Each transaction issued in the chip outside of the coherency domain is assigned an AMQ by the chip hardware.

The SMMU supports cache coherency for page table walks and DVM transactions for page table cache maintenance operations. The SMMU can support various PAGESIZE options (configured at run time).

The key guideline in the SMMU structure is that any transaction from any IP to a memory location (either to another IP or to the external memory) must go through the MMU. This is true even if the device generates transactions using a physical address, as the MMU is also required for resource separation between VMs and must therefore inspect every transaction. There are exceptions to this rule; for example, the GIC and DPAA IPs can access the Packet Express Buffer without going through the MMU.

The Access Management Qualifier (AMQ) is the key to enforcing system partitioning and virtualization for all non-GPP transactions that are serviced by the chip. As previously mentioned, each transaction issued in a chip outside of the coherency domain is assigned an AMQ by the chip hardware. The mechanism used to specify the AMQ for each transaction is determined by the transaction source. This is typically accomplished using configuration registers or hardware descriptors that are managed exclusively by the hypervisor, although there are other methods such as tying the attributes to specific values in the chip. There are also special cases (specifically, in the DPAA2) where certain parts of the AMQ may be controlled by the guest OS or user.

The AMQ consists of a set of attributes:

- Privileged Level (PL) bit: The Privileged Level attribute indicates that the source of the transaction is authorized to issue a transaction with the Bypass Memory Translation attribute set. The Privileged Level bit can only be set by the hypervisor or other trusted entity. This bit typically comes from a configuration register, although it may be tied one way or the other in the chip logic.
- Bypass Memory Translation (BMT) bit: The Bypass Memory Translation attribute indicates that the transaction contains a chip physical address and is therefore requesting to be exempt from authentication and translation by the I/O MMU. It is possible for this attribute to be set by user or guest OS software in certain circumstances; therefore this attribute is qualified by the Privileged Level (PL) attribute. It is ultimately up to the hypervisor, via the I/O MMU or other means, to decide if the request is actually honored. In a well-behaved system the BMT bit will not be set unless the PL bit is set.
- ICID[14:0] field: The Isolation Context Identifier, or ICID, field is used by system software to identify a container, a collection of hardware and software resources that are permitted to interact with each other on behalf of some high-level “owner”. Because the resources within a container all interact, they must be regarded as a single inseparable object, and therefore must be managed as a whole. This field is controlled by the hypervisor. The ICID assignments must not be modifiable by user or guest OS software.

6.5 Arm Generic Timer (Global Generic Reference Timer)

The Arm Generic Timer (Global Generic Reference Timer) provides a single system counter to measure the passing of time in real-time, even if individual cores or clusters of cores may be operating at different frequencies.

A global timebase, Arm Generic Timer, is used to provide the same counter value to each core cluster. Note that the system counter cannot be clocked with a core cluster clock because each core cluster may be run at a different frequency.

The Arm Generic Timer (Global Generic Reference Timer) is implemented as follows:

- CNTCV is implemented as a 56-bit counter which is converted to a gray coded value before sending to GPP cores.
- This system counter is clocked with a ‘base frequency’ of System Clock / 4, providing frequencies from (64/4 to 133.3/4 MHz) depending on the value in RCW[SYSCLK_FREQ].
- SCTR is always incremented with a value of 1 at the ‘base frequency’ rate.
- Arm’s “frequency modes table” is not implemented.
- CNTCR[FCREQ] is reserved and must always be written with zero. The chip does not support the use of this field. Changing this value in software does not affect the Generic Timer hardware. Writes of values other than zero are unsupported.

For more information on the registers please refer section I1.2.2,Counter module control and status register summary in System Level Implementation of the Generic Timer section of the ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile.

Chapter 7
Service Processor

7.1 General Format of Pre-Boot Initialization Image

This section defines the commands interpreted by the Service Processor during the pre-boot initialization phase of reset. The Pre-Boot Initialization Image consists of a required 4-byte preamble, which identifies a valid PBI Image, followed by a sequence of commands.

Each command contains a 1-byte header which conveys the class of command as well as size information.

The minimum command size is 4 bytes. The full size of Configuration Write commands can be calculated based on the first byte, while the full size of General Commands can be determined from the first two bytes.

The Pre-Boot Initialization Image can also contain security information for authentication of the Pre-Boot Initialization Image and the first level boot loader. This information is placed in SRK tables and RSA Signature Tables.

7.1.1 PBI Command List

The Pre-Boot Initialization commands are summarized in the following table.

Table 7-1. PBI Command Summary

Command	Number	Size (bytes)	Description
Configuration Write Commands			
CCSR Write	-	8	32-byte write to a CCSR register
Alternate Configuration Write	-	variable	variable size write to offset from current value of ALTCFG_BAR
Block Copy Commands			

Table continues on the next page...

Table 7-1. PBI Command Summary (continued)

Command	Number	Size (bytes)	Description
Block Copy	0x00	16	Copy data from any of the available memory interfaces to a RAM
Special Load commands			
Load RCW with Checksum	0x10	136	Read Reset Configuration Word, perform simple 32-bit checksum, and update RCW registers
Load RCW w/o Checksum	0x11	136	Read Reset Configuration Word and update RCW registers without performing checksum
Load Alternate Configuration Window	0x12	4	Read in condition 14-bit base pointer for alternate configuration space
Load Condition	0x14	12	Read in condition information for subsequent Conditional Jump
Load Security Header	0x20	84	Read CSF Header for authentication of PBI Image
Load Boot 1 CSF Header Ptr	0x22	8	Read in a pointer to CSF header for authentication of Boot 1 code
Control Commands			
Poll Short Poll Long	0x80 0x81	16	Poll a specified address for a specified address for a specified value
Wait	0x82	4	Pause PBI sequence for specified number of iteration of a FOR loop
Jump	0x84	8	Unconditional jump forward in PBI command sequence
Jump Conditional	0x85	12	Conditional jump forward in PBI boot sequence
CRC and Stop	0x8F	8	Stop the PBI sequence and indicate the expected CRC value
Stop	0xFF	8	Stop the PBI sequence

7.1.2 Header Format for a CCSR Configuration Write

A CCSR Write Command begins with a 1-byte header in which the most significant bit is a zero (Alternate Configuration Space = 1'b0) as shown in the following table.

Table 7-2. CCSR Write Header

Bit	7	6	5	4	3	2	1	0
Header	ACS=0	RSVD=0	B		ADDR			

7.1.3 Header Format for an AltConfig Write Command

An AltConfig Write Command begins with a 1-byte header in which the most significant bit is a one (Alternate Configuration Space = 1'b1) and the byte count field is non-zero (B!=0x0) as shown in the following table.

Table 7-3. AltConfig Write Header

Bit	7	6	5	4	3	2	1	0
Header	ACS=1	RSVD=0	B != 0				ADDR	

7.1.4 Format of General Commands

General commands include a command field after the header, followed by the number of data bytes needed to complete the command. The format for a general PBI command is shown below.

Table 7-4. General Command Format

Bit	7	6	5	4	3	2	1	0
Header	ACS=1	RSVD=0	0	0	0	0	RSVD=0	RSVD=0
Command Byte	COMMAND							
Data Bytes	DATA BYTE							
	DATA BYTE							
	. . .							
	DATA BYTE							

General commands are identified by a header value of 0x80

The command number is found in the 2nd byte.

Total command size is always a multiple of 4 bytes.

7.1.5 Use of Addresses in PBI Commands

Pre-boot initialization commands are processed by the Service Processor which uses 32-bit addresses.

PBI commands that generate transactions into the system use addresses that start in this 32-bit address space. Addresses above 0x1400_0000 are passed through an address translation unit (ATU) that can map to arbitrary regions in the 40-bit system address space.

Addresses that miss all windows of the ATU are passed through directly to the low order 4GB of the 40-bit system address space. In this way the PBI transactions can directly reach CCSR space as well as regions of IFC and QuadSPI interfaces and OCRAM.

7.1.6 CRC Checking of PBI Commands

The STOP command at the end of the PBI command list can include a CRC check value. CRC checks use the following CRC-32 polynomial:

$1+X^1+X^2+X^4+X^5+X^7+X^8+X^{10}+X^{11}+X^{12}+X^{16}+X^{22}+X^{23}+X^{26}+X^{32}$

with an initial condition of 0xFFFF_FFFF and a final XOR with 0x0000_0000.

The CRC value covers all PBI commands starting after the Load RCW Command up to and including the STOP w/ CRC command. It is expected that this contiguous block of memory is checked for the CRC value before the commands are executed.

NOTE

Implementation of CRC Checking

It is possible to calculate CRC on the fly as commands are executed until the STOP w/ CRC command is found. However, if this is done then it is required that all commands be consumed in order, even if a JUMP command skips forward past some of the commands. A better solution is to use the RCW[PBI_LENGTH] field to find the last command, and if that command is a STOP w/ CRC, then calculate the CRC for the full list of commands before executing any.

It is possible for STOP commands to be placed in the command sequence before the final STOP or STOP w/ CRC that is pointed to by the RCW[PBI_LENGTH] field. This allows a PBI developer to create a buffer for additional commands to be added during development without changing the RCW value. Until development is complete the final command can be a STOP. When development is complete the final command can be changed to a STOP w/ CRC with the final CRC check value.

7.2 PBI Command Detailed Descriptions

Full descriptions of the PBI commands are below.

7.2.1 Preamble

The preamble is always the first element in a PBI Image. It is contains a standard pattern that identifies the memory location as the beginning of a valid PBI Image. The preamble is a 4-byte pattern arbitrarily defined as 0xAA55_AA55, as shown below.

Table 7-5. PBI Image Preamble Format

31			28	27			24	23			20	19			16	15			12	11			8	7			4	3			0	Word Offset
0xA				0xA				0x5				0x5				0xA				0xA				0x5				0x5				0x00

Error checking

The preamble is required and must appear as the first record in the PBI image.

If the specified pattern is not found at the expected location the Boot 0 Reset Handler code will generate an error.

7.2.2 Configuration Writes

These commands include the data and the target address. They cause writes of data embedded in the command to be performed to the specified memory address.

Write commands to either CCSR space or AltConfig space do not have a command field. These common commands are implicit in the header so that single word writes can be expressed efficiently in a 2-word command.

7.2.2.1 CCSR Write

CCSR Write commands write 1 or 4 bytes of data to Configuration, Control, and Status Registers. The 28-bit address specified is the offset from the base of the 256MB CCSR space.

CCSR access size and alignment

NOTE

CCSR and DCSR registers are defined as 32-bit words and should be accessed with 4-byte transactions. Exception include UART and I2C registers which are 1-byte registers. 2-byte writes are not defined as legal CCSR/DCSR accesses and

should not be used. In general, accesses should be aligned to their size, so word accesses must be aligned to word boundaries. There is no restriction on alignment of byte accesses.

NOTE

USB PHY registers should not be accessed during the PBI phase before the system is ready.

The CCSR Write command is identified by ACS=0. It is an 8-byte command.

The format for a CCSR Write command is shown below.

Table 7-6. CCSR Write Command Format (Word view)

31			28	27																											0	WordOffset
HEADER				SYS_ADDR																												0
CCSR_DATA																																1

More detail is shown below.

Table 7-7. CCSR Write Command Format (Byte view)

	Bit	7	6	5	4	3	2	1	0
PBI Command Word 0	Header	ACS=0	RSVD=0	B[1-0]		SYS_ADDR[27-24]			
	Address	SYS_ADDR[23-16]							
		SYS_ADDR[15-8]							
		SYS_ADDR[7-0]							
PBI Command Word 1	Data Bytes	CCSR_DATA_BYTE_3							
		CCSR_DATA_BYTE_2							
		CCSR_DATA_BYTE_1							
		CCSR_DATA_BYTE_0							

Byte count N = 2^{B-1} where B is in the range 1-3 (1, 2, or 4 bytes). B=2'b00 is reserved. In addition, since 2-byte writes are not defined as legal for CCSR accesses, B=2'b10 is not considered legal.

Unused bytes should be set to zeroes. Data in the CCSR Write Command is right shifted so that it is aligned the way it is needed in the 4-byte CPU register. For instance, an access of a single byte will always require the data to be in CCSR_DATA_BYTE0 regardless of the low order bits of SYS_ADDR. A single byte could then immediately be written with a store-byte command to the target address.

Error checking

B must not equal 2'b00 or 2'b10.

Typically B will always be 2'b11. It can be 2'b01 for accesses to I2C or DUART or some debug registers.

7.2.2.2 AltConfig Write

Alternate Configuration Write commands are used to write configuration data to a region other than CCSR space. The AltConfig_Write command writes up to 512 bytes of data to an offset from the Alternate Configuration Window base address. The 26-bit ALTCFG_OFFSET indicates an offset into a 64MB window of memory set up by the Load Alternate Configuration Window command.

Since the Service Processor can only write to a 32-bit address space, a 64MB region of the SP address space is mapped by the SP Address Translation Unit (ATU) to the full 40-bit system address space. Specifically, AltConfig Write data is written to 0x8000_0000 + ALTCFG_OFFSET in the SP 32-bit address space. The ATU then translates this address to the 40 bit system address space. The Load_AltCfg_Window command is used to create the ATU translation.

The AltConfig_Write command is identified by ACS=1 and B[3-0] != 4'b0. Its length is variable with the size identified by the B field of the header.

The format for the AltConfig_Write command is shown below.

Table 7-8. AltConfig Write Command Format (Word view)

31 26	25 0	WordOffset
HEADER	ALTCFG_OFFSET	0
DATA_WORD_1		1
. . .		...
DATA_WORD_n		n

More detail is shown below.

Table 7-9. AltConfig Write Command Format (Byte view)

	Bit	7	6	5	4	3	2	1	0
PBI Command Word 0	Header	ACS=1	RSVD=0	B[3-0]				OFFSET[25-24]	
	Address	ALTCFG_OFFSET[23-16]							
		ALTCFG_OFFSET[15-8]							
		ALTCFG_OFFSET[7-0]							

Table continues on the next page...

Table 7-9. AltConfig Write Command Format (Byte view) (continued)

	Bit	7	6	5	4	3	2	1	0
PBI Command Word 1	Data Bytes	BYTE3							
		BYTE2							
		BYTE1							
		BYTE0							
...		. . .							
PBI Command Word n (n = N/4)		BYTE N-1							
		BYTE N-2							
		BYTE N-3							
		BYTE N-4							

Byte count N = 2^{B-1} where B is in the range 1-10 (byte count 1 to 512). Other values are reserved.

For B>2 ALTCFG_OFFSET must be word aligned (ALTCFG_OFFSET[1:0] = 2'b00)

Error checking

Check for invalid values of B.

Check that a Load_AltCfg_Wndw command has preceded this command to initialize the ATU properly.

Check that resulting (ALT_CFG_WNDW || ALTCFG_OFFSET) is not in SP Stack or program space.

7.2.3 Block Copy Command

The Block Copy command is used to move larger blocks of data efficiently from non-volatile memory to on-chip RAM while keeping the size of the PBI Image small (the data itself is not embedded in the command). Specifically this command can be used to move a boot loader into SRAM for authentication and/or execution.

The target of the destination can be any randomly accessible memory. Typically this will be on-chip SRAM (OCRAM). It can be DRAM, but only if it is practical to configure the DRAM controller of a given board with PBI commands; this is often not feasible due to the complexity of configuring DRAM, especially DIMMs.

The source of the data can be randomly accessible memory (e.g. IFC NOR Flash or QuadSPI) or it can be memory that is not directly randomly accessible (e.g. IFC NAND, SD/MMC, or SPI). The command includes a descriptor of the source interface so that software can know how to access it.

The format for a Block Copy command is shown below.

Table 7-10. Block Copy Command Format

31							24	23						16	15															0	WordOffset
HDR = 0x80								CMD = 0x00							SRC																0
SRC_ADDR																															1
DEST_ADDR																															2
BLOCK_SIZE																															3

SRC field values will follow the encodings used for CFG_RCW_SRC field with the additional definition that a SRC value of 0x00 (reserved in CFG_RCW_SRC definition) can be used here for randomly accessible memories including OCRAM, other on-chip RAMs, or even DRAM if the DRAM controller is configured.

Destination Address must always map to a randomly accessible, executable memory.

Source and destination addresses are initiated in the 32-bit SP address space. As such they can map to CCSR or user-visible DCSR regions, or they can fall through to the low 4GB region of the 40 bit system address space where they can hit OCRAM, the low region of IFC or Serial NOR (QSPI), or the low region of DRAM.

Additional flexibility is possible by leveraging the ATU window that can be configured through the Load_AltCfg_wndw command. The Load_AltCfg_wndw command creates a 64MB translation window starting at 0x8000_0000. Therefore, by setting the high order bit of the SRC_ADDR or DEST_ADDR fields, either the source or the destination of the block copy can reach a 64MB region of the full 40-bit system address space, allowing access to larger regions of IFC orSerial NOR (QSPI) or DRAM space.

The block size is specified in bytes. If an ATU window is being used for the transfer, the block size must not be such that the transfer exceeds the bounds of the ATU window.

Memories must have been previously configured by POR Config pin strappings (CFG_RCW_SRC) or prior PBI commands.

This command automatically performs a Flush at the completion of the block copy. That is, the last address written is chased by a read of that same address to force it to completion before the next PBI command is executed.

Error checking

Bounds checking OCRAM limits is must be performed.

7.2.4 Load RCW

This command loads the 1024 bit Reset Configuration Word from the interface specified by Power-on Reset Configuration strapping pins (POR Config pins).

The RCW will generally always be present, though it is possible to specify through POR configuration pin strapping that a hardcoded RCW value is to be used by the hardware. In this case the service processor will not look for an RCW.

There are two formats for the Load RCW command. One includes a 32-bit Checksum at the end and requires a simple checksum check (command 0x10), and the second does not (0x11). Both commands are the same size; the version without the checksum includes padding with zeroes in the place of the checksum value. This is done so that during development, when RCW is changing often, a new checksum calculation for each RCW is not needed. But when final RCW if chosen it can be adequately protected by a checksum without changing the size of the command.

The format for a Load RCW command is shown below.

Table 7-11. Load RCW Command Format

31	24	23	16	15	0	WordOffset
HDR = 0x80		CMD = 0x10 or 0x11		0x0000		0
RCW_WORD_0						1
. . .						...
RCW_WORD_31						32
CHKSUM or Zeroes						33

CHKSUM is calculated as a 32-bit unsigned integer summation (initial value zero) of the RCW Preamble, the Load-RCW-with-Checksum command, and each of the thirty two, 32-bit words of the RCW.

Error checking

A simple 32-bit checksum is used for validation of the command. Pseudocode is shown here:

```
checksum(RCW_WORD[]) {
unsigned_32bit_int sum=0xAA55AA55 + 0x80100000 // Preamble + Load RCW Command;
for (i=0; i<32; i++) sum += RCW_WORD[i];
return (sum) }
```

7.2.5 Load Alternate Configuration Window

The Load Alternate Configuration Window (Load_AltCfg_wndw) command specifies to software a 14-bit value to be used by subsequent configuration writes as a pointer to a 64MB alternate configuration space.

The ALTCFG_WNDW value becomes the high order 14 bits of the 40-bit system address generated by the AltCfg Write command. The offset specified in the AltCfg Write command becomes the low order 26 bits of the system address.

The format for a Load Alt Cfg Window command is shown below.

Table 7-12. Load Alt Cfg Window Command Format

31							24	23							16	15	14	13													0	WordOffset
HDR = 0x80								CMD = 0x12								2'b0		ALTCFG_WNDW														0

Since the Service Processor uses a 32-bit address space, an Address Translation Unit must be used to generate the full 40-bit address. Therefore, when an Load Alternate Configuration Window command is issues, the result is the configuration of a translation window in the Address Translation Unit (ATU) block of Service Processor complex.

ALTCFG_WNDW must be a value of 5 or greater so that the corresponding 32-bit SP address is in the region that maps to the ATU to be translated into the system address space. That is, the resulting system address must be at or above 0x1400_0000. See .

An ALTCFG_WNDW value of 0 is reserved. Other ALTCFG_WNDW values less than 5 will result in errors.

A 64MB window of the SP 32-bit address space is selected as the local mapping of the Alternate Configuration Window. This window is arbitrarily set at 0x8000_0000 which is in the region of SP 32-bit address space that is mapped by the ATU. The ATU is then configured to map a 64MB window from 0x8000_0000 to the 40-bit system address space specified using ALTCFG_WNDW as the 14 high order address bits.

This translation will subsequently map the configuration writes generated by the AltCfg_Write command.

Error checking

ALTCFG_WNDW must be greater than or equal to 5 in order to map to the 40 bit system address space. However a value of 0 is reserved and should not cause an error. Values of 1 through 4 should cause an error.

7.2.6 Load Condition

The Load Condition command loads a 32-bit value from a specified randomly accessible memory location and applies a mask to create a condition value to be used by subsequent conditional control commands.

As an example, this command can be used to extract a field from the Reset Condition Register so that a Conditional Jump command can be used to jump to a set of PBI commands appropriate to the type of reset that was performed.

The format for a Load Condition command is shown below.

Table 7-13. Load Condition Command Format

31	24	23	16	15	0	WordOffset
HDR = 0x80		CMD = 0x14		0x0000		0
ADDR						1
MASK						2

The value loaded from the address indicated by the ADDR field (typically a CCSR register) is ANDed with the MASK value (bitwise AND) to select any or all fields of interest. The resulting condition will be saved in an internal variable by the PBL routine to be used in processing a subsection Conditional command.

Currently only a conditional Jump command is defined.

Conditional commands will include a CONDITION field which will be directly compared with the condition value generated by this command.

7.2.7 Load Security Header

This command loads information required for authentication of the PBI image. The Security Header includes pointers to an SRK key table and RSA signatures as well as other flags and IDs.

The format for a Load Security Header command is shown below.

Table 7-14. Load Security Header Command Format

31	24	23	16	15	0	WordOffset
HDR = 0x80		CMD = 0x20		0x0000		0

Table continues on the next page...

Table 7-14. Load Security Header Command Format (continued)

CSF_WORD_0	1
...	...
CSF_WORD_19	20

Error checking

This command is required as part of the Secure Boot process, but is not needed for non-secure boot.

Address must be 4-byte aligned.

7.2.8 Load Boot 1 CSF Header Pointer

This command loads a pointer to a CSF Header used for authentication of the Boot 1 Secondary Program Loader

This 32-bit value used by the Boot 0 ISBC and is required for secure boot.

The format for the Load Boot 1 CSF Header command is shown below.

Table 7-15. Load Boot 1 CSF Header Pointer Command Format

31							24	23							16	15															0	WordOffset
HDR = 0x80								CMD = 0x22								0x0000																0
CSF_PTR																																1

Error checking

This command is required as part of the Secure Boot process, but is not needed for non-secure boot.

CSF_PTR must be 4-byte aligned.

7.2.9 Poll

The Poll command causes repeated reads of a specified address until a specified condition is found. A mask is also specified and is applied to both the result of the read as well as the condition to select a specific field.

There are two version of the command that only differ in the number of iterations executed before failing. Command 0x80 (Poll Short) will time out after 100,000 accesses while command 0x81 (Poll Long) will time out after 10,000,000 accesses. Assuming a 400MHz clock and minimum register access of 100 cycles there would be approximately 4 million accesses per second. So the Poll Short command would time out after no less than 25 ms, and the Poll Long command would time out after no less than 2.5 seconds.

The intent of the Poll command is to cause the PBI sequence to wait until the result of a previous command is visible at a given memory location.

The Poll command degenerates to a flushing operation (which pushes the last command to completion) if the mask is all zeroes. This is true because a Poll command with a mask of all zeroes will result in a single read of the target address; the Condition is immediately met because the read value and the condition are both zeroed by the mask.

The format for a Poll command is shown below.

Table 7-16. Poll Command Format

31							24	23							16	15															0	WordOffset
HDR = 0x80								CMD = 0x80 or 0x81								0x0000																0
ADDR																																1
MASK																																2
CONDITION																																3

Error checking

CSF_PTR must be 4-byte aligned.

7.2.10 Wait

The Wait command forces the PBI sequence to wait for an approximate amount of time. Time is counted by iterations through a tight loop, where each loop iteration approximately CPU cycles running at half of the chip platform frequency.

The format for a Wait command is shown below.

Table 7-17. Wait Command Format

31							24	23							16	15															0	WordOffset
HDR = 0x80								CMD = 0x82								WAIT																0

7.2.11 Jump

The PBL routine will jump forward from the beginning of the current command by the number of bytes specified in the OFFSET field to find then next PBI command to be executed.

Since all PBI commands are an integer number of 4-byte words, OFFSET should be a multiple of 4. Since the command itself is 8 bytes, the jump OFFSET must be greater than 8.

The format for the Jump command is shown below.

Table 7-18. Jump Command Format

31							24	23							16	15															0	WordOffset
HDR = 0x80								CMD = 0x84								0x0000																0
OFFSET																																1

Error checking

Check for Jump beyond PBI Length.

Check for 4-byte alignment and OFFSET greater than 8.

7.2.12 Jump Conditional

The Jump command causes the PBI sequence to jump to a new command location based on a match of a specified condition value. This might be done to do different PBI commands on a Warm Reset versus a Cold Reset based on the load of the Reset Condition register. Or it might jump over configuration of IP that is not used in one personality versus another by jumping based on a load of the System Version Register.

If the CONDITION field matches the condition previously loaded by the last Load Condition command, then execution moves forward to the PBI command offset from the beginning of the current command by the number of bytes specified in the OFFSET field.

Since all PBI commands are an integer number of 4-byte words, OFFSET should be a multiple of 4. Since the command itself is 8 bytes, the jump OFFSET must be greater than 8.

The format for a Jump Conditional command is shown below.

Table 7-19. Conditional Jump Command Format

31							24	23							16	15															0	WordOffset
HDR = 0x80								CMD = 0x85								0x0000																0
OFFSET																																1
CONDITION																																2

Error checking

Check for Jump beyond PBI Length.

Check for 4-byte alignment and OFFSET greater than 8.

7.2.13 CRC and Stop

This command ends the PBI sequence and specifies a cumulative CRC check.

The CRC check value covers all commands from the first command after the RCW up to and including this CRC and Stop command, regardless of whether any are skipped by Jump commands during execution. That is, it is not just the cumulative CRC of commands executed up to the CRC and Stop command; rather, it is intended that the SEC engine could calculate the block across the entire image before PBI execution begins.

If there are branches in PBI command sequence then it is possible to have multiple endpoints in the PBI command sequence. In this case the CRC value will have no meaning because Boot 0 PBL code will only check the CRC based on the last command in the PBI image.

The format for a CRC and Stop command is shown below.

Table 7-20. CRC and Stop Command Format

31							24	23							16	15															0	WordOffset
HDR = 0x80								CMD = 0x8F								0x0000																0
CRC																																1

7.2.14 Stop

The Stop command ends the PBI sequence immediately. It does not include a CRC value, but it instead has a 32-bit padding with zeroes so that it is the same size as the Stop with CRC command.

The Stop command can be used as a termination point for a branch of the command sequence that is not at the end of the PBI command list, or it can be used during development of the sequence before the final CRC value is calculated.

The format for a Stop command is shown below.

Table 7-21. Stop Command Format

31							24	23						16	15															0	WordOffset
HDR = 0x80								CMD = 0xFF							0x0000																0
0x0000_0000																															1

7.3 Error Codes

All Error codes are written first in the 32 bit DCFG_SCRATCHRW3 register. If the error is in RCW or PBI Phase, then the 8 bit error code is also written in Error code field in RCW_COMPLETION or PBI_COMPLETION register respectively.

For device read related errors, the 8 bit error code is modified before being written to 32 bit SCRATCH Register.

- Bits 31-28 indicate the Phase.
- Bits 27-24 indicate the CSF_HDR Flag.

The phases are:

- PHASE_RCW (RCW Loading form offset 0)
- PHASE_RCW_DONE (PBI Loading from offset 140)
- PHASE_PBI (PBI Command Block Copy or CSF_HDR for PBI)
- PHASE_PBI_DONE (CSF_HDR for Boot1)

For example- Error code 0x24 in read during CSF_HDR for PBI will give error code in SCRATCH as 0x31000024

Table 7-22. Error Codes

When Error is Generated	Error Code	Value	Description
RCW Errors to be written to RCW_COMPLETION Register			
Reset Handler	ERROR_PREAMBLE	0x50	Preamble not found.
Reset Handler	ERROR_RCW_CMD_NOT_FOUND	0x51	RCW command not found
Reset Handler	ERROR_RCW_CHECKSUM_MISMATCH	0x52	Checksum mismatch in RCW
PBI Errors written to PBI_COMPLETION Register			

Table continues on the next page...

Table 7-22. Error Codes (continued)

When Error is Generated	Error Code	Value	Description
Reset Handler (Load PBI)	ERROR_PBI_LENGTH	0x65	PBI Length defined in RCW[PBI_LEN] field is invalid
Reset Handler (Load PBI)	ERROR_PBI_LAST_CMD_NOT_STOP	0x66	STOP or CRC&STOP not found at the end in the specified PBI Length.
Execute_PBI	ERROR_PBI_ COMMAND_UNKNOWN	0x67	An invalid command parsed by PBI Parser
Reset Handler	ERROR_SEC_SELF_TEST	0x6a	SEC self-test failed
Execute_PBI	ERROR_PBI_ COPY_INVALID_ SRC_TYPE	0x70	Copy command, src field doesn’tmatch the RCW_SRC field
Execute_PBI	ERROR_PBI_ COPY_INVALID_ DST_ADDR	0x71	Copy command, dest field isn’t 0x00
Execute_PBI	ERROR_PBI_ COPY_INVALID_SRC_ADDR	0x72	SRC Address is invalid (ROM/ OCRAM reserved for SP)
Execute_PBI	ERROR_PBI_CCSR_BYTE_COUNT	0x74	Byte count in CCSR Write not valid
Execute_PBI	ERROR_PBI_CCSR_4_BYTE_ALLIGNED	0x75	Offset is not 4 byte aligned
Execute_PBI	ERROR_PBI_ACSR_INVALID_ADDRESS	0x78	Source address in ACSR invalid (invalid addresses - OCRAM or ROM address)
Execute_PBI	ERROR_PBI_ACSR_BYTE_COUNT	0x79	Byte count in ACSR write command not valid
Execute_PBI	ERROR_PBI_ACSR_WINDOW_NOT_SET	0x7a	ATU Window is not configured
Execute_PBI	ERROR_PBI_ALTCFG_WNDW_INVALID	0x7c	ATU Window is invalid
Execute_PBI	ERROR_PBI_JUMP_OUT_LENGTH	0x80	Offset specified in JUMP command doesn’t lie in PBI length range
Execute_PBI	ERROR_PBI_JUMP_4_BYTE_ALLIGNED	0x81	Offset specified in JUMP command is not 4 byte aligned
Execute_PBI	ERROR_PBI_JUMP_OFFSET_INVALID	0x82	Offset specified in JUMP command is invalid
	ERROR_PBI_LOADC_4_BYTE_ALLIGNED	0x84	Address specified in LOAD
Execute_PBI			Condition command is not 4 byte aligned
Execute_PBI	ERROR_PBI_JUMPC_OUT_LENGTH	0x88	Offset specified in JUMP command doesn’t lie in PBI length range
Execute_PBI	ERROR_PBI_JUMPC_4_BYTE_ALLIGNED	0x89	Offset specified in JUMP Conditional command is not 4 byte aligned
Execute_PBI	ERROR_PBI_JUMPC_CONDITION_NOT_SET	0x8A	Jump conditional command encountered before condition is set using Load Condition
Execute_PBI	ERROR_PBI_CRC_MISMATCH	0x90	CRC Mismatch

Table continues on the next page...

Table 7-22. Error Codes (continued)

When Error is Generated	Error Code	Value	Description
Execute_PBI	ERROR_PBI_POLL	0x91	Poll Timeout
Execute_PBI	ERROR_PBI_POLL_4_BYTE_ALLIGNED	0x92	Address specified in Poll Command is not 4 byte alligned
Execute_PBI	ERROR_PBI_BOOT1_CSF_INVALID_ADDR	0x94	Address of CSF Header is not valid.
Execute_PBI	ERROR_PBI_BOOT1_CSF_ALLIGNED	0x95	Address of CSF Header is not 4 byte aligned
Execute_PBI	ERROR_PBI_CCSR_OPS_TYPE_INVALID	0xA4	Only two operations are valid SET and CLEAR

Chapter 8
Device Configuration

8.1 Introduction

The Device Configuration Unit provides general purpose configuration and status for the device.

8.2 Features

The device configuration unit features the following:

- Pin sampling of device configuration pins at power-on reset and a corresponding POR status register for capturing the values of these configuration pins
- Reset Configuration Word (RCW) support via a set of RCW status registers written by the Preboot Loader (PBL) during power-on or hard reset in the PBL's RCW stage
- Boot release registers(s) used for releasing cores for booting
- Register file for the Reset module including:
  - Register for initiating a device RESET_REQ_B through software
  - Set of registers for control and status of sources on the device which can drive the device's RESET_REQ_B pin
- Core and device disable registers used for gating off clocks for any IP blocks or cores which are not used at all by an application
- Two small sets of scratch registers:
  - One set of read / write scratch registers
  - One set of write-once / read scratch registers

8.3 Device Config register descriptions

The tables below shows the memory-mapped CCSR of the Device Configuration module and lists the offset, name, and a cross-reference to the complete description of each register. These registers only support 32-bit accesses.

8.3.1 DCFG_ccsr memory map

DCFG base address: 1E0_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	POR Status Register 1 (PORSR1)	32	RO	Table 8-
4h	POR Status Register 2 (PORSR2)	32	RO	Table 8-
20h	General-Purpose POR Configuration Register (GPPORCR1)	32	RO	0000_0000h
60h	Fuse Status Register (FUSESR)	32	RO	0000_0000h
70h	Device Disable Register 1 (DEVDISR1)	32	RW	0000_0000h
74h	Device Disable Register 2 (DEVDISR2)	32	RW	0000_0000h
78h	Device Disable Register 3 (DEVDISR3)	32	RW	0000_0000h
7Ch	Device Disable Register 4 (DEVDISR4)	32	RW	0000_0000h
80h	Device Disable Register 5 (DEVDISR5)	32	RW	0000_0000h
84h	Device Disable Register 6 (DEVDISR6)	32	RW	0000_0000h
94h	Core Disable Register (COREDISR)	32	RW	0000_0000h
A4h	System Version Register (SVR)	32	RO	8703_0110h
100h	Reset Configuration Word Status Register 1 (RCWSR1)	32	RO	0000_0000h
104h	Reset Configuration Word Status Register 2 (RCWSR2)	32	RO	0000_0000h
108h	Reset Configuration Word Status Register 3 (RCWSR3)	32	RU	0000_0000h
10Ch	Reset Configuration Word Status Register 4 (RCWSR4)	32	RU	0000_0000h
110h	Reset Configuration Word Status Register 5 (RCWSR5)	32	RO	0000_0000h
114h	Reset Configuration Word Status Register 6 (RCWSR6)	32	RO	0000_0000h
118h	Reset Configuration Word Status Register 7 (RCWSR7)	32	RO	0000_0000h
11Ch	Reset Configuration Word Status Register 8 (RCWSR8)	32	RO	0000_0000h
120h	Reset Configuration Word Status Register 9 (RCWSR9)	32	RO	0000_0000h
124h	Reset Configuration Word Status Register 10 (RCWSR10)	32	RO	0000_0000h
128h	Reset Configuration Word Status Register 11 (RCWSR11)	32	RU	0000_0000h
12Ch	Reset Configuration Word Status Register 12 (RCWSR12)	32	RO	0000_0000h
130h	Reset Configuration Word Status Register 13 (RCWSR13)	32	RO	0000_0000h
134h	Reset Configuration Word Status Register 14 (RCWSR14)	32	RO	0000_0000h
138h	Reset Configuration Word Status Register 15 (RCWSR15)	32	RO	0000_0000h
13Ch	Reset Configuration Word Status Register 16 (RCWSR16)	32	RO	0000_0000h
140h	Reset Configuration Word Status Register 17 (RCWSR17)	32	RU	0000_0000h
144h	Reset Configuration Word Status Register 18 (RCWSR18)	32	RU	0000_0000h
148h	Reset Configuration Word Status Register 19 (RCWSR19)	32	RU	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
14Ch	Reset Configuration Word Status Register 20 (RCWSR20)	32	RU	0000_0000h
150h	Reset Configuration Word Status Register 21 (RCWSR21)	32	RU	0000_0000h
154h	Reset Configuration Word Status Register 22 (RCWSR22)	32	RU	0000_0000h
158h	Reset Configuration Word Status Register 23 (RCWSR23)	32	RU	0000_0000h
15Ch	Reset Configuration Word Status Register 24 (RCWSR24)	32	RU	0000_0000h
160h	Reset Configuration Word Status Register 25 (RCWSR25)	32	RO	0000_0000h
164h	Reset Configuration Word Status Register 26 (RCWSR26)	32	RU	0000_0000h
168h	Reset Configuration Word Status Register 27 (RCWSR27)	32	RO	0000_0000h
16Ch	Reset Configuration Word Status Register 28 (RCWSR28)	32	RO	0000_0000h
170h	Reset Configuration Word Status Register 29 (RCWSR29)	32	RO	0000_0000h
174h	Reset Configuration Word Status Register 30 (RCWSR30)	32	RO	0000_0000h
178h	Reset Configuration Word Status Register 31 (RCWSR31)	32	RU	0000_0000h
17Ch	Reset Configuration Word Status Register 32 (RCWSR32)	32	RU	0000_0000h
200h - 228h	Scratch Read / Write Register n (SCRATCHRW1 - SCRATCHRW11)	32	RW	0000_0000h
22Ch	Scratch Read / Write Register n (SCRATCHRW12)	32	RW	0000_0000h
230h - 23Ch	Scratch Read / Write Register n (SCRATCHRW13 - SCRATCHRW16)	32	RW	0000_0000h
300h - 30Ch	Scratch Read Register n (SCRATCHW1R1 - SCRATCHW1R4)	32	WONCE	0000_0000h
400h	Boot location pointer low-order address (BOOTLOCPTRL)	32	RW	0000_0000h
404h	Boot location pointer high-order address (BOOTLOCPTRH)	32	RW	0000_0000h
520h - 524h	USBa_AMQR Register (USB1_AMQR - USB2_AMQR)	32	RW	0000_0000h
530h	SDMM1_AMQR Register (SDMM1_AMQR)	32	RW	0000_0000h
550h	SATA n Access Management Qualifier register (SATA1_AMQR)	32	RW	0000_0000h
570h - 578h	MISC n Access Management Qualifier register (MISC1_AMQR - MISC3_AMQR)	32	RW	0000_0000h
590h - 594h	SPARE n Access Management Qualifier register (SPARE1_AMQR - SPARE2_AMQR)	32	RW	0000_0000h
740h - 754h	Topology Initiator Type n Register (TP_ITYP0 - TP_ITYP5)	32	RO	0000_0000h
844h	Core Cluster n Topology Register (TP_CLUSTER1L)	32	RO	0000_0000h
84Ch	Core Cluster n Topology Register (TP_CLUSTER2L)	32	RO	0000_0000h
900h	QSPI Control Register (QSPICR1)	32	RW	0000_0014h
920h - 93Ch	I/O Quality of Services Register (IOQOSCR1 - IOQOSCR8)	32	RW	000D_0D0Dh
960h	FlexTimer Control Register (FTMCR)	32	RW	0000_0000h
990h	Core Disable Status Register (COREDISABLESR)	32	RO	0000_0000h
A00h	Secure Debug configuration register (SDBGCR)	32	RW	0000_0000h
BF8h	IP Block Revision Register 1 (IPBRR1)	32	RO	0280_0600h
BFCh	IP Block Revision Register 2 (IPBRR2)	32	RO	0000_0000h

8.3.2 POR Status Register 1 (PORSR1)

8.3.2.1 Offset

Register	Offset
PORSR1	0h

8.3.2.2 Function

PORSR1 captures the values of the device's POR configuration pins.

8.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RCW_SRC									0	RSP_DIS	Reserved	Reserved	IFC_TE	Reserved	Reserved
W																
Reset	u	u	u	u	u	u	u	u	u	1	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				0				SOC	0						
W																
Reset	u	u	u	u	1	1	1	1	u	1	1	1	1	1	1	1

8.3.2.4 Fields

Field	Function
31-23 RCW_SRC	Reset Configuration Word Source Indicates which Flash memory contains the RCW information. Loaded with the value of CFG_RCW_SRC[8:0].
22 UNUSED_22	Reserved
21 RSP_DIS	Reset Pause Loaded with the value of the CFG_RSP_DIS pin at power-on reset.

Table continues on the next page...

Field	Function
	0b - Reset state machine will pause at appropriate state 1b - Reset state machine will not pause at appropriate state
20 —	Reserved
19 —	Reserved
18 IFC_TE	IFC External Transceive Enable Polarity Selection Loaded with the value of the CFG_IFC_TE pin at power-on reset. 0b - Transceiver Enable polarity is configured active high 1b - Transceiver Enable polarity is configured active low
17 —	Reserved
16 —	Reserved
15-12 —	Reserved
11-8 UNUSED_11_8	Reserved
7 SOC	SoC Use Loaded with the value of the CFG_SOC_USE pin at power-on reset. 0b - Config pin was not set 1b - Config pin was set
6-0 UNUSED_6_0	Reserved

8.3.3 POR Status Register 2 (PORSR2)

8.3.3.1 Offset

Register	Offset
PORSR2	4h

8.3.3.2 Function

PORSR2 captures the values of the device's POR configuration pins.

8.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SVR_VER		Reserved									RCLK_CFG	Reserved			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

8.3.3.4 Fields

Field	Function
31-30 SVR_VER	System Version Register Version Loaded with the value of the CFG_SVR[1:0] pin at power-on reset. Used in specifying the personality of the device.
29-21 —	Reserved
20 RCLK_CFG	Power-on-reset reference clock configuration. This is loaded with the value of cfg_eng_use[0]. 0b - Differential clock (DIFF_SYSCLK) provides the System Clock 1b - SYSCLK provides the System Clock
19-0 —	Reserved

8.3.4 General-Purpose POR Configuration Register (GPPORCR1)

8.3.4.1 Offset

Register	Offset
GPPORCR1	20h

8.3.4.2 Function

GPPORCR stores the value sampled from the local bus address/data signals, LAD[7:0]integrated Flash controller address/data signals, IFC_AD[7:0], using sampling logic similar to that for the configuration pins. It is provided for customer use and always samples 8-bits, independent of the width of an SoC flash controller's LAD bus. Software can use this value to inform the operating system about initial system configuration. Typical interpretations include circuit board type, board ID number, or a list of available peripherals.

8.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								POR_CFG_VEC							
W																
Reset	0	0	0	0	0	0	0	0	0[1]	0	0	0	0	0	0	0

1. Reset value supplied by cfg_gpinput[7:0].

8.3.4.4 Fields

Field	Function
31-8 —	Reserved
7-0 POR_CFG_VEC	General-purpose POR configuration vector Sampled from local bus address/data signals, LAD[7:0]integrated Flash controller address/data signals, IFC_AD[7:0] signals (uppermost 8-bits) at the negation of PORESET. **NOTE:** Reset value supplied by cfg_gpinput[7:0].

8.3.5 Fuse Status Register (FUSESR)

8.3.5.1 Offset

Register	Offset
FUSESR	60h

8.3.5.2 Function

The fuse status register provides the values from on-chip voltage ID efuses programmed at the factory. These values define the voltage requirements for the chip. Boot software reads FUSESR and translates the values into the appropriate commands to set the voltage output value of an external voltage regulator.

8.3.5.3 Diagram

8.3.5.4 Fields

Field	Function
31-30 —	Reserved
29-25 DA_V	VDD voltage This is the minimum voltage required to reach the frequency specified. Note that most of the possible voltages levels will never be specified by the factory. 00000b - 1.025 V (default) 00001b - 0.9875 V

Table continues on the next page...

Field	Function
	00010b - 0.9750 V 00011b - Reserved 00100b - Reserved 00101b - Reserved 00110b - Reserved 00111b - Reserved 01000b - 0.9 V 01001b - Reserved 01010b - Reserved 01011b - Reserved 01100b - Reserved 01101b - Reserved 01110b - Reserved 01111b - Reserved 10000b - 1.0000 V 10001b - 1.0125 V 10010b - 1.0250 V 10011b - Reserved 10100b - Reserved 10101b - Reserved 10110b - Reserved 10111b - Reserved 11000b - Reserved 11111b - Reserved
24-20 DA_ALT_V	This is a secondary voltage field for VDD. The field encodings are the same as DA_V, except 11111 is reserved. **NOTE:** If this field is not all zeros, the the correct VID value should be read from here. Otherwise, the correct VID value should be read from DA_V.
19-0 —	Reserved

8.3.6 Device Disable Register 1 (DEVDISR1)

8.3.6.1 Offset

Register	Offset
DEVDISR1	70h

8.3.6.2 Function

Device Disable Register n

A given application may not use all the peripherals on the device. In this case, it may be desirable to disable unused peripherals. DEVDISRn provides a mechanism for gating clocks to IP blocks that are not used when running an application.

NOTE

IP Blocks disabled by setting the corresponding bit in the DEVDISRn register must not be re-enabled via software. The only supported mechanism for re-enabling an IP block disabled in this manner is via power-on or hard reset.

8.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									SEC	Reserved					SATA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		USB2	USB1	Reserved			qDMA	Reserved				HDLC	eSDHC	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.6.4 Fields

Field	Function
31-23 —	Reserved
22 SEC	SEC
21-17 —	Reserved
16 SATA	SATA
15-14 —	Reserved
13	USB2

Table continues on the next page...

Field	Function
USB2	
12 USB1	USB1
11-9 —	Reserved
8 qDMA	qDMA
7-4 —	Reserved
3 HDLC	HDLC/uQE
2 eSDHC	eSDHC
1-0 —	Reserved

8.3.7 Device Disable Register 2 (DEVDISR2)

8.3.7.1 Offset

Register	Offset
DEVDISR2	74h

8.3.7.2 Function

Device Disable Register n

A given application may not use all the peripherals on the device. In this case, it may be desirable to disable unused peripherals. DEVDISRn provides a mechanism for gating clocks to IP blocks that are not used when running an application.

NOTE

IP Blocks disabled by setting the corresponding bit in the DEVDISRn register must not be re-enabled via software. The only supported mechanism for re-enabling an IP block disabled in this manner is via power-on or hard reset.

8.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						MAC_10	MAC_9	MAC_8	MAC_7	MAC_6	MAC_5	MAC_4	MAC_3	MAC_2	MAC_1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.7.4 Fields

Field	Function
31-10 —	Reserved
9 MAC_10	MAC_10
8 MAC_9	MAC_9
7 MAC_8	MAC_8
6 MAC_7	MAC_7
5 MAC_6	MAC_6
4 MAC_5	MAC_5
3 MAC_4	MAC_4
2 MAC_3	MAC_3
1 MAC_2	MAC_2
0 MAC_1	MAC_1

8.3.8 Device Disable Register 3 (DEVDISR3)

8.3.8.1 Offset

Register	Offset
DEVDISR3	78h

8.3.8.2 Function

Device Disable Register n

A given application may not use all the peripherals on the device. In this case, it may be desirable to disable unused peripherals. DEVDISRn provides a mechanism for gating clocks to IP blocks that are not used when running an application.

NOTE

IP Blocks disabled by setting the corresponding bit in the DEVDISRn register must not be re-enabled via software. The only supported mechanism for re-enabling an IP block disabled in this manner is via power-on or hard reset.

8.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															AIOP
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		MC	QBMAN	Reserved									PCIe3	PCIe2	PCIe1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.8.4 Fields

Field	Function
31-17 —	Reserved
16 AIOP	AIOP
15-14 —	Reserved
13 MC	MC
12 QBMAN	QBMAN
11-3 —	Reserved
2 PCIe3	PCIe3
1 PCIe2	PCIe2
0 PCIe1	PCIe1

8.3.9 Device Disable Register 4 (DEVDISR4)

8.3.9.1 Offset

Register	Offset
DEVDISR4	7Ch

8.3.9.2 Function

Device Disable Register n

A given application may not use all the peripherals on the device. In this case, it may be desirable to disable unused peripherals. DEVDISRn provides a mechanism for gating clocks to IP blocks that are not used when running an application.

NOTE

IP Blocks disabled by setting the corresponding bit in the DEVDISRn register must not be re-enabled via software. The only supported mechanism for re-enabling an IP block disabled in this manner is via power-on or hard reset.

8.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						I2C4	I2C3	Reserved		SPI	QSPI	DUART2	DUART1	I2C2	I2C1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.9.4 Fields

Field	Function
31-10 —	Reserved
9 I2C4	I2C4
8 I2C3	I2C3
7-6 —	Reserved
5 SPI	SPI
4 QSPI	QSPI
3 DUART2	DUART2
2 DUART1	DUART1

Table continues on the next page...

Field	Function
1 I2C2	I2C2
0 I2C1	I2C1

8.3.10 Device Disable Register 5 (DEVDISR5)

8.3.10.1 Offset

Register	Offset
DEVDISR5	80h

8.3.10.2 Function

Device Disable Register n

A given application may not use all the peripherals on the device. In this case, it may be desirable to disable unused peripherals. DEVDISRn provides a mechanism for gating clocks to IP blocks that are not used when running an application.

NOTE

IP Blocks disabled by setting the corresponding bit in the DEVDISRn register must not be re-enabled via software. The only supported mechanism for re-enabling an IP block disabled in this manner is via power-on or hard reset.

8.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											PEBM	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	FlexTImers1_4	TMU	OCRAM	Reserved	DBG	GPIO	IFC	Reserved							DDR1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.10.4 Fields

Field	Function
31-21 —	Reserved
20 PEBM	PEBM
19-15 —	Reserved
14 FlexTImers1_4	FlexTImers1-4
13 TMU	TMU
12 OCRAM	OCRAM
11 —	Reserved
10 DBG	DBG
9 GPIO	GPIO
8 IFC	IFC

Table continues on the next page...

Field	Function
7-1 —	Reserved
0 DDR1	DDR1

8.3.11 Device Disable Register 6 (DEVDISR6)

8.3.11.1 Offset

Register	Offset
DEVDISR6	84h

8.3.11.2 Function

Device Disable Register n

A given application may not use all the peripherals on the device. In this case, it may be desirable to disable unused peripherals. DEVDISRn provides a mechanism for gating clocks to IP blocks that are not used when running an application.

NOTE

IP Blocks disabled by setting the corresponding bit in the DEVDISRn register must not be re-enabled via software. The only supported mechanism for re-enabling an IP block disabled in this manner is via power-on or hard reset.

8.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															WRIOP
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.11.4 Fields

Field	Function
31-1 —	Reserved
0 WRIOP	WRIOP

8.3.12 Core Disable Register (COREDISR)

8.3.12.1 Offset

Register	Offset
COREDISR	94h

8.3.12.2 Function

COREDISR provides a mechanism for disabling and power gating any physical cores on the device that are not used when running an application.

COREDISR register should only be configured in the following conditions:

- Before system ready, COREDISR register can be programmed by an external debugger, external agent (PCIe, etc.) or Pre-Boot Initialization.
- After system ready, a COREDISR register bit can be programmed for the corresponding core by an external debugger, external agent (PCIe, etc.) or system initialization (e.g. uboot) software while the core to be disabled is in boot hold-off mode

NOTE

Cores disabled by setting the corresponding bit in the COREDISR register must not be re-enabled via software. The only supported mechanism for re-enabling a Core disabled in this manner is via power-on reset, hard reset, device warm reset, or core warm reset. In the core warm reset case, the corresponding bit in COREDISRn must first be cleared and then the core warm reset can be issued.

NOTE

Cores that have an interrupt pending will not be disabled by COREDISR until the interrupt is cleared.

8.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.12.4 Fields

Field	Function
31-8 —	Reserved
7 CD7	Core 7 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled

Table continues on the next page...

Field	Function
6 CD6	Core 6 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
5 CD5	Core 5 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
4 CD4	Core 4 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
3 CD3	Core 3 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
2 CD2	Core 2 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
1 CD1	Core 1 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
0 CD0	Core 0 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled

8.3.13 System Version Register (SVR)

8.3.13.1 Offset

Register	Offset
SVR	A4h

8.3.13.2 Function

The SVR contains the system version number for the device.

8.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MFR_ID				FAMILY				Reserved		SOC_DEV_ID					
W																
Reset	1	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		VAR_PER						MAJOR_REV				MINOR_REV			
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

8.3.13.4 Fields

Field	Function
31-28 MFR_ID	Manufacturer ID
27-24 FAMILY	Family
23-22 —	Reserved
21-16 SOC_DEV_ID	SoC Device ID
15-14 —	Reserved
13-8 VAR_PER	Various Personalities 000000b - LS1088A with security 000001b - LS1088A without security 000010b - LS1084A with security 000011b - LS1084A without security 100000b - LS1048A with security 100001b - LS1048A without security 100010b - LS1044A with security 100011b - LS1044A without security
7-4	Major Revision Number

Table continues on the next page...

Field	Function
MAJOR_REV	
3-0 MINOR_REV	Minor Revision Number

8.3.14 Reset Configuration Word Status Register 1 (RCWSR1)

8.3.14.1 Offset

Register	Offset
RCWSR1	100h

8.3.14.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CGA_PLL1_RAT						CGA_PLL1_CFG		Reserved						Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MEM_PLL_RAT						MEM_PLL_CFG		Reserved	SYS_PLL_RAT					SYS_PLL_CFG	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.14.4 Fields

Field	Function
31-26 CGA_PLL1_RAT	Cluster Group A PLL 1 Multiplier/Ratio Configures the ratio between the System Clock and the CGAn PLL output.
25-24 CGA_PLL1_CFG	Cluster Group A PLL 1 Configuration
23-18 —	Reserved
17-16 —	Reserved
15-10 MEM_PLL_RAT	Memory Controller Complex PLL Multiplier/Ratio (applies to all Class 1 DDR controllers). Configures the ratio between the DDR Reference Clock and the DDR data rate. See RCW[DDR_REFCLK_SEL] which selects the DDR Reference clock. **NOTE:** Only async mode of operation is supported for the memory controller complex.
9-8 MEM_PLL_CFG	Memory Controller Complex PLL Configuration (applies to all Class 1 DDR controllers).
7 —	Reserved
6-2 SYS_PLL_RAT	System PLL Multiplier/Ratio Configures the ratio between the System Clock and the SoC's coherency domain. See XREF which specifies how the Platform clock relates to the SoC's Coherency Domain clock. **NOTE:** Odd ratios are not supported when the Coherency domain runs at double the rate of the Platform clock.
1-0 SYS_PLL_CFG	System PLL Configuration.

8.3.15 Reset Configuration Word Status Register 2 (RCWSR2)

8.3.15.1 Offset

Register	Offset
RCWSR2	104h

8.3.15.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						Reserved		Reserved						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						Reserved		CGA_PLL2_RAT						CGA_PLL2_CFG	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.15.4 Fields

Field	Function
31-26 —	Reserved
25-24 —	Reserved
23-18 —	Reserved
17-16 —	Reserved
15-10 —	Reserved
9-8 —	Reserved

Table continues on the next page...

Field	Function
7-2 CGA_PLL2_RAT	Cluster Group A PLL 2 Multiplier/Ratio See CGA_PLL1_RAT
1-0 CGA_PLL2_CFG	Cluster Group A PLL 2 Configuration See CGA_PLL1_CFG

8.3.16 Reset Configuration Word Status Register 3 (RCWSR3)

8.3.16.1 Offset

Register	Offset
RCWSR3	108h

8.3.16.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						Reserved		Reserved						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Reserved		Reserved						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.16.4 Fields

Field	Function
31-26 —	Reserved
25-24 —	Reserved
23-18 —	Reserved
17-16 —	Reserved
15-10 —	Reserved
9-8 —	Reserved
7-2 —	Reserved
1-0 —	Reserved

8.3.17 Reset Configuration Word Status Register 4 (RCWSR4)

8.3.17.1 Offset

Register	Offset
RCWSR4	10Ch

8.3.17.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						Reserved		Reserved						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Reserved		Reserved						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.17.4 Fields

Field	Function
31-26 —	Reserved
25-24 —	Reserved
23-18 —	Reserved
17-16 —	Reserved
15-10 —	Reserved
9-8 —	Reserved
7-2 —	Reserved
1-0 —	Reserved

8.3.18 Reset Configuration Word Status Register 5 (RCWSR5)

8.3.18.1 Offset

Register	Offset
RCWSR5	110h

8.3.18.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				C2_PLL_SEL				C1_PLL_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CGA_PLL2_SPD	CGA_PLL1_SPD	Reserved	Reserved	MEM_PLL_SPD	SYS_PLL_SPD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.18.4 Fields

Field	Function
31-28 —	Reserved
27-24 —	Reserved
23-20 C2_PLL_SEL	Cluster 2 PLL Select Selects a clock from one of up to two or three available Cluster Group n PLLs
19-16	Cluster 1 PLL Select

Table continues on the next page...

Field	Function
C1_PLL_SEL	Selects a clock from one of up to two or three available Cluster Group 1 PLLs
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5 CGA_PLL2_SPD	Cluster Group A PLL 2 Speed Select See SYS_PLL_SPD
4 CGA_PLL1_SPD	Cluster Group A PLL 1 Speed Select See SYS_PLL_SPD
3 —	Reserved
2 —	Reserved
1 MEM_PLL_SPD	Memory Controller Complex PLL Speed Select See SYS_PLL_SPD
0 SYS_PLL_SPD	System PLL Speed Select See Hardware Spec for correct setting.

8.3.19 Reset Configuration Word Status Register 6 (RCWSR6)

8.3.19.1 Offset

Register	Offset
RCWSR6	114h

8.3.19.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	Reserved			Reserved			Reserved			HWA_CGA_M2_CLK_SEL			HWA_CGA_M1_CLK_SEL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				Reserved				Reserved				Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.19.4 Fields

Field	Function
31 —	Reserved
30-28 —	Reserved

Table continues on the next page...

Field	Function
27-25 —	Reserved
24-22 —	Reserved
21-19 HWA_CGA_M2 _CLK_SEL	Hardware Accelerator Block Cluster Group A Mux 2 Clock Select Allows for a platform accelerator block's (or multiple blocks') frequency to be maximized (using async clock) by leveraging Cluster Group A PLL 1 or 2 as the source clock. There may be other options selectable by this mux depending on the SoC's implementation (clock domain 'X' and 'Y' below). This bit selects the input clock for eSDHC.
18-16 HWA_CGA_M1 _CLK_SEL	Hardware Accelerator Block Cluster Group A Mux 1 Clock Select Allows for a platform accelerator block's (or multiple blocks') frequency to be maximized (using async clock) by leveraging Cluster Group A PLL 1 or 2 as the source clock. There may be other options selectable by this mux depending on the SoC's implementation (clock domain 'X' and 'Y' below). This bit selects the input clock for QuadSPI.
15-12 —	Reserved
11-8 —	Reserved
7-4 —	Reserved
3-0 —	Reserved

8.3.20 Reset Configuration Word Status Register 7 (RCWSR7)

8.3.20.1 Offset

Register	Offset
RCWSR7	118h

8.3.20.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	Reserved		Reserved		Reserved	Reserved	Reserved	DDR_RATE	DRAM_LAT		DDR_REFCLK_SEL		Reserved	Reserved	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved			Reserved			Reserved			Reserved			Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.20.4 Fields

Field	Function
31 —	Reserved
30-29 —	Reserved
28-27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23 DDR_RATE	DDR Data Rate This setting is defined by the factory.
22-21 DRAM_LAT	DDR Latency This applies to all DDR Controllers (Class 1 and Class 2). **NOTE:** if DDR latency is not known at the time the RCW is loaded, it is acceptable to conservatively configure this field as 2'b01. This field is used for optimizing the DDRinterface. No functional issues result if this field does not match the latency of the actual DRAM's used.

Table continues on the next page...

Field	Function
20-19 DDR_REFCLK_SEL	DDR Reference Clock Selection Selects the Reference Clock for the DDR PLL
18 —	Reserved
17 —	Reserved
16-14 —	Reserved
13-11 —	Reserved
10-8 —	Reserved
7-5 —	Reserved
4-2 —	Reserved
1-0 —	Reserved

8.3.21 Reset Configuration Word Status Register 8 (RCWSR8)

8.3.21.1 Offset

Register	Offset
RCWSR8	11Ch

8.3.21.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									REQD_CUST_CONFIG_230_229		Reserved		Reserved		Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.21.4 Fields

Field	Function
31-7 —	Reserved
6-5 REQD_CUST_CONFIG_230_229	Required Customer Configuration
4-3 —	Reserved
2-1 —	Reserved
0 —	Reserved

8.3.22 Reset Configuration Word Status Register 9 (RCWSR9)

8.3.22.1 Offset

Register	Offset
RCWSR9	120h

8.3.22.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PBI_LENGTH												FLASH_MODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FLASH_MODE					SB_EN	BOOT_HO	BOOT_LOC					Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.22.4 Fields

Field	Function
31-20 PBI_LENGTH	Pre-Boot Initialization Length Size in bytes of the PBI image up until the final requires STOP or CRC and STOP command. The length specified here does not include the Preamble or RCW records. If there is no PBI image, this value must be set to zero.

Table continues on the next page...

Field	Function
	NOTE: A value of zero means that there are no PBI commands to execute as described in Section 24.5.3.3, “Boot 0 Pre-Boot Loader Routine”. This is the means for skipping the reset sequence’s PBI phase.
19-11 FLASH_MODE	Flash Mode This field specifies the mode for the IFC or Serial NOR which contains the boot code. It is always used in conjunction with RCW[BOOT_LOC]. Specifically, It is used for configuring the flash interface (IFC or Serial NOR) for cases where the RCW (and optionally PBI) source was something other than one of these interfaces. The field is only valid when BOOT_LOC specifies either "IFC" or "Serial NOR". Note that CFG_RCW_SRC may have precedence over configuring the IFC or Serial NOR. See the definition for the BOOT_LOC field. For the case where this is used for Serial NOR configuration as described in RCW[BOOT_LOC], see options below.
10 SB_EN	Secure Boot Enable Secure boot is enabled if either this RCW bit is set or the ‘Intent to Secure’ fuse value is set. The intended use of the SB_EN bit is to emulate the effect of the ITS fuse. SB_EN itself cannot replace the use of the ITS fuse for production systems.
9 BOOT_HO	Boot Holdoff This field is used to specify which cores remain in boot holdoff on exiting reset. Note that the core released for boot, referred to as "core X", can be identified using the read-only BRCORENBR[CORE_NBR] value provided by the Reset block.
8-4 BOOT_LOC	Boot Location This field indicates the default boot interface from which the core fetches instructions. **NOTE:** This field is only used when the DCFG boot location pointer registers are null. Using the boot location pointer register is the preferred method when there is a pre-boot initialization image. In simpler cases with no PBI, RCW[BOOT_LOC] is the means for locating the boot image. The following restrictions apply: - Only one IFC mode can be used for CFG_RCW_SRC and BOOT_LOC - Similarly, only one Serial NOR mode can be used for CFG_RCW_SRC and BOOT_LOC See RCW, PBI, and Boot Options for a summary of allowed options. If the IFC encoding is selected RCW[FLASH_MODE] configures the IFC, provided CFG_RCW_SRC[] has not configured IFC already. If the Serial NOR encoding is selected: RCW[FLASH_MODE] configures the Serial NOR, provided CFG_RCW_SRC[] has not configured it already.
3-0 —	Reserved

8.3.23 Reset Configuration Word Status Register 10 (RCWSR10)

8.3.23.1 Offset

Register	Offset
RCWSR10	124h

8.3.23.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								GPIO_LED_EN	GPIO_LED_NUM						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		SYSCLK_FREQ										Reserved			SDBGEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.23.4 Fields

Field	Function
31-24 —	Reserved
23	GPIO LED Enable

Table continues on the next page...

Field	Function
GPIO_LED_EN	Enable for GPIO output assertion on Secure Boot Failure. This bit enables the indication of a Secure Boot failure by driving a user selected GPIO output as asserted. The selected GPIO output is used to drive a LED. See XREF.
22-16 GPIO_LED_NUM	GPIO LED Number Selected GPIO Output bit for indicating Secure Boot Failure. This field selects the GPIO output pin to indicate a secure boot failure when the feature is enabled by RCW[GPIO_LED_EN]. This field is ignored if RCW[GPIO_LED_EN] is 1'b0.
15-14 —	Reserved
13-4 SYSCLK_FREQ	System Clock Frequency This field is provided so that software can determine the System Clock Frequency of the SoC. In addition, this field is used by hardware for proper configuration of the Arm Generic Timer. The encoded value is multiplied by 166.667 KHz to determine the frequency of the System Clock (see XREF). The allowable range of values of this field is SoC-specific, as it depends on the range of SYSCLK frequencies supported.
3-1 —	Reserved
0 SDBGEN	Secure Debug Enable Secure Debug (CoreSight SPIDEN) is enabled after RCW is loaded if this RCW bit is set and the 'Intent to Secure' fuse value is cleared. See Table 7-8 for exact behavior.

8.3.24 Reset Configuration Word Status Register 11 (RCWSR11)

8.3.24.1 Offset

Register	Offset
RCWSR11	128h

8.3.24.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.24.4 Fields

Field	Function
31-0 —	Reserved

8.3.25 Reset Configuration Word Status Register 12 (RCWSR12)

8.3.25.1 Offset

Register	Offset
RCWSR12	12Ch

8.3.25.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					IRQ_EXT			IRQ2_EXT			IRQ1_EXT			SDHC_EXT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SDHC_EXT	SPI_EXT			IIC4_EXT			IIC3_EXT			IIC2_EXT			UART_EXT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.25.4 Fields

Field	Function
31-27 —	Reserved
26-24 IRQ_EXT	IRQ3-11 Pin Configuration ("Extension" portion) Together with IRQ_BASE field, this field configures functionality of IRQ[3:11] pins.
23-21 IRQ2_EXT	IRQ2 Pin Configuration ("Extension" portion) This field configures functionality of IRQ[2] pin
20-18 IRQ1_EXT	IRQ1 Pin Configuration ("Extension" portion) This field configures functionality of IRQ[1] pin
17-15 SDHC_EXT	SDHC Pin Configuration ("Extension" portion) Together with SDHC_BASE_BASE and SDHC_PCS_BASE fields, this field configures functionality of SPI_* pins.
14-12 SPI_EXT	SPI Pin Configuration ("Extension" portion) Together with SPI_BASE_BASE and SPI_PCS_BASE fields, this field configures functionality of SPI_* pins.
11-9 IIC4_EXT	IIC4 Pin Configuration ("Extension" portion) Together with IIC4_BASE field, this field configures functionality of IIC4_* pins.
8-6 IIC3_EXT	IIC3 Pin Configuration ("Extension" portion) Together with IIC3_BASE field, this field configures functionality of IIC3_* pins.
5-3 IIC2_EXT	IIC2 Pin Configuration ("Extension" portion) Together with IIC2_BASE field, this field configures functionality of IIC2_* pins.

Table continues on the next page...

Field	Function
2-0 UART_EXT	UART Pin Configuration ("Extension" portion) Together with UART_BASE field, this field configures functionality of UART pins: {UART1_SOUT, UART1_SIN, UART1_RTS_B, UART1_CTS_B, UART2_SOUT, UART2_SIN, UART2_RTS_B, UART2_CTS_B}

8.3.26 Reset Configuration Word Status Register 13 (RCWSR13)

8.3.26.1 Offset

Register	Offset
RCWSR13	130h

8.3.26.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							ASLEEP	RTC	EVT_9	IRQ03_BASE	IRQ04_BASE	IRQ05_BASE	IRQ06_BASE	IRQ07_BASE	IRQ08_BASE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ09_BASE	IRQ10_BASE	IRQ11_BASE	SDHC_BASE	SPI_PCS_BASE		SPI_BASE_BASE		IIC4_BASE		IIC3_BASE		IIC2_BASE		UART_BASE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.26.4 Fields

Field	Function
31-25 —	Reserved
24 ASLEEP	ASLEEP Pin Configuration This field configures functionality of ASLEEP pin.
23 RTC	RTC Pin Configuration This field configures functionality of RTC pin.
22 EVT_9	EVT_9 Pin Configuration This field configures functionality of the EVT[9] pin. **NOTE:** EVT[9] functionality, if selected by this field, defaults to input functionality until reconfigured by software to output functionality.
21 IRQ03_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
20 IRQ04_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.

Table continues on the next page...

Field	Function
19 IRQ05_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
18 IRQ06_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
17 IRQ07_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
16 IRQ08_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
15 IRQ09_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
14 IRQ10_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
13 IRQ11_BASE	IRQ Pin Configuration This field configures functionality of IRQ[3:11] pins **NOTE:** for SoC's which have IRQ_EXT field as valid: IRQ_EXT field must be all 0's for IRQ_BASE to take effect.
12 SDHC_BASE	SDHC Pin Configuration ("Base" portion) This field configures functionality of SDHC pins {SDHC_CMD, SDHC_DAT[0:3], SDHC_CLK}. **NOTE:** for SoC's which have SDHC_EXT field as valid: SDHC_EXT field must be all 0's for SDHC_BASE to take effect. **NOTE:** the SDHC pins are driven with SDHC functionality regardless of the setting of this field or the SDHC extension field if CFG_RCW_SRC selects SD/MMC.
11-10 SPI_PCS_BASE	SPI_PCS Pin Configuration ("Base" portion) This field configures functionality of the SPI_PCS[0:3] pins. **NOTE:** for SoC's which have SPI_EXT field as valid: SPI_EXT field must be all 0's for SPI_PCS_BASE to take effect.
9-8	SPI Pin Configuration ("Base" portion) This field configures functionality of the SPI_SOUT, SPI_SIN, SPI_SCK pins.

Table continues on the next page...

Field	Function
SPI_BASE_BASE	**NOTE:** for SoC's which have SPI_EXT field as valid: SPI_EXT field must be all 0's for SPI_BASE_BASE to take effect.
7-6 IIC4_BASE	IIC4 Pin Configuration ("Base" portion) This field configures functionality of IIC4_* pins.
5-4 IIC3_BASE	IIC3 Pin Configuration ("Base" portion) This field configures functionality of IIC3_* pins.
3-2 IIC2_BASE	IIC2 Pin Configuration ("Base" portion) This field configures functionality of IIC2_* pins.
1-0 UART_BASE	UART Pin Configuration ("Base" portion) This field configures functionality of 6 UART pins { UART1_RTS_B, UART1_CTS_B, UART2_SOUT, UART2_SIN, UART2_RTS_B, UART2_CTS_B} It does not control UART1_SOUT, UART1_SIN which have no functional multiplexed modes. **NOTE:** for SoC's which have UART_EXT field as valid: UART_EXT field must be all 0's for UART_BASE to take effect.

8.3.27 Reset Configuration Word Status Register 14 (RCWSR14)

8.3.27.1 Offset

Register	Offset
RCWSR14	134h

8.3.27.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					IFC_GRP_I_EXT			IFC_GRP_H_EXT			IFC_GRP_G_EXT			IFC_GRP_F_EXT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IFC_GRP_F_EXT	IFC_GRP_E_EXT			IFC_GRP_D_EXT			IFC_GRP_C_EXT			IFC_GRP_B_EXT			IFC_GRP_A_EXT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.27.4 Fields

Field	Function
31-27 —	Reserved
26-24 IFC_GRP_I_EXT	IFC_GRP_I_EXT Pin Configuration ("Extension" portion) **NOTE:** this field configures functionality of "Group I" of the IFC pins {IFC_WE2_B, IFC_WE3_B, IFC_BCTL, IFC_NDDQS, IFC_RB_B[0], IFC_CLK[0:1], IFC_NDWE_B]}.
23-21 IFC_GRP_H_EXT	IFC_GRP_H_EXT Pin Configuration ("Extension" portion) This field configures functionality of "Group H" of the IFC pins {IFC_AD[0:7], IFC_WE0_B, IFC_TE, IFC_AVD, IFC_CLE, IFC_OE_B, IFC_WP_B[0]}. **NOTE:** extension use of these pins are output-only.
20-18 IFC_GRP_G_EXT	IFC_GRP_G_EXT Pin Configuration ("Extension" portion) This field configures functionality of "Group G" of the IFC pins {IFC_AD[8:15]}.
17-15 IFC_GRP_F_EXT	IFC_GRP_F_EXT Pin Configuration ("Extension" portion) This field configures functionality of "Group F" of the IFC pins {IFC_AD[16:24]}.
14-12	IFC_GRP_E_EXT Pin Configuration ("Extension" portion)

Table continues on the next page...

Field	Function
IFC_GRP_E_EXT	Together with IFC_GRP_E_BASE field, this field configures functionality of "Group E1, E2 and E3" of the IFC pins {IFC_CS_B[1:3]}.
11-9 IFC_GRP_D_EXT	IFC_GRP_D_EXT Pin Configuration ("Extension" portion) Together with IFC_GRP_D_BASE field, this field configures functionality of "Group D" of the IFC pins {IFC_PAR[0:1], IFC_PERR_B, IFC_PAR[2:3]}.
8-6 IFC_GRP_C_EXT	IFC_GRP_C_EXT Pin Configuration ("Extension" portion) Together with IFC_GRP_C_BASE field, this field configures functionality of “Group C” of the IFC pins {IFC_A[26:31]} used in 32b data bus pinouts of the IFC.
5-3 IFC_GRP_B_EXT	IFC_GRP_B_EXT Pin Configuration ("Extension" portion) Together with IFC_GRP_B_BASE field, this field configures functionality of "Group B" of the IFC pins ({IFC_AD[28:31} for 32b data bus pinouts or {IFC_A[28:31]} for 16b data bus pinouts which pinout 32b of addressability).
2-0 IFC_GRP_A_EXT	IFC_GRP_A_EXT Pin Configuration ("Extension" portion) Together with IFC_GRP_A_BASE field, this field configures functionality of "Group A" of the IFC pins ({IFC_AD[25:27]} for 32b data bus pinouts or {IFC_A[25:27]} for 16b data bus pinouts which pinout 28b or 32b of addressability).

8.3.28 Reset Configuration Word Status Register 15 (RCWSR15)

8.3.28.1 Offset

Register	Offset
RCWSR15	138h

8.3.28.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		QSPI_OCT_EN	IFC_A_8_6	IFC_GRP_FGHI_BASE		IFC_GRP_E_BASE		IFC_GRP_D_BASE		Reserved	IFC_GRP_C_BASE	IFC_GRP_B_BASE		IFC_GRP_A_BASE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.28.4 Fields

Field	Function
31-14 —	Reserved
13 QSPI_OCT_EN	QSPI Pin Configuration This field configures the functionality of QuadSPI (multiplexed on IFC pins) This field is only used if RCW_SRC or PBI_SRC or IFC_GRP_A_BASE is configured to QuadSPI, as it only affects pinout when QuadSPI operation is selected.
12 IFC_A_8_6	IFC_A_8_6 Pin Configuration This field configures functionality of the IFC pins IFC_A[22:24] for 16b data bus pinouts which pinout 28b of addressability. This field is ignored if IFC_GRP_FGHI_BASE is non-zero.
11-10 IFC_GRP_FGHI_BASE	IFC_GRP_FGHI_BASE Pin Configuration ("Base" portion) This field configures functionality of "Group F, G, H and I" of the IFC pins {IFC_AD/IFC_A[0:24]} This field is ignored for any pinsthat have multiplexed QuadSPI functionality if RCW_SRC or PBI_SRC or IFC_GRP_A_BASE are configured to QuadSPI, as any of these events configured to QuadSPI forces such pins to QuadSPI operation, rather than whatever is programmed in this field. Pins IFC_CS_B[0], IFC_WE*_B, IFC_BCTL, IFC_TE, IFC_NDDQS, IFC_AVD, IFC_CLE,IFC_OE_B, IFC_WP_B[0], IFC_RB_B[0], IFC_CLK[0:n], IFC_NDWE_B all have the same functionality (either IFC or GPIO) as IFC_AD[0:7] as defined by this field.
9-8 IFC_GRP_E_BASE	IFC_GRP_E_BASE Pin Configuration ("Base" portion) Together with IFC_GRP_E_EXT field, this field configures functionality of "Group E1" of the IFC pins {IFC_RB_B[1],IFC_CS_B[1:7]}

Table continues on the next page...

Field	Function
7-6 IFC_GRP_D_BASE	IFC_GRP_D_BASE Pin Configuration ("Base" portion) Together with IFC_GRP_D_EXT field, This field configures functionality of "Group D" of the IFC pins {IFC_PAR[0:1], IFC_PERR_B, IFC_PAR[2:3]} This field is ignored if RCW_SRC or PBI_SRC or IFC_GRP_A_BASE are configured to QuadSPI, as any of these events configured to QuadSPI forces such pins to QuadSPI operation, rather than whateveris programmed in this field.
5 —	Reserved
4 IFC_GRP_C_BASE	IFC_GRP_C_BASE Pin Configuration ("Base" portion) This field configures functionality of "Group C" of the IFC pins {IFC_A[26:31]} used in 32b data bus pinouts of the IFC. **NOTE:** For SoC's which have IFC_GRP_C_EXT field as valid: IFC_GRP_C_EXT field must be all 0's for IFC_GRP_C_BASE to take effect.
3-2 IFC_GRP_B_BASE	IFC_GRP_B_BASE Pin Configuration ("Base" portion) This field configures functionality of "Group B" of the IFC pins ({IFC_AD[28:31} for 32b data bus pinouts or {IFC_A[28:31]} for 16b data bus pinouts which pinout 32b of addressability). **NOTE:** For SoC's which have IFC_GRP_B_EXT field as valid: IFC_GRP_B_EXT field must be all 0's for IFC_GRP_B_BASE to take effect.
1-0 IFC_GRP_A_BASE	IFC_GRP_A_BASE Pin Configuration ("Base" portion) This field configures functionality of "Group A" of the IFC pins ({IFC_AD[25:27]} for 32b data bus pinouts or {IFC_A[25:27]} for 16b data bus pinouts which pinout 28b or 32b of addressability). This field is ignored if RCW_SRC or PBI_SRC are configured to QSPI, as in such modes the boot source defines the functionality of these pins.

8.3.29 Reset Configuration Word Status Register 16 (RCWSR16)

8.3.29.1 Offset

Register	Offset
RCWSR16	13Ch

8.3.29.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							Reserved	HOST_AGT_PEX3	HOST_AGT_PEX2	HOST_AGT_PEX1	Reserved	Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.29.4 Fields

Field	Function
31-9 —	Reserved
8 —	Reserved
7 HOST_AGT_PEX3	Host/Agent PEX3 Configures Host/Agent mode for PEX 3 Interface if present on an SoC
6 HOST_AGT_PEX2	Host/Agent PEX2 Configures Host/Agent mode for PEX 2 Interface if present on an SoC
5 HOST_AGT_PEX1	Host/Agent PEX1 Configures Host/Agent mode for PEX 1 Interface if present on an SoC
4 —	Reserved
3 —	Reserved
2-0 —	Reserved

8.3.30 Reset Configuration Word Status Register 17 (RCWSR17)

8.3.30.1 Offset

Register	Offset
RCWSR17	140h

8.3.30.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.30.4 Fields

Field	Function
31-0 —	Reserved

8.3.31 Reset Configuration Word Status Register 18 (RCWSR18)

8.3.31.1 Offset

Register	Offset
RCWSR18	144h

8.3.31.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.31.4 Fields

Field	Function
31-0 —	Reserved

8.3.32 Reset Configuration Word Status Register 19 (RCWSR19)

8.3.32.1 Offset

Register	Offset
RCWSR19	148h

8.3.32.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.32.4 Fields

Field	Function
31-0 —	Reserved

8.3.33 Reset Configuration Word Status Register 20 (RCWSR20)

8.3.33.1 Offset

Register	Offset
RCWSR20	14Ch

8.3.33.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.33.4 Fields

Field	Function
31-0 —	Reserved

8.3.34 Reset Configuration Word Status Register 21 (RCWSR21)

8.3.34.1 Offset

Register	Offset
RCWSR21	150h

8.3.34.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.34.4 Fields

Field	Function
31-0 —	Reserved

8.3.35 Reset Configuration Word Status Register 22 (RCWSR22)

8.3.35.1 Offset

Register	Offset
RCWSR22	154h

8.3.35.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.35.4 Fields

Field	Function
31-0 —	Reserved

8.3.36 Reset Configuration Word Status Register 23 (RCWSR23)

8.3.36.1 Offset

Register	Offset
RCWSR23	158h

8.3.36.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.36.4 Fields

Field	Function
31-0 —	Reserved

8.3.37 Reset Configuration Word Status Register 24 (RCWSR24)

8.3.37.1 Offset

Register	Offset
RCWSR24	15Ch

8.3.37.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.37.4 Fields

Field	Function
31-0 —	Reserved

8.3.38 Reset Configuration Word Status Register 25 (RCWSR25)

8.3.38.1 Offset

Register	Offset
RCWSR25	160h

8.3.38.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GP_INFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GP_INFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.38.4 Fields

Field	Function
31-0 GP_INFO	General Purpose Information This data doesn't configure the SoC hardware in any way. It is available to pass information to software.

8.3.39 Reset Configuration Word Status Register 26 (RCWSR26)

8.3.39.1 Offset

Register	Offset
RCWSR26	164h

8.3.39.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.39.4 Fields

Field	Function
31-0 —	Reserved

8.3.40 Reset Configuration Word Status Register 27 (RCWSR27)

8.3.40.1 Offset

Register	Offset
RCWSR27	168h

8.3.40.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		USB2_REFCLK_SEL		USB1_REFCLK_SEL		USB		Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								EM2	EM1	EC2			EC1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.40.4 Fields

Field	Function
31-30 —	Reserved
29-28 USB2_REFCLK_SEL	USB2 Reference Clock Select Selects reference clock for USB2 PHY PLL.
27-26	USB1 Reference Clock Select

Table continues on the next page...

Field	Function
USB1_REFCLK_SEL	Selects reference clock for USB1 PHY PLL.
25-24 USB	USB Pin Configuration Configures the functionality of the USB_DRVBUS and USB_PWRFAULT pins
23-8 —	Reserved
7 EM2	EM2 Pin Configuration Configures the functionality of the External Media Interface2 pins
6 EM1	EM1 Pin Configuration Configures the functionality of the External Media Interface1 pins
5-3 EC2	EC2 Pin Configuration Configures the functionality of the EC2 pins
2-0 EC1	EC1 Pin Configuration Configures the functionality of the EC1 pins

8.3.41 Reset Configuration Word Status Register 28 (RCWSR28)

8.3.41.1 Offset

Register	Offset
RCWSR28	16Ch

8.3.41.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				USB2_CLK_FSEL						USB1_CLK_FSEL					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.41.4 Fields

Field	Function
31-12 —	Reserved
11-6 USB2_CLK_FSEL	USB2 PHY reference clock frequency select. Drives the USB2 PHY fsel[5:0] input. This bit should be set to 0x100111. See USB PHY documentation for details.
5-0 USB1_CLK_FSEL	USB1 PHY reference clock frequency select. Drives the USB1 PHY fsel[5:0] input. This bit should be set to 0x100111. See USB PHY documentation for details.

8.3.42 Reset Configuration Word Status Register 29 (RCWSR29)

8.3.42.1 Offset

Register	Offset
RCWSR29	170h

8.3.42.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SRDS_PRTCL_S1_LN 3				SRDS_PRTCL_S1_LN 2				SRDS_PRTCL_S1_LN 1				SRDS_PRTCL_S1_LN 0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												SRDS_PLL_PD_PLL 4	SRDS_PLL_PD_PLL 3	SRDS_PLL_PD_PLL 2	SRDS_PLL_PD_PLL 1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.42.4 Fields

Field	Function
31-28 SRDS_PRTCL_S1_LN3	SerDes Protocol Select - SerDes 1 Lane3 Proper use of this field is described in "Correct Configuration of Networking SerDes" within the SerDes Solution Chapter.
27-24 SRDS_PRTCL_S1_LN2	SerDes Protocol Select - SerDes 1 Lane2 Proper use of this field is described in "Correct Configuration of Networking SerDes" within the SerDes Solution Chapter.
23-20	SerDes Protocol Select - SerDes 1 Lane1

Table continues on the next page...

Field	Function
SRDS_PRTCL_S1_LN1	Proper use of this field is described in "Correct Configuration of Networking SerDes" within the SerDes Solution Chapter.
19-16 SRDS_PRTCL_S1_LN0	SerDes Protocol Select - SerDes 1 Lane0 Proper use of this field is described in "Correct Configuration of Networking SerDes" within the SerDes Solution Chapter.
15-4 —	Reserved
3 SRDS_PLL_PD_PLL4	SERDES2 PLL 2 Power Down See SRDS_PLL_PD_PLL1
2 SRDS_PLL_PD_PLL3	SERDES2 PLL 1 Power Down See SRDS_PLL_PD_PLL1
1 SRDS_PLL_PD_PLL2	SERDES1 PLL 2 Power Down See SRDS_PLL_PD_PLL1
0 SRDS_PLL_PD_PLL1	SERDES1 PLL 1 Power Down SerDes PLLs are powered up in the POR Reset state titled "SERDES Released from Reset". If this field is set to 0, it informs RESET_REQ logic that it should assert RESET_REQ if a proper reference clock isn't applied or the SERDES PLL does not lock.

8.3.43 Reset Configuration Word Status Register 30 (RCWSR30)

8.3.43.1 Offset

Register	Offset
RCWSR30	174h

8.3.43.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				SRDS_S2_REFCLK_SRC_SEL		SRDS_S1_REFCLK_SRC_SEL		SRDS_DIV_PEX_S2		SRDS_DIV_PEX_S1		SRDS_PLL_REF_CLK_SEL_S2		SRDS_PLL_REF_CLK_SEL_S1	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRDS_PRTCL_S2_LN3				SRDS_PRTCL_S2_LN2				SRDS_PRTCL_S2_LN1				SRDS_PRTCL_S2_LN0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.43.4 Fields

Field	Function
31-28 —	Reserved
27-26 SRDS_S2_REFCLK_SRC_SEL	SERDES2 PLL1,2 Reference Clock Source Select This field selects the reference clock for SerDes2 PLL1 and PLL2. 954: SERDES2, PLL1 955: SERDES2, PLL2
25-24 SRDS_S1_REFCLK_SRC_SEL	SERDES1 PLL1,2 Reference Clock Source Select This field selects the reference clock for SerDes1 PLL1 and PLL2. 952: SERDES1, PLL1 953: SERDES1, PLL2
23-22	SerDes Frequency Divider - SerDes 2

Table continues on the next page...

Field	Function
SRDS_DIV_PEX_S2	This field configures the frequency of PCI Express protocols on this SerDes. Lanes that are supporting other protocols and frequencies are unaffected by this field.
21-20 SRDS_DIV_PEX_S1	SerDes Frequency Divider - SerDes 1 This field configures the frequency of PCI Express protocols on this SerDes. Lanes that are supporting other protocols and frequencies are unaffected by this field.
19-18 SRDS_PLL_REF_CLK_SEL_S2	SerDes PLL Reference Clock Select - SerDes 2 See SRDS_PLL_REF_CLK_SEL_S1.
17-16 SRDS_PLL_REF_CLK_SEL_S1	SerDes PLL Reference Clock Select - SerDes 1 This field selects the PLL reference clock frequency for this SerDes. - RCW[928]: SerDes1, PLL1 - RCW[929]: SerDes1, PLL2 1'b0: 100MHz for protocols operating at 1.25 or 2.5 or 5.0GHz 100MHz for protocols operating at 3.0 or 6.0GHz 125MHz for protocols operating at 3.125 or 6.25GHz 156.25MHz for protocols operating at 10.3125GHz 1'b1: 125MHz for protocols operating at 1.25 or 2.5 or 5.0GHz 125MHz for protocols operating at 3.0 or 6.0GHz 156.25MHz for protocols operating at 3.125 or 6.25GHz
15-12 SRDS_PRTCL_S2_LN3	SerDes Protocol Select - SerDes 2 Lane3 Proper use of this field is described in "Correct Configuration of Non-Networking SerDes" within the SerDes Solution Chapter.
11-8 SRDS_PRTCL_S2_LN2	SerDes Protocol Select - SerDes 2 Lane2 Proper use of this field is described in "Correct Configuration of Non-Networking SerDes" within the SerDes Solution Chapter.
7-4 SRDS_PRTCL_S2_LN1	SerDes Protocol Select - SerDes 2 Lane1 Proper use of this field is described in "Correct Configuration of Non-Networking SerDes" within the SerDes Solution Chapter.
3-0 SRDS_PRTCL_S2_LN0	SerDes Protocol Select - SerDes 2 Lane0 Proper use of this field is described in "Correct Configuration of Non-Networking SerDes" within the SerDes Solution Chapter.

8.3.44 Reset Configuration Word Status Register 31 (RCWSR31)

8.3.44.1 Offset

Register	Offset
RCWSR31	178h

8.3.44.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.44.4 Fields

Field	Function
31-0 —	Reserved

8.3.45 Reset Configuration Word Status Register 32 (RCWSR32)

8.3.45.1 Offset

Register	Offset
RCWSR32	17Ch

8.3.45.2 Function

Contains the Reset Configuration Word (RCW) information written with values read from flash memory by the device at power-on reset and read-only upon exiting reset.

8.3.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.45.4 Fields

Field	Function
31-0 —	Reserved

8.3.46 Scratch Read / Write Register n (SCRATCHRW1 - SCRATCHRW11)

8.3.46.1 Offset

For a = 1 to 11:

Register	Offset
SCRATCHRWa	1FCh + (a × 4h)

8.3.46.2 Function

This register is available for use by the Service Processor as well as the GPP Boot ROM software.

8.3.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.46.4 Fields

Field	Function
31-0 VAL	32-bit scratch content

8.3.47 Scratch Read / Write Register n (SCRATCHRW12)

8.3.47.1 Offset

Register	Offset
SCRATCHRW12	22Ch

8.3.47.2 Function

Used to supplement the CRSTSRn register for reporting core warm resets.

8.3.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.47.4 Fields

Field	Function
31-0 VAL	32-bit scratch content

8.3.48 Scratch Read / Write Register n (SCRATCHRW13 - SCRATCHRW16)

8.3.48.1 Offset

Register	Offset
SCRATCHRW13	230h
SCRATCHRW14	234h
SCRATCHRW15	238h
SCRATCHRW16	23Ch

8.3.48.2 Function

The SCRATCHRWn provides read / write scratch register locations available to the user.

8.3.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.48.4 Fields

Field	Function
31-0 VAL	32-bit scratch content

8.3.49 Scratch Read Register n (SCRATCHW1R1 - SCRATCHW1R4)

8.3.49.1 Offset

Register	Offset
SCRATCHW1R1	300h
SCRATCHW1R2	304h
SCRATCHW1R3	308h
SCRATCHW1R4	30Ch

8.3.49.2 Function

The SCRATCHW1Rn provides scratch register locations available to the user. These are write-once registers. After these have been written once, they can only be written after a power-on or hard reset.

8.3.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.49.4 Fields

Field	Function
31-0 VAL	32-bit scratch content

8.3.50 Boot location pointer low-order address (BOOTLOCPTRL)

8.3.50.1 Offset

Register	Offset
BOOTLOCPTRL	400h

8.3.50.2 Function

Used by Service Processor to pass the boot address to GPPs

8.3.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BOOTLOCPTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BOOTLOCPTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.50.4 Fields

Field	Function
31-0 BOOTLOCPTRL	Low-order boot location pointe

8.3.51 Boot location pointer high-order address (BOOTLOCPTRH)

8.3.51.1 Offset

Register	Offset
BOOTLOCPTRH	404h

8.3.51.2 Function

Used by Service Processor to pass the boot address to GPPs

8.3.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BOOTLOCPTRH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BOOTLOCPTRH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.51.4 Fields

Field	Function
31-0 BOOTLOCPTRH	High-order boot location pointe

8.3.52 USBa_AMQR Register (USB1_AMQR - USB2_AMQR)

8.3.52.1 Offset

Register	Offset
USB1_AMQR	520h
USB2_AMQR	524h

8.3.52.2 Function

USB n Access Management Qualifier register

The AMQR provides the Access Management Qualifier for initiator peripherals on the device which do not have a dedicated internal register for this.

8.3.52.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													PL	BMT	VA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									ICID						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.52.4 Fields

Field	Function
31-19 —	Reserved
18 PL	PL (Privileged Level) bit PL attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction has the correct privelege level for bypassing memory translation and accessing datapath memory resources if desired.
17 BMT	BMT (Bypass Memory Translation) bit

Table continues on the next page...

Field	Function
	BMT attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction desires to bypass memory translation and may desire access to the datapath memory resources. The BMT bit is always be qualified with the PL bit to determine the final action performed.
16 VA	VA (Virtual Address) bit VA attribute that is attached to the outgoing transaction on an appropriate window match for providing additional information about the transaction. This bit indicates that the transaction contains a virtual address.
15-7 —	Reserved
6-0 ICID	Isolation Context 7-bit ICID that is attached to the outgoing transaction on an appropriate window match

8.3.53 SDMM1_AMQR Register (SDMM1_AMQR)

8.3.53.1 Offset

Register	Offset
SDMM1_AMQR	530h

8.3.53.2 Function

SD/MMC Access Management Qualifier register

The AMQR provides the Access Management Qualifier for initiator peripherals on the device which do not have a dedicated internal register for this.

8.3.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													PL	BMT	VA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									ICID						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.53.4 Fields

Field	Function
31-19 —	Reserved
18 PL	PL (Privileged Level) bit PL attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction has the correct privelege level for bypassing memory translation and accessing datapath memory resources if desired.
17 BMT	BMT (Bypass Memory Translation) bit BMT attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction desires to bypass memory translation and may desire access to the datapath memory resources. The BMT bit is always be qualified with the PL bit to determine the final action performed.
16 VA	VA (Virtual Address) bit VA attribute that is attached to the outgoing transaction on an appropriate window match for providing additional information about the transaction. This bit indicates that the transaction contains a virtual address.
15-7 —	Reserved
6-0 ICID	Isolation Context 7-bit ICID that is attached to the outgoing transaction on an appropriate window match

8.3.54 SATA n Access Management Qualifier register (SATA1_AMQR)

8.3.54.1 Offset

Register	Offset
SATA1_AMQR	550h

8.3.54.2 Function

The AMQR provides the Access Management Qualifier for initiator peripherals on the device which do not have a dedicated internal register for this.

8.3.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													PL	BMT	VA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									ICID						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.54.4 Fields

Field	Function
31-19 —	Reserved
18 PL	PL (Privileged Level) bit PL attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction has the correct privelege level for bypassing memory translation and accessing datapath memory resources if desired.
17	BMT (Bypass Memory Translation) bit

Table continues on the next page...

Field	Function
BMT	BMT attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction desires to bypass memory translation and may desire access to the datapath memory resources. The BMT bit is always be qualified with the PL bit to determine the final action performed.
16 VA	VA (Virtual Address) bit VA attribute that is attached to the outgoing transaction on an appropriate window match for providing additional information about the transaction. This bit indicates that the transaction contains a virtual address.
15-7 —	Reserved
6-0 ICID	Isolation Context 7-bit ICID that is attached to the outgoing transaction on an appropriate window match

8.3.55 MISC n Access Management Qualifier register (MISC1_AMQR - MISC3_AMQR)

8.3.55.1 Offset

Register	Offset
MISC1_AMQR	570h
MISC2_AMQR	574h
MISC3_AMQR	578h

8.3.55.2 Function

The AMQR provides the Access Management Qualifier for initiator peripherals on the device which do not have a dedicated internal register for this.

8.3.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													PL	BMT	VA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									ICID						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.55.4 Fields

Field	Function
31-19 —	Reserved
18 PL	PL (Privileged Level) bit PL attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction has the correct privelege level for bypassing memory translation and accessing datapath memory resources if desired.
17 BMT	BMT (Bypass Memory Translation) bit BMT attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction desires to bypass memory translation and may desire access to the datapath memory resources. The BMT bit is always be qualified with the PL bit to determine the final action performed.
16 VA	VA (Virtual Address) bit VA attribute that is attached to the outgoing transaction on an appropriate window match for providing additional information about the transaction. This bit indicates that the transaction contains a virtual address.
15-7 —	Reserved
6-0 ICID	Isolation Context 7-bit ICID that is attached to the outgoing transaction on an appropriate window match

8.3.56 SPARE n Access Management Qualifier register (SPARE1_AMQR - SPARE2_AMQR)

8.3.56.1 Offset

Register	Offset
SPARE1_AMQR	590h
SPARE2_AMQR	594h

8.3.56.2 Function

The AMQR provides the Access Management Qualifier for initiator peripherals on the device which do not have a dedicated internal register for this.

8.3.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													PL	BMT	VA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									ICID						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.56.4 Fields

Field	Function
31-19 —	Reserved
18 PL	PL (Privileged Level) bit PL attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction has the correct privelege level for bypassing memory translation and accessing datapath memory resources if desired.

Table continues on the next page...

Field	Function
17 BMT	BMT (Bypass Memory Translation) bit BMT attribute that is attached to the outgoing transaction on an appropriate window match for steering it to its correct destination in the SoC's address space. This bit indicates that the transaction desires to bypass memory translation and may desire access to the datapath memory resources. The BMT bit is always be qualified with the PL bit to determine the final action performed.
16 VA	VA (Virtual Address) bit VA attribute that is attached to the outgoing transaction on an appropriate window match for providing additional information about the transaction. This bit indicates that the transaction contains a virtual address.
15-7 —	Reserved
6-0 ICID	Isolation Context 7-bit ICID that is attached to the outgoing transaction on an appropriate window match

8.3.57 Topology Initiator Type n Register (TP_ITYP0 - TP_ITYP5)

8.3.57.1 Offset

Register	Offset
TP_ITYP0	740h
TP_ITYP1	744h
TP_ITYP2	748h
TP_ITYP3	74Ch
TP_ITYP4	750h
TP_ITYP5	754h

8.3.57.2 Function

Each Initiator Type Topology Register provides one entry of a 64 entry lookup table.

8.3.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								INIT_TYPE							
W																
Reset	0	0	0	0	0	0	0	0	0[1]	0	0	0	0	0	0	0

1. nnnnnnnn

8.3.57.4 Fields

Field	Function
31-8 —	Reserved
7-0 INIT_TYPE	Initiator Type. This field identifies the type of initiator (core or hardware accelerator) for this index. The lsb differentiates between an enabled and a disabled instance of the initiator. All encodings not listed below are reserved.

8.3.58 Core Cluster n Topology Register (TP_CLUSTER1L - TP_CLUSTER2L)

8.3.58.1 Offset

Register	Offset
TP_CLUSTER1L	844h
TP_CLUSTER2L	84Ch

8.3.58.2 Function

Each Topology Cluster Register (TP_CLUSTERn) contains four 6-bit fields, each of which is an index used for an initiator type lookup in a 64 entry initiator table implemented using the Topology Type Registers.

8.3.58.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EOC		IT_IDX_PC4						0		IT_IDX_PC3					
W																
Reset	0[1]	0	0[2]	0	0	0	0	0	0	0	0[2]	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		IT_IDX_PC2						0		IT_IDX_PC1					
W																
Reset	0	0	0[2]	0	0	0	0	0	0	0	0[2]	0	0	0	0	0

1. nn
2. nnnnnn

8.3.58.4 Fields

Field	Function
31-30 EOC	End of Clusters If the EOC field is non-zero, the register contains the information on the last cluster in the SoC. 2'b00 - not the last cluster 2'b01,10,11 - the last cluster in the SoC
29-24 IT_IDX_PC4	Initiator Type Index for this cluster's fourth initiator Provides a 6-bit index for accessing an entry stored in one of the TP_ITYPn registers. This index selects which of the 64 TP_ITYPn registers contains the identification information for this initiator.
23-22 —	Reserved
21-16 IT_IDX_PC3	Initiator Type Index for this cluster third initiator Provides a 6-bit index for accessing an entry stored in one of the TP_ITYPn registers. This index selects which of the 64 TP_ITYPn registers contains the identification information for this initiator.
15-14 —	Reserved
13-8 IT_IDX_PC2	Initiator Type Index for this cluster second initiator

Table continues on the next page...

Field	Function
	Provides a 6-bit index for accessing an entry stored in one of the TP_ITYPn registers. This index selects which of the 64 TP_ITYPn registers contains the identification information for this initiator.
7-6 —	Reserved
5-0 IT_IDX_PC1	Initiator Type Index for this cluster first initiator Provides a 6-bit index for accessing an entry stored in one of the TP_ITYPn registers. This index selects which of the 64 TP_ITYPn registers contains the identification information for this initiator.

8.3.59 QSPI Control Register (QSPICR1)

8.3.59.1 Offset

Register	Offset
QSPICR1	900h

8.3.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										QSPI_CLK_DIV					
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

8.3.59.3 Fields

Field	Function
31-6 —	Reserved
5-0	QSPI Clock Divisor

Field	Function
QSPI_CLK_DIV	000000-000001b - Reserved 000010b - divide by 8 000011b - divide by 12 000100b - divide by 16 000101b - divide by 20 000110b - divide by 24 000111b - divide by 28 001000b - divide by 32 001001-010011b - Reserved 010100b - divide by 80 010101-111111b - Reserved

8.3.60 I/O Quality of Services Register (IOQOSCR1 - IOQOSCR8)

8.3.60.1 Offset

For a = 1 to 8:

Register	Offset
IOQOSCRa	91Ch + (a × 4h)

8.3.60.2 Function

A set of 8 QoS registers, IOQoSCRn, are provided for I/O sources entering the coherency manager, with one register allocated per coherency manager I/O node. Each register supports three 4-bit fields.

NOTE

IMPLEMENTATION: On the first release, one register is allocated per RN-I.

8.3.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												QOS3			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				QOS2				0				QOS1			
W																
Reset	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1

8.3.60.4 Fields

Field	Function
31-20 —	Reserved
19-16 QOS3	Quality of Service 3
15-12 —	Reserved
11-8 QOS2	Quality of Service 2
7-4 —	Reserved
3-0 QOS1	Quality of Service 1

8.3.61 FlexTimer Control Register (FTMCR)

8.3.61.1 Offset

Register	Offset
FTMCR	960h

8.3.61.2 Function

8.3.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									CASC08_07	CASC07_06	CASC06_05	CASC05_04	CASC04_03	CASC03_02	CASC02_01
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.61.4 Fields

Field	Function
31-7 —	Reserved
6 CASC08_07	Cascade Flex Timer 8 to Flex Timer 7 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)
5 CASC07_06	Cascade Flex Timer 7 to Flex Timer 6 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)
4 CASC06_05	Cascade Flex Timer 6 to Flex Timer 5 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)
3 CASC05_04	Cascade Flex Timer 5 to Flex Timer 4 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)
2 CASC04_03	Cascade Flex Timer 4 to Flex Timer 3 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)

Table continues on the next page...

Field	Function
1 CASC03_02	Cascade Flex Timer 3 to Flex Timer 2 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)
0 CASC02_01	Cascade Flex Timer 2 to Flex Timer 1 0b - These timers are not cascaded together 1b - These timers are cascaded together (higher numbered timer provides the cascaded counter's MS portion)

8.3.62 Core Disable Status Register (COREDISABLESR)

8.3.62.1 Offset

Register	Offset
COREDISABLESR	990h

8.3.62.2 Function

Holds the logical OR of the COREDISR output and the vector of disabled cores based on personality definition.

8.3.62.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.62.4 Fields

Field	Function
31-8 —	Reserved
7 CD7	Core 7 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
6 CD6	Core 6 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
5 CD5	Core 5 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
4 CD4	Core 4 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
3 CD3	Core 3 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
2 CD2	Core 2 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
1 CD1	Core 1 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled
0 CD0	Core 0 Disable 0b - Selected Core is enabled 1b - Selected Core is disabled

8.3.63 Secure Debug configuration register (SDBGCR)

8.3.63.1 Offset

Register	Offset
SDBGCR	A00h

8.3.63.2 Function

SDBGCR is a configuration register supporting debug and factory use of TrustZone capable parts.

NOTE

This mechanism does not open up the Secure world. Instead, it allows access to both secure and non-secure resources if debug or manufacturing test has access through SPIDEN.

8.3.63.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															SDBG_NS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.63.4 Fields

Field	Function
31-1 —	Reserved
0 SDBG_NS	Secure Debug - TZ Not Secure 0b - SAP and Test Port accesses are considered TZ secure accesses when SPIDEN is enabled 1b - SAP and Test Port accesses are not considered TZ secure accesses even when SPIDEN is enabled

8.3.64 IP Block Revision Register 1 (IPBRR1)

8.3.64.1 Offset

Register	Offset
IPBRR1	BF8h

8.3.64.2 Function

IP block revision register.

8.3.64.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IP_MJ								IP_MN							
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

8.3.64.4 Fields

Field	Function
31-16 IP_ID	IP block ID
15-8 IP_MJ	Major revision
7-0 IP_MN	Minor revision

8.3.65 IP Block Revision Register 2 (IPBRR2)

8.3.65.1 Offset

Register	Offset
IPBRR2	BFCh

8.3.65.2 Function

IP block revision register.

8.3.65.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								IP_INT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								IP_CFG							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.65.4 Fields

Field	Function
31-24 —	Reserved
23-16 IP_INT	IP block integration options. Unused.
15-8 —	Reserved
7-0 IP_CFG	IP block configuration options. Unused.

8.4 Pin Control Analog Register Descriptions

8.4.1 Pin Control Analog Memory Map

PINCNTL_ANALOG base address: 0x1FF_8000h

Offset	Register	Width (In bits)	Access	Reset value
14h	Hyst Analog Control 1 (hyst_ana_cntrl1)	32	RW	00000000h
18h	Hyst Analog Control 2 (hyst_ana_cntrl2)	32	RW	00000000h
1Ch	Hyst Analog Control 3 (hyst_ana_cntrl3)	32	RW	00000000h
30h	VSEL Control 1 (vsel_cntrl1)	32	RW	00000000h

8.4.2 Hyst Analog Control 1 (hyst_ana_cntrl1)

8.4.2.1 Offset

Register	Offset
hyst_ana_cntrl1	14h

8.4.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	hyst_irq11	hyst_irq10	hyst_irq9	hyst_irq8	hyst_irq7	hyst_irq6	hyst_irq5	hyst_irq4	hyst_irq3	0					hyst_sdhc_clk	hyst_sdhc_dat3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	hyst_sdhc_dat2	hyst_sdhc_dat1	hyst_sdhc_dat0	hyst_sdhc_cmd	hyst_iic2_sda	hyst_iic2_scl	hyst_iic1_sda	hyst_iic1_scl	hyst_uart2_cts_b	hyst_uart1_cts_b	hyst_uart2_rts_b	hyst_uart1_rts_b	hyst_uart2_sin	hyst_uart1_sin	hyst_uart2_sout	hyst_uart1_sout
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.4.2.3 Fields

Field	Function
31	Hysteresis control for irq11

Table continues on the next page...

Field	Function
hyst_irq11	Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
30 hyst_irq10	Hysteresis control for irq10 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
29 hyst_irq9	Hysteresis control for irq9 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
28 hyst_irq8	Hysteresis control for irq8 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
27 hyst_irq7	Hysteresis control for irq7 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
26 hyst_irq6	Hysteresis control for irq6 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
25 hyst_irq5	Hysteresis control for irq5 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
24 hyst_irq4	Hysteresis control for irq4 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
23 hyst_irq3	Hysteresis control for irq3 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
22-18 —	Reserved.
17 hyst_sdhc_clk	Hysteresis control for sdhc_clk Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
16	Hysteresis control for sdhc_dat3

Table continues on the next page...

Field	Function
hyst_sdhc_dat3	Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
15 hyst_sdhc_dat2	Hysteresis control for sdhc_dat2 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
14 hyst_sdhc_dat1	Hysteresis control for sdhc_dat1 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
13 hyst_sdhc_dat0	Hysteresis control for sdhc_dat0 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
12 hyst_sdhc_cmd	Hysteresis control for sdhc_cmd Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
11 hyst_iic2_sda	Hysteresis control for iic2_sda Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
10 hyst_iic2_scl	Hysteresis control for iic2_scl Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
9 hyst_iic1_sda	Hysteresis control for iic1_sda Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
8 hyst_iic1_scl	Hysteresis control for iic1_scl Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
7 hyst_uart2_cts_b	Hysteresis control for uart2_cts_b Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
6 hyst_uart1_cts_b	Hysteresis control for uart1_cts_b Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input

Table continues on the next page...

Field	Function
5 hyst_uart2_rts_b	Hysteresis control for uart2_rts_b Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
4 hyst_uart1_rts_b	Hysteresis control for uart1_rts_b Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
3 hyst_uart2_sin	Hysteresis control for uart2_sin Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
2 hyst_uart1_sin	Hysteresis control for uart1_sin Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
1 hyst_uart2_sout	Hysteresis control for uart2_sout Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
0 hyst_uart1_sout	Hysteresis control for uart1_sout Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input

8.4.3 Hyst Analog Control 2 (hyst_ana_cntrl2)

8.4.3.1 Offset

Register	Offset
hyst_ana_cntrl2	18h

8.4.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								hyst_usb_pwrfault	hyst_usb_drvvbus	hyst_iic4_sda	hyst_iic4_scl	hyst_iic3_sda	hyst_iic3_scl	hyst_emi_mdio	hyst_emi_mdc
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.4.3.3 Fields

Field	Function
31-8 Reserved	Reserved.
7 hyst_usb_pwrfault	Hysteresis control for usb_pwrfault Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
6 hyst_usb_drvvbus	Hysteresis control for usb_drvvbus Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
5 hyst_iic4_sda	Hysteresis control for iic4_sda Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
4 hyst_iic4_scl	Hysteresis control for iic4_scl Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
3 hyst_iic3_sda	Hysteresis control for iic3_sda Control signal to select Schmitt trigger or CMOS input 0b - CMOS input

Table continues on the next page...

Field	Function
	1b - Schmitt trigger input
2 hyst_iic3_scl	Hysteresis control for iic3_scl Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
1 hyst_emi_mdio	Hysteresis control for emi_mdio Control signal to select Schmitt trigger or CMOS input Includes the following pins: • emi1_mdio • emi2_mdio 0b - CMOS input 1b - Schmitt trigger input
0 hyst_emi_mdc	Hysteresis control for emi_mdc Control signal to select Schmitt trigger or CMOS input Includes the following pins: • emi1_mdc • emi2_mdc 0b - CMOS input 1b - Schmitt trigger input

8.4.4 Hyst Analog Control 3 (hyst_ana_cntrl3)

8.4.4.1 Offset

Register	Offset
hyst_ana_cntrl3	1Ch

8.4.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						hyst_ec2_rx_dv	hyst_ec2_rx_clk	hyst_ec2_rxd3	hyst_ec2_rxd2	hyst_ec2_rxd1	hyst_ec2_rxd0	hyst_ec2_gtx_clk125	hyst_ec2_gtx_clk	hyst_ec2_tx_en	hyst_ec2_txd3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	hyst_ec2_txd2	hyst_ec2_txd1	hyst_ec2_txd0	hyst_ec1_rx_dv	hyst_ec1_rx_clk	hyst_ec1_rxd3	hyst_ec1_rxd2	hyst_ec1_rxd1	hyst_ec1_rxd0	hyst_ec1_gtx_clk125	hyst_ec1_gtx_clk	hyst_ec1_tx_en	hyst_ec1_txd3	hyst_ec1_txd2	hyst_ec1_txd1	hyst_ec1_txd0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.4.4.3 Fields

Field	Function
31-26 —	Reserved.
25 hyst_ec2_rx_dv	Hysteresis control for ec2_rx_dv Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
24 hyst_ec2_rx_clk	Hysteresis control for ec2_rx_clk Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
23 hyst_ec2_rxd3	Hysteresis control for ec2_rxd3 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
22	Hysteresis control for ec2_rxd2

Table continues on the next page...

Field	Function
hyst_ec2_rxd2	Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
21 hyst_ec2_rxd1	Hysteresis control for ec2_rxd1 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
20 hyst_ec2_rxd0	Hysteresis control for ec2_rxd0 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
19 hyst_ec2_gtx_clk125	Hysteresis control for ec2_gtx_clk125 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
18 hyst_ec2_gtx_clk	Hysteresis control for ec2_gtx_clk Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
17 hyst_ec2_tx_en	Hysteresis control for ec2_tx_en Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
16 hyst_ec2_txd3	Hysteresis control for ec2_txd3 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
15 hyst_ec2_txd2	Hysteresis control for ec2_txd2 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
14 hyst_ec2_txd1	Hysteresis control for ec2_txd1 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
13 hyst_ec2_txd0	Hysteresis control for ec2_txd0 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
12 hyst_ec1_rx_dv	Hysteresis control for ec1_rx_dv Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input

Table continues on the next page...

Field	Function
11 hyst_ec1_rx_clk	Hysteresis control for ec1_rx_clk Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
10 hyst_ec1_rxd3	Hysteresis control for ec1_rxd3 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
9 hyst_ec1_rxd2	Hysteresis control for ec1_rxd2 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
8 hyst_ec1_rxd1	Hysteresis control for ec1_rxd1 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
7 hyst_ec1_rxd0	Hysteresis control for ec1_rxd0 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
6 hyst_ec1_gtx_clk125	Hysteresis control for ec1_gtx_clk125 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
5 hyst_ec1_gtx_clk	Hysteresis control for ec1_gtx_clk Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
4 hyst_ec1_tx_en	Hysteresis control for ec1_tx_en Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
3 hyst_ec1_txd3	Hysteresis control for ec1_txd3 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
2 hyst_ec1_txd2	Hysteresis control for ec1_txd2 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input
1 hyst_ec1_txd1	Hysteresis control for ec1_txd1 Control signal to select Schmitt trigger or CMOS input

Table continues on the next page...

Field	Function
	0b - CMOS input 1b - Schmitt trigger input
0 hyst_ec1_txd0	Hysteresis control for ec1_txd0 Control signal to select Schmitt trigger or CMOS input 0b - CMOS input 1b - Schmitt trigger input

8.4.5 VSEL Control 1 (vsel_cntrl1)

8.4.5.1 Offset

Register	Offset
vsel_cntrl1	30h

8.4.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							tvdd_vsel		lvdd_vsel		evdd_vsel		dvdd_vsel		auto_volt_ctrl
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.4.5.3 Fields

Field	Function
31-9 Reserved	Reserved.
8-7 tvdd_vsel	TVDD Voltage Select Dont-care when auto_volt_ctrl is 0

Table continues on the next page...

Field	Function
	00b - 3.3V (don't use) 01b - 2.5V 10b - 2.5V 11b - 1.2V / 1.8V
6-5 lvdd_vsel	LVDD Voltage Select Dont-care when auto_volt_ctrl is 0 00b - 3.3V (don't use) 01b - 2.5V 10b - 2.5V 11b - 1.8V
4-3 evdd_vsel	EVDD Voltage Select Dont-care when auto_volt_ctrl is 0 00b - 3.3V 01b - 2.5V (don't use) 10b - 2.5V (don't use) 11b - 1.8V
2-1 dvdd_vsel	DVDD Voltage Select Dont-care when auto_volt_ctrl is 0 00b - 3.3V 01b - 2.5V (don't use) 10b - 2.5V (don't use) 11b - 1.8V
0 auto_volt_ctrl	Auto-Detect Voltage Control When set, enables dvdd_vsel, evdd_vsel, lvdd_vsel and tvdd_vsel. 0b - Auto-detect mode 1b - Manual voltage select mode

8.5 Pin Control Memory Map/Register Definition

The table below shows the memory-mapped Pin Control CCSR registers and lists the offset, name, and a cross-reference to the complete description of each register. These registers only support 32-bit accesses.

DCFG_PIN_CTRL_PIN_CTRL memory map

Offset address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FF_0060	DDR Clock Disable Register (DCFG_PIN_CTRL_PIN_CTRL_DDRCLKDR)	32	R/W	0000_0000h	8.5.1/540
1FF_0068	IFC Clock Disable Register (DCFG_PIN_CTRL_PIN_CTRL_IFCCLKDR)	32	R/W	0000_0000h	8.5.2/541

8.5.1 DDR Clock Disable Register (DCFG_PIN_CTRL_PIN_CTRL_DDRCLKDR)

DDRCLKDR allows for specific, unused clocks of the DDR Controllers' interface to be released to high impedance, thereby reducing power consumption.

Address: 1FF_0000h base + 60h offset = 1FF_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										D1_MCK0_DIS	D1_MCK1_DIS	D1_MCK2_DIS	D1_MCK3_DIS	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCFG_PIN_CTRL_PIN_CTRL_DDRCLKDR field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21 D1_MCK0_DIS	DDR Controller 1 clock 0 disable. The output is tri-stated, when disabled. 0 D1_MCK[0] is enabled 1 D1_MCK[0] is disabled
20 D1_MCK1_DIS	DDR Controller 1 clock 1 disable. The output is tri-stated, when disabled. 0 D1_MCK[1] is enabled 1 D1_MCK[1] is disabled
19 D1_MCK2_DIS	DDR Controller 1 clock 2 disable. The output is tri-stated, when disabled. 0 D1_MCK[2] is enabled 1 D1_MCK[2] is disabled
18 D1_MCK3_DIS	DDR Controller 1 clock 3 disable. The output is tri-stated, when disabled. 0 D1_MCK[3] is enabled 1 D1_MCK[3] is disabled
17–16 -	This field is reserved. Reserved

Table continues on the next page...

DCFG_PIN_CTRL_PIN_CTRL_DDRCLKDR field descriptions (continued)

Field	Description
15–12 -	Reserved This field is reserved.
-	This field is reserved. Reserved

8.5.2 IFC Clock Disable Register (DCFG_PIN_CTRL_PIN_CTRL_IFCCLKDR)

The IFCCLKDR allows for specific, unused clocks of the IFC interface to be not driven/ high-impedance, thereby reducing power consumption.

Address: 1FF_0000h base + 68h offset = 1FF_0068h

DCFG_PIN_CTRL_PIN_CTRL_IFCCLKDR field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2 IFC_CLK0_DIS	IFC clock 0 disable. The output is not driven and in a high impedance state, when disabled.

Table continues on the next page...

DCFG_PIN_CTRL_PIN_CTRL_IFCCLKDR field descriptions (continued)

Field	Description
	0 IFC_CLK0 is enabled 1 IFC_CLK0 is disabled
1 IFC_CLK1_DIS	IFC clock 1 disable. The output is not driven and in a high impedance state, when disabled. 0 IFC_CLK1 is enabled 1 IFC_CLK1 is disabled
0 -	This field is reserved. Reserved

Chapter 9
Data Path Acceleration Architecture (DPAA2) Overview and Implementation

9.1 LS1088A-Specific DPAA Implementation Details

The *Data Path Acceleration Architecture (DPAA2) Reference Manual* describes the superset of DPAA2 functionality. The LS1088A implements a unique subset of this functionality, as described in the following sections.

9.1.1 Queue Buffer Manager (QBMan) Implementation

The QBMan registers are at block base address: 0x818_0000h and supports following features:

- Two types of portal provided for communication with the QMan
  - Software portals (SWP) used by data path software executing on a processor core. These portals are memory mapped in the system. The number of software portals supported is 5 or 6 portals per GPP core.
  - Direct connect portals (DCP) used by hardware modules with a dedicated interface connected directly to QMan. There is a dedicated DCP portal for WRIOP, AIOP, SEC and qDMA.
- Two-level queueing architecture; frames are enqueued onto frame queues that are in turn enqueued onto WQs. This is referred to as FQ/WQ scheduling mode. An alternate mode referred to as customer edge egress traffic management (CEETM) scheduling mode is available on the egress side of certain direct connect portals. Work queues are grouped into channels with priority scheduling provided between the frame queues on a channel.
- Intra-WQ class scheduling controls the amount of data to be pulled from a frame queue (across one or multiple frames) when it is selected for presentation to a portal.

- SomeWQchannels are dedicated to a particular direct connect portal (DCP), while others are used as software portal channels from which multiple software portals may dequeue.
- Supports customer edge egress traffic management (CEETM) that provides hierarchical class based scheduling and traffic shaping
  - Available instead of FQ/WQ scheduling mode on the egress side of specific direct connect portals
  - Enhanced class based scheduling supporting 16 class queues per channel
  - Token bucket based dual rate shaping representing committed rate (CR) and excess rate (ER)
  - Congestion avoidance mechanism equivalent to that provided by FQ congestion groups
- Supports efficient order restoration for frames from a common source (order definition point) that may be misordered on their way to a destination (order restoration point).
- Supports FQ congestion groups within the QMan where one or more frame queues are configured for a congestion avoidance mechanism, such as RED/WRED.
- Supports tail drop congestion management on a per-frame queue basis. A number of algorithmic sequencers are provided, allowing multiple enqueue/dequeue operations to execute simultaneously to meet the specific performance goals of each SoC.
- The system target interface provides access to the QMan's software portals from processor cores. The system initiator interface is used by the QMan to read and write frame queue management data structures from/to main memory. All data accessed in main memory is private to the QMan, and treated as non-coherent when transferred between QMan and main memory

9.1.2 Advanced Input Output Processor (AIOP) Implementation

The Advanced I/O Processor (AIOP) is a fundamental component of NXP's DPAA2 architecture. The AIOP is a C-programmable engine that enables power efficient, packet-oriented processing by using a single-threaded per flow model; this model allows for the simple use of the AIOP's tightly coupled accelerators.

AIOP is at block base address 0x890_0000. The Advanced I/O Processor User Manual, included in the LS2085A Preliminary SDK Version 4.0, describes the AIOP architecture from the point of view of the AIOP program creator. The Advanced I/O Processor User Manual describes the AIOP service layer and also the many aspects of the hardware architecture (like scheduling) that programmers must know. It also describes how AIOP uses DPAA2 objects.

Pease refer Advanced IO processor (AIOP) for features supported.

9.1.3 Ethernet Wire Rate IO Processor (eWRIOP) Implementation

LS1088A WRIOP global block base address is 0x8B8_0000 and 0x8C0_0000 for physical port space

It supports the following features:

- Support for physical ports and recycled path ports
- Ten physical ports, one management and two recycle ports.
  - Two ports capable of operating at 10/2.5/1 Gbps,100/10 Mbps
  - Eight ports capable of operating at 1 Gbps /100 Mbps/10 Mbps
- Support for configuring physical ports as switch ports
- Up to 10 Ethernet MAC network interfaces
  - MDIO for each Ethernet controller as part of the MAC's access to the internal PCS MII Management Registers
  - Dedicated MDIO for external PHY management
  - Optional MACsec support for four of the eight1Gbps capable MACs
- Provide burst tolerance in system where the speed of the physical links is higher than the average sustained rate of the device
- Split the traffic in a way that allows for 'intelligent' QoS (back pressure and/or discard)
- Detect errors, discard or isolate the frames if needed
- IEEE Std. 802.1Qbb (Priority-based flow control) for up to eight queues/priorities
- 32 WRIOP interface profiles
- WRIOP management commands
- Statistics per port or per interface profile
- Error reporting through QMan and interrupts
- Interfaces with the queue manager (QMan)
  - Dequeues/enqueues frame descriptors (FD) to/from the QMan
  - Supports drop on tail-drop/WRED per the QMan response
  - Supports transmit pause frame due to queue congestion state
- Interfaces with the buffer manager (BMan)
  - Allocates/deallocates frame buffers
- Ethernet MAC features
  - IEEE Std. 802.3ae (full MAC layer and reconciliation sub-layer implementation)
  - Energy efficient Ethernet (IEEE Std. 802.3az)
  - Eight MAC address comparison on receive and one MAC address overwrite on transmit for NIC applications.
  - Ethernet pause frame (802.3 Annex 31A) termination providing fully automated flow control without any user application overhead

    - Priority flow control (PFC) frame support allowing 8 classes for higher layer congestion management
    - Magic packet detection
    - 802.3 basic and mandatory managed objects statistic counters and IETF management
    - Information database (MIB) package (RFC2665) and remote network monitoring (RMON) counters
    - Programmable clause 22 and clause 45 MDIO master interface for PHY device configuration and management
- The Ethernet MAC controller IEEE 1588 timestamp
    - Precise time-stamping of ingress frames and egress frames

9.1.4 Security and Encryption Engine (SEC) Implementation

The Security and Encryption Engine is implemented as described in the *LS1088A Security (SEC) Reference Manual*

The SEC is at block base address 0x800_0000. It supports the following features:

- Offloading of cryptographic functions via a programmable job descriptor language
    - Job descriptors can contain multiple function commands.
    - Job descriptors can be chained to additional job descriptors.
    - Job descriptors are submitted via 4 separate hardware-implemented job rings.
    - Job descriptors can be submitted via a Queue Manager portal.
    - Job descriptors can be submitted via a high-priority AIOP interface (AI).
- Register bus interface
    - 32-bit read/write slave bus
- DMA
    - One 128-bit DMA interfaces
    - Automatic byte, half-word, word and double-word ordering
    - Scatter/gather support for data
- Special-purpose cryptographic keys
    - Black keys
        - Keys stored in memory in encrypted form and decrypted on-the-fly when used
        - AES_ECB or AES_CCM encryption using a 256-bit key
    - Export and import of cryptographic blobs
        - Data encapsulated in a cryptographic data structure for storage in non-volatile memory
        - AES_CCM encryption using a 256-bit key
        - Each blob encrypted using its own randomly generated blob key.

    - Blob key encrypted using a non-volatile blob key encryption key
    - Blob key encryption key derived from non-volatile master key input
- Public key cryptography
  - Modular Arithmetic
    - Addition, subtraction, multiplication, exponentiation, reduction, inversion, greatest common denominator
    - Diffie-Hellman (DH) key agreement
    - DSA and DH key generation
    - Both integer and binary polynomial field functions
    - Modulus size up to 4096 bits
  - Elliptic curve cryptographic functions
    - Point add, point double, point multiply
    - Elliptic curve digital signature algorithm (ECDSA) sign and verify
    - Elliptic curve Diffie-Hellman key agreement
    - ECDSA and ECDH key generation
    - Both prime field and polynomial binary field functions
    - Modulus size up to 1024 bits
  - RSA public and private key exponentiation
    - Modulus size up to 4096 bits
    - Timing equalization to thwart side channel attack
  - Primality testing
    - Maximum size 2048 bits
- Cryptographic authentication
  - Hashing algorithms
    - MD5
    - SHA-1
    - SHA-224
    - SHA-256
    - SHA-384
    - SHA-512
    - SHA-512/224
    - SHA-512/256
  - Message authentication codes (MAC)
    - HMAC-all hashing algorithms
    - SSL 3.0 MAC (SMAC-MD5, SHA-1 only)
    - AES-CMAC
    - AES-XCBC-MAC
    - Kasumi f9
    - SNOW 3G f9
    - ZUC authentication

    - Auto padding
    - ICV checking
- Authenticated encryption algorithms
    - AES-CCM (counter with CBC-MAC)
    - AES-GCM (Galois counter mode)
- Symmetric key block ciphers
    - AES (128-bit, 192-bit or 256-bit keys)
    - DES (64-bit keys, including key parity)
    - 3DES (128-bit or 192-bit keys, including key parity)
    - Kasumi f8 with support for 3g, ECSD/EDGE, and GSM
    - SNOW 3G f8
    - ZUC encryption
    - Cipher modes
        - ECB, CBC, CFB, OFB for all block ciphers
        - CTR and XTS for AES
- Random-number generation
    - Entropy is generated via an independent free running ring oscillator
    - For lower-power consumption, oscillator is off when not generating entropy
    - NIST-compliant, random bit seeded using hardware-generated entropy
- Run-time integrity checking
    - SHA-256 message authentication
    - SHA-512 message authentication
    - Segmented data-gathering to support non-contiguous data blocks in memory
    - Support for up to four independent memory blocks
- Advanced protocol support
    - IPsec
    - SSL/TLS
    - DTLS
    - SRTP
    - IEEE 802.11i WiFi
    - IEEE 802.16 WiMAX
    - IEEE 802.1AE MacSec/LinkSec
    - Support for protocol-specific padding
    - 3gpp Release 11 (LTE) PDCP layer protocol
- Extensive virtualization features
    - Job rings can be time-shared by multiple security domains
    - Black keys are cryptographically separated per security domain
    - Blobs are cryptographically separated per security domain
    - Trusted descriptors are cryptographically separated per security domain

9.1.5 Queue direct memory access (qDMA)

The queue direct memory access (qDMA) controller transfers blocks of data between one source and one or more destinations. The qDMA supports the following features:

- Frame descriptor prefetch
- Channel virtualization
- Transaction based destination stride and source stride
- Destination and source address hold function
- Transaction based cache attribute setup
- Transaction based QoS
- Destination based throttle control
- DMA transfers with misaligned addresses and any transaction size
- Global DMA bandwidth control where transaction generated by the DMA is stalled when the system bandwidth consumption threshold allocated to the DMA has been reached
- Source/destination data scatter/gather through S/G table located in system memory
- Interrupt messaging

Chapter 10
QUICC Engine Block

10.1 Introduction

The *QUICC Engine™ Block Reference Manual with Protocol Interworking* (QEIWRM) describes all the functional units of the QUICC Engine block and must be used in conjunction with this device manual and this chapter.

The QEIWRM is a superset manual which includes some information not relevant to the device. This chapter serves as both a general overview of the QUICC Engine block and a guide to the specific implementation of the QUICC Engine block on the device.

- QUICC Engine block, gives a general overview of the QUICC Engine architecture and communication peripherals.
- QUICC Engine implementation details, lists the chapters that do apply. Implementation-specific details for some chapters follow.

10.2 QUICC Engine block

NOTE

Based on the microcode RAM package that is used, some of the features listed below may not be applicable. See the table below for the different features offered in each microcode configuration.

The QUICC Engine is a versatile communications complex that integrates several communications peripheral controllers. It provides on-chip system design for a variety of applications, particularly in communications and networking systems. The QUICC Engine has the following features:

- 32-bit RISC controller for flexible support of the communications peripherals
- Serial DMA channel for receive and transmit on all serial channels

- Universal communication controllers (UCCs) supporting the following protocols and interfaces (not all of them simultaneously):
  - HDLC and Transparent controllers up to 50 Mbit/s full-duplex; HDLC bus up to 10 Mbit/s
  - UART and asynchronous HDLC
  - BISYNC up to 2 Mbit/s
  - UMCC
- TDM interfaces, with T1/E1/J1 serial interfaces
- 256 channels of HDLC/Transparent per UCC
- RAM-based microcode

The figure below shows the internal architecture and the interfaces provided by the QUICC Engine block on the device. A RAM is used to store parameters for the RISC engine. The instruction RAM is used to optionally run additional code.

Figure 10-1. QUICC Engine block architectural block diagram

10.3 QUICC Engine implementation details

The table below lists the chapters from the *QUICC Engine™ Block Reference Manual with Protocol Interworking* (QEIWRM).

Not all chapters of the QEIWRM apply to this device and those that do apply may have application differences. Use this overview section as a guide for these application differences.

While using the QEIWRM, note this implementation-specific information in general:

- Two UCCs available-UCC1 and UCC3
- Up to two TDM ports
- UMCC

The table below lists the chapters from the *QUICC Engine Block Reference Manual with Protocol Interworking* (QEIWRM) and the chapters with implementation differences.

Table 10-1. QEIWRM chapters and implementation

Chapter Name	Implementation
Part I, "Introduction"	
System Interface	Applies to this device Refer System interface for implementation
Configuration	Applies to this device Refer Configuration for implementation
Multiplexing and Timers	Applies to this device Refer Multiplexing and timers for implementation
Part II, "Unified Communication Controllers (UCCs)"	
Unified Communications Controllers (UCCs)	Applies to this device Refer Unified communications controllers (UCCs) for implementation
UCC for Fast Protocols	Applies to this device Refer UCC for fast protocols for implementation
UCC Ethernet Controller (UEC)	Does not apply to this device
IEEE Standard 1588 Assist	Does not apply to this device
UTOPIA POS Bus Controller (UPC)	Does not apply to this device
ATM Adaptation Layer 2	Does not apply to this device
UCC POS Controller (UPOS)	Does not apply to this device
ATM Controller AAL0, AAL1, and AAL5	Does not apply to this device
HDLC Controller	Applies to this device Refer HDLC controller for implementation
Transparent Controller	Applies to this device Refer Transparent controller for implementation
UCC for Slow Protocols	Applies to this device Refer UCC for slow protocols for implementation
UART Mode and Asynchronous HDLC	Applies to this device Refer UART mode and asynchronous HDLC for implementation
UCC UART PROFIBUS MODE	Does not apply to this device.
Serial Peripheral Interface (SPI)	Does not apply to this device
Universal Serial Bus Controller	Does not apply to this device
BISYNC Mode	Applies to this device Refer BISYNC mode for implementation
Part III, "Time Division Multiplex Support (TDM)" [1]	

Table continues on the next page...

Table 10-1. QEIWRM chapters and implementation (continued)

Chapter Name	Implementation
Serial Interface with Time-Slot Assigner	Applies to this device Refer Serial interface with time-slot assigner for implementation
Multi-Channel Controller (MCC)	Does not apply to this device
Multi-Channel Controller on UCC (UMCC)	Applies to this device Refer Multi-channel controller on UCC (UMCC)
QUICC Multi-Channel Controller (QMC)	Does not apply to this device
Point-to-Point Protocol (PPP)	Does not apply to this device
Serial ATM Microcode	Does not apply to this device
Inverse Multiplexing for ATM (IMA)	Does not apply to this device
ATM AAL1 Circuit Emulation Service	Does not apply to this device
Part IV, "Multiprotocol Interworking"	
Frame Parse and Lookup	Does not apply to this device
Interworking Introduction	Does not apply to this device
Protocol Interworking Programming Model	Does not apply to this device
Virtual Port	Does not apply to this device
IP Reassembly Full	Does not apply to this device
IPv4/UDP Header Compression	Does not apply to this device
IPsec Microcode Package	Does not apply to this device
Part V, "Switching Functionality"	
Enhanced MSP Microcode	Does not apply to this device
L2 Ethernet Switch	Does not apply to this device

1. The TDM can only work with an external sync and external clock; other options are not supported.

The following subsections include device-specific details for the given chapters of the QEIWRM.

10.3.1 System interface

Of the UCCs, only UCC1 and UCC3 are supported on the device.

The following features are not supported: USB, MCC, SPI2, Ethernet, and ATM.

On this device, only a system bus is supported; references to a secondary bus should be disregarded. As such,

- In the "Bus Selection Mechanisms" subsection, references to the secondary bus should be disregarded.
- In the "Serial DMA Status Register (SDSR)" subsection, the BER_2 bit is not available and should be reserved.

- In the "Serial DMA Mode Register (SDMR)" subsection, BER_2_MSK and SBER_2_MSK bits are not available and should be reserved.
- In the "Serial DMA Transfer Address Registers (SDTA1 and SDTA2)" and "Serial DMA Transfer Communication Channel Number Registers (SDTM1 and SDTM2)" subsections, references to the secondary bus should be disregarded.

10.3.1.1 System interface-data paths

In the "Data paths" subsection of the QEIWRM, use this Arm Cortex core information:

The figure below is a simplified block diagram that shows the data paths. The data paths include a path from the SDMA system bus interface to targets connected to the Cache coherent interconnect. The Cache Coherent Interconnect block shown in the figure, is responsible for distribution of I/O masters transactions to the various possible targets (such as the PCI Express or DDR memory controller) based on the transaction's address. The SDMA channel must be configured for big-endian byte ordering for accessing buffer data. The big-endian byte ordering format is programmed in the receive and transmit bus mode registers associated with the peripherals (UMCC). Refer to each protocol of these peripherals for programming of this feature.

Figure 10-2. Data paths

10.3.1.2 System interface-interrupt configuration

In the "Interrupt Configuration" subsection, use the figure below to show the QUICC Engine interrupt structure.

Figure 10-3. QUICC engine module interrupt structure

10.3.1.3 System interface-bus arbitration

The "Bus Arbitration" subsection should be disregarded.

10.3.2 Configuration

Of the UCCs, this device only supports UCC1 and UCC3.

The following features are not supported: USB, MCC, SPI2, Ethernet, and ATM.

10.3.2.1 Configuration-parameter RAM

For the device, replace the "Default Parameter RAM Base Addresses" table with the table below.

Table 10-2. Default parameter RAM base addresses

Address	Peripheral	Size (Bytes)
0x3400	UCC1 (Rx and Tx)	256
0x3600	UCC3 (Rx and Tx)	256
0x3A00	TIMER	64

10.3.3 Multiplexing and timers

Of the UCCs, only UCC1 and UCC3 are supported; of the TDMs, only TDMA and TDMB are supported.

USB, SPI, Ethernet, and ATM are not supported.

The following information applies:

- A bank of 5 external clocks CLK[8-12]
- Two internal clocks CLK[3,15]
- Four supported BRGs: BRG[1-4]
- Two UCCs: UCC1 and UCC3
- Two TDMs: TDMA and TDMB

10.3.3.1 Multiplexing and timers-NMSI configuration

For the device, replace the "Bank of Clocks" figure with the figure below.

Figure 10-4. Bank of clocks

NOTE

1. CLK3 and CLK15 are connected in chip level to the internal platform/2 clock.
2. External CLK12 pin is connected internally to both CLK12 and CLK8 pins of QE.

In the "NMSI Configuration" subsection, refer to the tables below.

The table below shows the clock source options for the serial controllers and TDM channels.

Table 10-3. Clock source options-external clock signals

Clock	External CLK Number						
	3	8	9	10	11	12	15
UCC1 Rx			V	V	V	V	V
UCC1 Tx			V	V	V	V	V
UCC3 Rx			V	V	V	V	V
UCC3 Tx			V	V	V	V	V
TDMA1 Rx	V	V					
TDMA1 Tx			V				
TDMB1 Rx				V			
TDMB1 Tx					V		
QUICC Engine Timer					V	V	

The table below shows the clock routing options using the internal clock generators.

Table 10-4. Clock source options-internal clock

Clock	BRG clock number			
	1	2	3	4
UCC1 Rx	V	V		
UCC1 Tx	V	V		
UCC3 Rx	V	V		
UCC3 Tx	V	V		
TDMA1 Rx[1]			V	V
TDMA1 Tx[1]			V	V
TDMB1 Rx[1]			V	V
TDMB1 Tx[1]			V	V

1. TDM options are used only for internal loopback mode.

The table below shows the UART autobaud clock options.

Table 10-5. Clock source options-UART autobaud clock options

Clock	BRG Clock Number for UART Autobaud			
	1	2	3	4
UCC1 Rx	V			
UCC1 Tx	V			
UCC3 Rx		V		
UCC3 Tx		V		

10.3.3.2 Multiplexing and timers-baud-rate generators (BRGs), BRG configuration registers 1-4 (BRGC*n*)

The table below describes the BRGC*n* fields.

Table 10-6. BRGC*n* field descriptions

Bits	Name	Description
0-13	-	Reserved, should be cleared.
14	RST	Reset BRG. Performs a software reset of the BRG identical to that of an external reset. A reset disables the BRG and drives BRGO high. This is externally visible only if BRGO is connected to the corresponding parallel I/O pin. 0 Enable the BRG. 1 Reset the BRG (software reset).
15	EN	Enable BRG count. Used to dynamically stop the BRG from counting-useful for low-power modes. 0 Stop all clocks to the BRG. 1 Enable clocks to the BRG.
16-17	EXTC	External clock source. Selects the BRG input clock. 00 The BRG input clock comes from the BRGCLK (internal clock generated from the QUICC Engine clock, it is one-half of the QUICC Engine clock). 01 If BRG1, 2: The BRG input clock comes from the CLK3 pin (driven internally by platform/2 clock). If BRG3, 4: The BRG input clock comes from the CLK9 pin. 10 If BRG1, 2: The BRG input clock comes from the CLK3 pin (driven internally by platform/2 clock). If BRG3, 4: The BRG input clock comes from the CLK15 pin (driven internally by platform/2 clock). 11 Reserved
18	ATB	Autobaud. Selects autobaud operation of the BRG on the corresponding RXD. ATB must remain zero until the UCC receives the three Rx clocks. Then the user must set ATB to obtain the correct baud rate. After the baud rate is obtained and locked, it is indicated by setting AB in the UART event register, see "UCC UART Event Register (UCCE) and Mask Register (UCCM)," in *QUICC Engine™ Block Reference Manual with Protocol Interworking*. 0 Normal operation of the BRG. 1 When RXD goes low, the BRG determines the length of the start bit and synchronizes the BRG to the actual baud rate.
19-30	CD	Clock divider. CD presets an internal 12-bit counter that is decremented at the DIV16 output rate. When the counter reaches zero, it is reloaded with CD. CD = 0xFFF produces the minimum clock rate for BGRO (divide by 4,096); CD = 0x000 produces the maximum rate (divide by 1). When dividing by an odd number, the counter ensures a 50% duty cycle by asserting the terminal count; once on clock low and next on clock high. The terminal count signals that the counter has expired and then toggles the clock. 0x000 Divide by 1 (maximum clock rate) 0x001 Divide by 2 ... 0xFFF Divide by 4096 (minimum clock rate)
31	DIV16	Divide-by-16. Selects a divide-by-1 or divide-by-16 prescaler before reaching the clock divider. 0 Divide by 1. 1 Divide by 16.

The table below shows the possible external clock sources for the BRGs.

Table 10-7. BRG external clock source options

BRG	CLK						
	3 [1]	8	9	10	11	12	15 [1]
BRG1	V						
BRG2	V						
BRG3			V				V
BRG4			V				V

1. CLK3 and CLK15 are connected to the internal platform/2 clock.

10.3.4 Unified communications controllers (UCCs)

Of the UCCs, only UCC1 and UCC3 are supported.

The following features are not supported: USB, MCC, SPI, Ethernet, and ATM.

10.3.4.1 Unified communications controllers (UCCs)-UCC page base address

Refer to the table below for the UCC base addresses.

Table 10-8. Parameter RAM-UCC default base addresses

Page	Address [1]	Peripheral	Size (Bytes)
1	0x3400	UCC1	256
3	0x3600	UCC3	256

1. Offset from RAM_Base

10.3.5 HDLC controller

The ratio between the HDLC interface serial clock frequency and the QUICC Engine clock frequency should be at least 1:3.

NOTE

The HDLC serial clock must not exceed the maximal frequency as specified in the data sheet.

10.3.5.1 HDLC controller-introduction

It should be noted that GUMR[RTSM] should be set for the HDLC nibble mode.

10.3.6 Transparent controller

The chip does not support the automatic sync feature, so all references to this should be disregarded.

The QUICC Engine frequency must be higher than the data bit-rate multiplied by 8/3.

The transparent protocol is supported on all UCCs, but octal mode is not supported. UCC1 only supports 1-bit data (serial) mode, while UCC3 supports 1- and 4-bit data modes.

10.3.7 UCC for fast protocols

Only a system bus is supported; references to a secondary bus should be disregarded.

As such,

- In the "Bus Mode Registers (RBMR and TBMR)" subsection of the QEIWRM, DTB and BDB should be reserved.

10.3.8 UCC for slow protocols

Only a system bus is supported; references to a secondary bus should be disregarded.

As such,

- In the "Bus Mode Registers (RBMR and TBMR)" subsection of the QEIWRM, DTB and BDB should be reserved.

10.3.9 UART mode and asynchronous HDLC

The ratio between the UART serial clock frequency and the QUICC Engine clock frequency should be at least 1:16.

10.3.10 BISYNC mode

The ratio between the BISYNC serial clock frequency and the QUICC Engine clock frequency should be at least 1:7.

Generally, the BISYNC serial frequency is less than 20 MHz.

10.3.11 Serial interface with time-slot assigner

The two TDMs, TDMA and TDMB, only support 1-bit data (serial) mode.

The TDM can only work with an external sync and external clock; other options are not supported. The external clock and external sync need to meet the timing specifications as provided in data sheet.

This device supports TDM in a high-speed mode. To run in this mode, the QUICC Engine platform to TDM interface frequency ratio should be 8:1, where a ratio of 400 MHz:50 MHz is recommended. The TDM maximum frequency is 50 MHz.

For the ISDN protocol, the maximum frequency supported is 25 MHz, with the condition that the QUICC Engine platform to ISDN interface frequency ratio is 16:1. For example, if the QUICC Engine platform frequency is 400 MHz, then the ISDN interface maximum frequency is 25 MHz.

10.3.12 Multi-channel controller on UCC (UMCC)

UMCC is supported with only the HDLC and transparent modes. The SS7 mode is not supported.

- The UMCC controller supports two TDM lines.

Chapter 11
Cache Coherent Interconnect

11.1 The CCI-400 module as implemented on the chip

The CCI-400 is the main coherent interconnect of the chip. The CCI-400 interconnect supports two ACE masters, three-Ace-Lite masters, and three slave interfaces.

This section provides details about how the CCI-400 module is implemented on the chip.

11.1.1 LS1088A CCI module integration

The programming model for CCI-400 configuration resides in CCSR at offset 0x409_0000 as defined in the Memory Map chapter.

The CCI-400 supports coherent/snoop transactions (write unique and read once) from the masters. See LS1088A transaction shareability configuration for a description of coherent transaction configuration per master.

Only nERRORIRQ interrupt is connected to the interrupt controller.

NOTE

The SOC does not support lock transactions (generated by LDXR/STXR Arm CPU instructions) for non-cached address regions. Executing these instructions targeting non-cached memory region causes CPU exception.

11.1.2 Connections to the CCI-400 interconnect

The following are the connections to the CCI-400 interconnect:

- 2x full ACE slave ports. One (S3) is associated with A53 Core Cluster 1. The other (S4) is associated with A53 Core Cluster 2.
- 3x ACE-Lite and DVM ports, used to connect to IO-Coherent masters

- S0 Ace-Lite slave interface (PCIe1,2,3, SATA through TBU2; qDMA via TBU3; C-DTC)
- S1 Ace-Lite slave interface (WRIOP, AIOP.FDMA, AIOP.ATSI via TBU1; QM/BM, MC via TBU4; SAP, DAP, eSDHC, USB, uQE, Service Processor via TBU5)
- S2 Ace-Lite slave interface (SEC, AIOP.CTLU via TBU0; GIC, TCU)

Three ACE-Lite master ports

- M0 (connected to Interconnect Fabric through CPE425 for PCI express traffic)
- M1 for peripheral access (all the peripherals except PCI express controllers through Interconnect fabric)
- M2 associated with DDR access

Figure 11-1. Chip interconnect

11.1.3 LS1088A transaction shareability configuration

The following table provides the snoop transaction configurations for different modules:

Table 11-1. Snoop transaction configuration for modules

Module	Snoop transaction configuration
WRIOP	Refer LS1088ADPAARM for details.
BMAN/QMAN	
AIOP	
MC	
SEC	All transactions from SEC are tagged as snoop configuration if the SEC_MCFGR[ARCACHE], and MCFG[AWCACHE] bits are set. Refer Master Configuration (MCFGR) in LS1088A Security Reference Manual for details.
PEX1, 2, 3	Snoop transactions are dependent on PCI Express protocol and controlled by PCI Express IATU registers. Refer SR-IOV Instance (PEX1) register descriptions for details.
qDMA	Global snoop is controlled by DMA mode register, legacy mode - DMA legacy attribute register, queue mode - command queue mode register. Refer LS1088ADPAARM for details.
SATA	All SATA transactions are outer-shareable.
USB3.0	All USB transactions are outer-shareable.
QUICC Engine Lite	Snoop transactions from QUICC Engine are controlled by Bus Mode register. Refer to the Bus Mode register in QEIWRM for details.
eSDHC	eSDHC transactions may be configured as outer-shareable based on the ESDHCCTL[SNOOP] configuration register.
SAP/DAP/SP	SAP, DAP, and SP transactions may be configured as outer-shareable based on SAP, DAP, and SP configuration registers respectively.

11.2 CCI400 register descriptions

11.2.1 CCI400 Registers memory map

CCI400 base address: 409_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control Override Register (Control_Override_Register)	32	RW	Table 11-1
4h	Speculation Control Register (Speculation_Control_Register)	32	RW	Table 11-1
8h	Secure Access Register (Secure_Access_Register)	32	RW	Table 11-1

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	Status Register (Status_Register)	32	RO	Table 11-1
10h	Imprecise Error Register (Imprecise_Error_Register)	32	RW	Table 11-1
1000h	Snoop Control Registers (Snoop_Control_Register_S0)	32	RW	Table 11-1
1004h	Shareable Override Registers (Shareable_Override_Register_S0)	32	RW	Table 11-1
2000h	Snoop Control Registers (Snoop_Control_Register_S1)	32	RW	Table 11-1
2004h	Shareable Override Registers (Shareable_Override_Register_S1)	32	RW	Table 11-1
3000h	Snoop Control Registers (Snoop_Control_Register_S2)	32	RW	Table 11-1
3004h	Shareable Override Registers (Shareable_Override_Register_S2)	32	RW	Table 11-1
4000h	Snoop Control Registers (Snoop_Control_Register_S3)	32	RW	Table 11-1
4004h	Shareable Override Registers (Shareable_Override_Register_S3)	32	RW	Table 11-1
5000h	Snoop Control Registers (Snoop_Control_Register_S4)	32	RW	Table 11-1
5004h	Shareable Override Registers (Shareable_Override_Register_S4)	32	RW	Table 11-1

11.2.2 Control Override Register (Control_Override_Register)

11.2.2.1 Offset

Register	Offset
Control_Override_Register	0h

11.2.2.2 Function

The Control Override register is an additional control register that provides a fail-safe override for some CCI-400 functions, if these cause problems that you cannot otherwise work around. If you cannot avoid using them, only set them using non-bufferable transactions, and before barriers, shareable transactions, or DVM messages are issued into the CCI-400. This can be, for example, very early in the boot sequence, prior to the installation of any Secure OS. You can access the Control Override Register using Secure transactions only, irrespective of the programming of the Secure Access Register . Available in all CCI-400 configurations.

11.2.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										Disable_Retry_Reduction_Buffers	Reserved	Terminate_Barriers	Disable_Speculative_Fetches	DVM_Message_Disable	Snoop_Disable
Reset	u	u	u	u	u	u	u	u	u	u	0	0	0	0	0	0

11.2.2.4 Fields

Field	Function
31-6 —	Reserved
5 Disable_Retry_Reduction_Buffers	Disable retry reduction buffers for speculative fetches 0b - Retry reduction buffers enabled. 1b - Retry reduction buffers disabled.
4 —	Reserved
3 Terminate_Barriers	Terminate Barriers 0b - Master interfaces terminate barriers according to the BARRIERTERMINATE inputs. 1b - All master interfaces terminate barriers.
2 Disable_Speculative_Fetches	Disable Speculative Fetches 0b - Send speculative fetches according to the Speculation Control Register. See Speculation Control Register. 1b - Disable speculative fetches from all master interfaces.
1 DVM_Message_Disable	DVM Message Disable 0b - Send DVM messages according to the Snoop Control Registers. See Snoop Control Registers. 1b - Disable propagation of all DVM messages.

Table continues on the next page...

Field	Function
0 Snoop_Disable	Snoop Disable 0b - Snoop masters according to the Snoop Control Registers. See Snoop Control Registers. 1b - Disable all snoops, but not DVM messages.

11.2.3 Speculation Control Register (Speculation_Control_Register)

11.2.3.1 Offset

Register	Offset
Speculation_Control_Register	4h

11.2.3.2 Function

The Speculation Control register disables speculative fetches for a master interface or for traffic through a specific slave interface. Speculative fetches are not issued if they are disabled in either the slave or master interface for a particular transaction. Access controlled by Secure Access Register, see Secure Access Register. Available in all CCI-400 configurations.

11.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											Disable_Speculative_Fetches_S				
Reset	u	u	u	u	u	u	u	u	u	u	u	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													Disable_Speculative_Fetches_M		
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	0	0	0

11.2.3.4 Fields

Field	Function
31-21 —	Reserved
20-16 Disable_Speculative_Fetches_S	Disable speculative fetches from slave Disable speculative fetches for transactions through a slave interface. One bit for each slave interface: S4, S3, S2, S1, and S0: 00000b - Enable speculative fetches. 00001b - Disable speculative fetches.
15-3 —	Reserved

Table continues on the next page...

Field	Function
2-0 Disable_Speculative_Fetches_M	Disable speculative fetches from master Disable speculative fetches from a master interface. One bit for each master interface: M2, M1, and M0. 000b - Enable speculative fetches. 001b - Disable speculative fetches.

11.2.4 Secure Access Register (Secure_Access_Register)

11.2.4.1 Offset

Register	Offset
Secure_Access_Register	8h

11.2.4.2 Function

The Secure Access register controls secure access. You can only write to this register using Secure transactions. Available in all CCI-400 configurations.

NOTE

This register enables Non-secure access to the CCI-400 registers for all masters. This compromises the security of your system.

11.2.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															Secure_Access_Control
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0

11.2.4.4 Fields

Field	Function
31-1 —	Reserved
0 Secure_Access _Control	Secure Access Control Non-secure register access override: 0b - Disable Non-secure access to CCI-400 registers. 1b - Enable Non-secure access to CCI-400 registers.

11.2.5 Status Register (Status_Register)

11.2.5.1 Offset

Register	Offset
Status_Register	Ch

11.2.5.2 Function

The Status Register safely enables and disables snooping. When changing the snoop or DVM message enables using the Snoop Control Registers, see Snoop Control Registers, there is a delay until the changes are registered in all parts of the CCI-400. The change_pending bit in the Status Register indicates whether there are any changes to the enables that have not yet been applied, or whether a slave interface has been disabled for future snoop and DVM messages, but has outstanding AC requests. There are no usage constraints. Available in all CCI-400 configurations.

NOTE

After writing to the snoop or DVM enable bits, the controller must wait for the register write to complete, then test that the change_pending bit is LOW before it turns an attached device on or off.

11.2.5.3 Diagram

11.2.5.4 Fields

Field	Function
31-1 —	Reserved
0 CCI_Status	CCI_Status Indicates whether any changes to the snoop or DVM enables is pending in the CCI-400 0b - No change pending. 1b - Change pending.

11.2.6 Imprecise Error Register (Imprecise_Error_Register)

11.2.6.1 Offset

Register	Offset
Imprecise_Error_Register	10h

11.2.6.2 Function

The Imprecise Error register records the CCI-400 interfaces that received an error that is not signaled precisely. The appropriate bit is set, with respect to the interface on which the error was received. Bits are set when one or more error responses are detected, and they are reset on a write of 1 to the corresponding bit. Accessible using only Secure accesses, unless you set the Secure Access Register. See Secure Access Register bit assignments. There are no usage constraints. Available in all CCI-400 configurations.

NOTE

If any of the imprecise error indicator bits are set, the nERRORIRQ signal is asserted, active LOW.

11.2.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved											Imp_Err_S4	Imp_Err_S3	Imp_Err_S2	Imp_Err_S1	Imp_Err_S0
Reset	u	u	u	u	u	u	u	u	u	u	u	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved													Imp_Err_M2	Imp_Err_M1	Imp_Err_M0
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	0	0	0

11.2.6.4 Fields

Field	Function
31-21 —	Reserved
20 Imp_Err_S4	Imprecise error indicator for slave interface S4
19 Imp_Err_S3	Imprecise error indicator for slave interface S3
18 Imp_Err_S2	Imprecise error indicator for slave interface S2
17 Imp_Err_S1	Imprecise error indicator for slave interface S1
16 Imp_Err_S0	Imprecise error indicator for slave interface S0
15-3 —	Reserved
2 Imp_Err_M2	Imprecise error indicator for master interface M2
1 Imp_Err_M1	Imprecise error indicator for master interface M1
0 Imp_Err_M0	Imprecise error indicator for master interface M0 0b - No error from the time this bit was last reset. 1b - An error response has been received, but not signalled precisely.

11.2.7 Snoop Control Registers (Snoop_Control_Register_S0 - Snoop_Control_Register_S4)

11.2.7.1 Offset

Register	Offset
Snoop_Control_Register_S0	1000h
Snoop_Control_Register_S1	2000h
Snoop_Control_Register_S2	3000h

Table continues on the next page...

Register	Offset
Snoop_Control_Register_S3	4000h
Snoop_Control_Register_S4	5000h

11.2.7.2 Function

The Snoop Control register controls the issuing of snoop and DVM requests on each slave interface. You can read the register to determine if the interface supports snoops or DVM messages. Enabling snoop or DVM requests on an interface that does not support them has no effect. One Snoop Control Register exists for each slave interface. Accessible using only Secure accesses, unless you set the Secure Access Register. See Secure Access Register bit assignments. Available in all CCI-400 configurations.

NOTE

- If the ACCHANNELEN input is LOW for this interface, write accesses to this register are ignored and snoop or DVM requests cannot be enabled.
- If snoops are disabled in the Control Override Register, write accesses to the snoop enable bit[0] are ignored.
- If DVM messages are disabled in the Control Override Register, write accesses to the DVM enable bit[1] are ignored.

11.2.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Support_DVMs	Support_snoops	Reserved													
W																
Reset	0	0	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														Enable_DVMs	Enable_Snoop
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0	u

11.2.7.4 Fields

Field	Function
31 Support_DVMs	Slave interface supports DVM messages This is overridden to 0x0 if you set the Control Override Register [1]. See Control Override Register.
30 Support_snoops	Slave interface supports snoops This is overridden to 0x0 if you set the Control Override Register [0]. See Control Override Register.
29-2 —	Reserved
1 Enable_DVMs	Enable DVMs Enable issuing of DVM message requests from slave interface. RAZ/WI for interfaces not supporting DVM messages: 0b - Disable DVM message requests. 1b - Enable DVM message requests.
0 Enable_Snoop	Enable Snoop Enable issuing of snoop requests from this slave interface. RAZ/WI for interfaces not supporting snoops: **NOTE:** This bit is reserved for Snoop_Control_Register_S0, Snoop_Control_Register_S1, Snoop_Control_Register_S2, and Snoop_Control_Register_S3. 0b - Disable snoop requests. 1b - Enable snoop requests.

11.2.8 Shareable Override Registers (Shareable_Override_Register_S0 - Shareable_Override_Register_S4)

11.2.8.1 Offset

Register	Offset
Shareable_Override_Register_S0	1004h
Shareable_Override_Register_S1	2004h
Shareable_Override_Register_S2	3004h
Shareable_Override_Register_S3	4004h
Shareable_Override_Register_S4	5004h

11.2.8.2 Function

The Shareable Override register overrides shareability of normal transactions through this interface. The following transaction types are unaffected by any override:

- FIXED-type bursts.
- Device transactions.
- Barrier.
- DVM message transactions.

Usage constraints This register is for ACE-Lite slave interfaces only. See the AMBA AXI and ACE Protocol Specification. Accessible using only Secure accesses, unless you set the Secure Access Register. See Secure Access Register bit assignments. Available in all CCI-400 configurations.

NOTE

Exclusive accesses must not be issued on an interface that is being overridden as shareable. If the CCI-400 is programmed to override transactions as shareable, Exclusive accesses are overridden to normal accesses. An exclusive write then receives

an OKAY response to indicate that the slave does not support exclusive accesses.

11.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														AxDomain_Override	
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0	0

11.2.8.4 Fields

Field	Function
31-2 —	Reserved
1-0 AxDomain_Override	AxDOMAIN override Shareable override for slave interface 00b - Do not override AxDOMAIN inputs. 01b - Do not override AxDOMAIN inputs. 10b - Override AxDOMAIN inputs to 0b00, all transactions are treated as non-shareable.ReadOnce becomes ReadNoSnoop.WriteUnique and WriteLineUnique become WriteNoSnoop. 11b - Override AxDOMAIN inputs to 0b01, normal transactions are treated as shareable.ReadNoSnoop becomes ReadOnce.WriteNoSnoop becomes WriteUnique.

11.3 Functional Description

This chapter describes the functionality of the CoreLink CCI-400 Cache Coherent Interconnect.

11.3.1 About the functions

The CCI-400 combines interconnect and coherency functions into a single module.

11.3.2 Snoop connectivity and control

The CCI-400 has a fully-connected snoop topology, so if they are enabled:

- Each ACE slave interface snoops the other ACE slave interface.
- ACE-Lite slave interfaces snoop both ACE slave interfaces.
- DVM messages are broadcast through all enabled slave interfaces.

Snooping and DVM message broadcast are disabled at reset, so you must enable the appropriate masters for snooping and DVM messages using the Snoop Control Registers before shareable or DVM messages are sent to the CCI-400. See Snoop Control Registers (Snoop_Control_Register_S0 - Snoop_Control_Register_S4).

Each ACE slave interface has programmable bits in the Snoop Control Registers. These bits control the issuing of snoop and DVM message requests on that interface.

NOTE

ACE-Lite slave interfaces have programmable bits to enable DVM messages only.

These programmable bits of the Snoop Control Registers are tied LOW at reset so you must program them HIGH for each master in the shareable domain before shareable transactions or DVM messages are sent to the CCI-400. Before disabling a master, you must disable snoop and DVM messages to that master by programming the relevant Snoop Control Register bits LOW.

To avoid deadlock through having AC requests enabled to interfaces where masters are not present, or not able to process them, the CCI-400 has the following hardware and software override mechanisms:

- Each slave interface has an **ACCHANNELEN** input bit that, if you tie it LOW, prevents that interface from issuing any AC requests.

  NOTE

  These bits are only sampled at reset.

- There are bits in the Control Override Register to disable all snooping or all DVM message broadcast, irrespective of the programming of the Snoop Control Register.

11.3.2.1 Removing a master from the coherent domain

If you want to remove a master from the coherent domain, for example if a processor is being powered down, take the following actions:

1. Stop the processor from issuing shareable transactions. See the documentation of the processor.
2. Clean any shareable data in the processor cache. See the documentation of the processor.
3. Use the Snoop Control Register to prevent any more snoops or DVM messages being sent to the processor. See the Snoop Control Registers (Snoop_Control_Register_S0 - Snoop_Control_Register_S4).
4. Poll the CCI Status Register to confirm that the changes to the Snoop Control Register have completed. See the Status Register (Status_Register).

After you complete these actions, the master is no longer in the coherent domain and you can power it down or disable it. You must enable snoops to that master again before it allocates any shareable data in its cache.

11.3.3 Speculative fetch

For an application where the probability of a miss is high, then the snoop request and response time adds directly to the latency for each transaction labelled as shareable. To mitigate this, you can program each master interface to issue a fetch downstream in parallel with issuing a snoop request. This is known as a *speculative fetch*.

In the event that the snoop associated with a speculative fetch hits in a cache, then the data from the snoop is returned in preference to the data from the speculative fetch.

A speculative fetch is issued before all hazards that had arisen from the corresponding snoop have been resolved. Therefore, it is sometimes necessary to discard the data returned from memory and retry the fetch. These cases are:

- When a hazarding write transaction is detected. This hazarding write transaction must be ordered before the speculative fetch.
- When data from the speculative fetch returns before the snoop response for that transaction, and the read data buffer is already occupied by data waiting for a snoop response.

You can use the PMU to record the number of retry transactions for each master interface.

NOTE

Speculative fetches are only issued for these read-type transactions:

- ReadOnce .
- ReadClean .
- ReadNotSharedDirty .
- ReadUnique .
- ReadShared.

Although speculative fetches reduce the latency in the case of a snoop miss, there is a bandwidth and power penalty because of the additional transactions on a snoop hit. Therefore, you can disable speculative fetches where you expect a significant number of snoops to hit. You can use the Speculation Control Register to disable speculative fetches for a master or a slave interface. For example, you can disable speculative fetches for transactions from a master that is not latency sensitive. See Speculation Control Register (Speculation_Control_Register).

11.3.4 Security

If you are building a system based on the Secure and Non-secure capabilities that Arm TrustZone® technology provides, then you must consider security issues. This section describes:

- Internal programmers view.
- Security of master interfaces.

11.3.4.1 Internal programmers view

With the exception of the PMU registers, the programmers view defaults to Secure access only, as follows:

- Non-secure read requests to Secure registers receive a DECERR response, **RRESP[1:0]** == 0b11 , and zeroed data.
- Non-secure write requests to Secure registers receive a DECERR response, **BRESP[1:0]** == 0b11 and are *Write-Ignored* (WI).

NOTE

Some accesses might receive a response before they reach the CCI-400 registers and so do not receive a DECERR response nor affect the register values. An example of this is a cache maintenance operation that incorrectly addresses the CCI-400 register space.

You can override these security settings in the Secure Access Register. At reset, you can only access this using Secure requests, but if you write to it, this enables Non-secure access to all registers in the CCI-400 except for the Control Override Register and Secure Access Register. See Control Override Register (Control_Override_Register) and Secure Access Register (Secure_Access_Register).

11.3.4.2 Security of master interfaces

Transactions from the CCI-400 master interfaces always retain the security setting of the originating transactions. This applies to:

- Non-shareable transactions.
- Snoop misses.
- Speculative fetches.
- CCI-400-generated writes.

11.3.5 Error responses

The CCI-400 uses a mixture of precise and imprecise signaling of error responses, where:

- Precise errors are signalled back to the master on the R and B channels for the precise transaction that caused the error.
- Imprecise errors are not signalled on R and B channels but are instead signalled using the **nERRORIRQ** output pin (See Table "Interrupt Assignments" in the "Interrupt Assignments" chapter). You can identify the interface that received the error response by reading the Imprecise Error Register. See Imprecise Error Register (Imprecise_Error_Register).

11.3.5.1 Imprecise errors

Table 11-2 shows the errors that are signalled as imprecise. All other sources of error are signalled precisely.

NOTE

An error is signalled either precisely or imprecisely, but never both.

Table 11-2. Imprecise errors

Transaction causing error	Channel receiving error	Imprecise error indicator from
A ReadX snoop that misses in the cache and fetches data from downstream	CR	Slave interface receiving the CR response
Distributed Virtual Memory message	CR	Slave interface receiving the CR response
Speculative fetch that returns an error, but the snoop returns data	R	Master interface receiving the R response
Speculative fetch that must be retried	R	Master interface receiving the R response
Write that the CCI-400 generated	B	Master interface receiving the B response
Snoop error generated by a WriteLineUnique or WriteUnique transaction	CR	Master interface receiving the CR response
Write error for WriteUnique transactions that have been split but not the last transaction in the split sequence	B	Slave interface that received the transaction that was split

The CCI-400 generates a precise DECERR response in the case of a security violation on a CCI-400 register access. See Imprecise Error Register (Imprecise_Error_Register) and Security .

11.3.6 Barriers

The CCI-400 supports all types of AMBA 4 barrier transactions. Each slave interface broadcasts these to every master interface, ensuring that intermediate transaction source and sink points observe the barrier correctly.

11.3.7 DVM messages

The ACE slave interfaces support DVM messages through their regular AC and CR channels. The ACE-Lite interfaces all contain AC and CR channels to support DVM messages only. Each slave interface has a programmable enable bit to determine whether

it supports the issuing of AC requests for DVM messages. DVM messages are handled as regular transactions in the CCI-400, except that they are decoded based on the DVM message indicator, instead of the address, to ensure that multi-transaction DVM messages are correctly ordered.

The Snoop Control Registers and Control Override Register control the issuing of DVM message requests. See Snoop Control Registers (Snoop_Control_Register_S0 - Snoop_Control_Register_S4) and Control Override Register (Control_Override_Register).

ACE and ACE-Lite, plus DVM slave interfaces support all types of DVM transaction. These are:

- DVM Operation.
- DVM Synchronization.
- DVM Complete.

The R channel response to a DVM messages is sent immediately by the CCI-400. If the DVM message results in an error response, this is signaled imprecisely. For more information, see Error responses .

NOTE

- A master that issues DVM messages must also be able to receive DVM messages. The slave interface through which the master connects must have DVM messages enabled.

Chapter 12
DDR Memory Controller

12.1 DDR Introduction

The DDR SDRAM controller controls processor and I/O interactions with system memory. The memory system supports a wide range of memory devices. Built-in error checking and correction (ECC) ensures very low bit-error rates for reliable high-frequency operation. Dynamic power management and auto-precharge modes simplify memory system design. A large set of special features, including ECC error injection, support rapid system debug.

NOTE

In this chapter, the word 'bank' refers to a physical bank specified by a chip select; 'logical bank' refers to one of the four or eight sub-banks in each SDRAM chip. A sub-bank is specified by the 2 or 3 bits on the bank address (MBA) pins during a memory access.

This figure is a high-level block diagram of the DDR memory controller with its associated interfaces.

Figure 12-1. DDR Memory Controller Simplified Block Diagram

12.2 DDR Features

The DDR memory controller includes these distinctive features:

- Support for DDR standards:
  - DDR4

NOTE

The DDR controller is designed to support DDR4 SDRAMs that are compliant with JESD79-4 from September 2012.

- Data bus widths supported:
  - 64-/72-bit
  - 32-/40-bit
- Programmable settings for meeting all SDRAM timing parameters
- The following SDRAM configurations are supported:
  - As many as four physical banks (chip selects), each bank independently addressable

  - 64-Mbit to 16-Gbit devices depending on internal device configuration
  - Supports x8/x16 data ports
  - Unbuffered and registered DIMMs
- Support for data mask signals and read-modify-write for sub-double-word writes. Note that a read-modify-write sequence is only necessary when ECC is enabled.
- Support for double-bit error detection and single-bit error correction ECC (8-bit check word across 64-bit data)
- Support for address parity for registered DIMMs
- Support for address parity for DDR4 discrete memories
- Open page management (dedicated entry for each logical bank)
- Automatic DRAM initialization sequence or software-controlled initialization sequence
- Automatic DRAM data initialization
- Interrupt driven rapid clear of memory
- Write leveling supported
- Support for up to eight posted refreshes
- Memory controller clock frequency of two times the SDRAM clock with support for sleep power management
- Support for error injection

12.2.1 DDR Modes of Operation

The DDR memory controller supports the following modes:

- Dynamic power management mode. The DDR memory controller can reduce power consumption by negating the SDRAM CKE signal when no transactions are pending to the SDRAM.
- Auto-precharge mode. Clearing DDR_SDRAM_INTERVAL[BSTOPRE] causes the memory controller to issue an auto-precharge command with every read or write transaction. Auto-precharge mode can be enabled for separate chip selects by setting CS *n*_CONFIG[AP_ *n*_EN].

12.3 DDR External Signal Descriptions

This section describes external signals of the DDR memory controller. It describes each signal's behavior when the signal is asserted or negated and when the signal is an input or an output.

12.3.1 DDR Signals Overview

Memory controller signals are grouped as follows:

- Memory interface signals
- Clock signals

This table shows how DDR memory controller external signals are grouped. The device data sheet has a pinout diagram showing pin numbers. It also lists all electrical and mechanical specifications.

NOTE

When ECC DRAM is present ECC should be enabled.

Table 12-1. DDR Memory Interface Signal Summary

Name	Function/Description	Reset	Pins	I/O
MALERT_B	Address parity error	One	1	I
MPAR	Address parity out	Zero	1	O
MDQ[0:63]	Data bus	All zeros	64	I/O
MDQS[0:8]	Data strobes	All zeros	9	I/O
MDQS[0:8]_B	Complement data strobes	All ones	9	I/O
MECC[0:7]	Error checking and correcting	All zeros	8	I/O
MCAS_B/MA15	Column address strobe	One	1	O
MA[13:0]	Address bus	All zeros	16	O
MBG[1:0]	Bank group (for DDR4)	All zeros	2	O
MBA[1:0]	Logical bank address	All zeros	3	O
MCS[0:3]_B	Chip selects	All ones	4	O
MWE_B/MA14	Write enable	One	1	O
MRAS_B/MA16	Row address strobe	One	1	O
MACT_B	Activate	One	1	O
MDM[0:8]/MDBI[0:8]	Data mask	All zeros	9	O
MCK[0:1]	DRAM clock outputs NUM_MCK	All zeros	2	O
MCK[0:1]_B	DRAM clock outputs (complement)	All zeros	2	O
MCKE[0:1]	DRAM clock enable	All zeros	2	O
MODT[0:1]	DRAM on-die termination	All zeros	2	O
MDIC[0:1]	Driver impedance calibration	See chip datasheet for more information	2	I/O

This table shows the memory address signal mappings for DDR4.

Table 12-2. Memory Address Signal Mappings for DDR4

Controller Signal Name (Outputs)		DRAM/DIMM Signal Name (Inputs)
msb	MA13	A13
	MA12	A12
	MA11	A11
	MA10	A10 (AP for DDR)
	MA9	A9
	MA8	A8
	MA7	A7
	MA6	A6
	MA5	A5
	MA4	A4
	MA3	A3
	MA2	A2
	MA1	A1
lsb	MA0	A0
	MBG1	BG1
msb	MBG0	BG0
	MBA1	BA1
lsb	MBA0	BA0
	MACT_B	ACT_n
	MRAS_B/MA16	RAS_n/A16
	MCAS_B/MA15	CAS_n/A15
	MWE_B/MA14	WE_n/A14
msb	MCS0_B	CS0_n
	MCS1_B	CS1_n
	MCS2_B	CS2_n
lsb	MCS3_B	CS3_n
	MODT1	ODT1
	MDM[0:8]/MDBI[0:8]	DM_n[0:8]/DBI_n[0:8]
	MALERT_B	ALERT_n
	MAPAR	PAR

12.3.2 DDR Detailed Signal Descriptions

The following sections describe the DDR SDRAM controller input and output signals, the meaning of their different states, and relative timing information for assertion and negation.

12.3.2.1 Memory Interface Signals

This table describes the DDR controller memory interface signals.

Table 12-3. Memory Interface Signals-Detailed Signal Descriptions

Signal	I/O	Description	
MDQ[0:63]	I/O	Data bus. Both input and output signals on the DDR memory controller.	
	O	As outputs for the bidirectional data bus, these signals operate as described below.	
		State Meaning	Asserted/Negated - Represent the value of data being driven by the DDR memory controller.
		Timing	Assertion/Negation - Driven with valid data during writes to memory. High impedance-No READ or WRITE command is in progress; data is not being driven by the memory controller or the DRAM.
	I	As inputs for the bidirectional data bus, these signals operate as described below.	
		State Meaning	Asserted/Negated - Represents the state of data being driven by the external DDR SDRAMs.
		Timing	Assertion/Negation - The DDR SDRAM drives data during a READ transaction. High impedance-No READ or WRITE command in progress; data is not being driven by the memory controller or the DRAM.
MDQS[0:8] MDQS[0:8]_B	I/O	Data strobes. Inputs with read data, outputs with write data. The data strobes must be differential.	
	O	As outputs, the data strobes are driven by the DDR memory controller during a write transaction.	
		State Meaning	Asserted/Negated - Driven high when positive capture data is transmitted and driven low when negative capture data is transmitted. Centered in the data "eye" for writes; coincident with the data eye for reads. Treated as a clock. Data is valid when signals toggle. See DDR SDRAM Interface Operation for byte lane assignments.
		Timing	Assertion/Negation - If a WRITE command is registered at clock edge *n*, data strobes at the DRAM assert centered in the data eye on clock edge *n* + 1. See the JEDEC DDR SDRAM specification for more information.
	I	As inputs, the data strobes are driven by the external DDR SDRAMs during a read transaction. The data strobes are used by the memory controller to synchronize data latching.	
		State Meaning	Asserted/Negated - Driven high when positive capture data is received and driven low when negative capture data is received. Centered in the data eye for writes; coincident with the data eye for reads. Treated as a clock. Data is valid when signals toggle. See DDR SDRAM Interface Operation for byte lane assignments.
		Timing	Assertion/Negation - If a READ command is registered at clock edge *n*, and the latency is programmed in TIMING_CFG_1[CASLAT] to be *m* clocks, data strobes at the DRAM assert coincident with the data on clock edge *n* + *m*. See the JEDEC DDR SDRAM specification for more information.
MECC[0:7]	I/O	Error checking and correcting codes. Input and output signals for the DDR controller's bidirectional ECC bus.	
	O	As normal mode outputs the ECC signals represent the state of ECC driven by the DDR controller on writes.	
		State Meaning	Asserted/Negated - Represents the state of ECC being driven by the DDR controller on writes.
		Timing	Assertion/Negation - Same timing as MDQ High impedance - Same timing as MDQ
	I	As inputs, the ECC signals represent the state of ECC driven by the SDRAM devices on reads.	

Table continues on the next page...

Table 12-3. Memory Interface Signals-Detailed Signal Descriptions (continued)

Signal	I/O	Description	
		State Meaning	Asserted/Negated - Represents the state of ECC being driven by the DDR SDRAMs on reads.
		Timing	Assertion/Negation - Same timing as MDQ High impedance-Same timing as MDQ
MA[13:0]	O	Address bus. Memory controller outputs for the address to the DRAM. MAn carry the address bits for the DDR memory interface comprising the row and column address bits. MA0 is the lsb of the address output from the memory controller.	
		State Meaning	Asserted/Negated - Represents the address driven by the DDR memory controller. Contains different portions of the address depending on the memory size and the DRAM command being issued by the memory controller. See DDR SDRAM Address Multiplexing for a complete description of the mapping of these signals.
		Timing	Assertion/Negation - The address is always driven when the memory controller is enabled. It is valid when a transaction is driven to DRAM (when MCS_B*n* is active). High impedance - When the memory controller is disabled
MBA[1:0]	O	Logical bank address. Outputs that drive the logical (or internal) bank address pins of the SDRAM. Each SDRAM supports four addressable logical sub-banks. Bit zero of the memory controller's output bank address must be connected to bit zero of the SDRAM's input bank address. MBA0, the least-significant bit of the bank address signals, is asserted during the mode register set command to specify the extended mode register.	
		State Meaning	Asserted/Negated - Selects the DDR SDRAM logical (or internal) bank to be activated during the row address phase and selects the SDRAM internal bank for the read or write operation during the column address phase of the memory access. DDR SDRAM Address Multiplexing describes the mapping of these signals in all cases.
		Timing	Assertion/Negation - Same timing as MA*n* High impedance - Same timing as MA*n*
MBG[1:0]	O	Logical bank group. Outputs that drive the logical (or internal) bank group pins of the SDRAM. Each SDRAM supports four addressable logical bank groups. Bit zero of the memory controller's output bank address must be connected to bit zero of the SDRAM's input bank group address.	
		State Meaning	Asserted/Negated-Selects the DDR SDRAM logical (or internal) bank group to be activated during the row address phase and selects the SDRAM internal bank group for the read or write operation during the column address phase of the memory access. DDR SDRAM Address Multiplexing describes the mapping of these signals in all cases.
		Timing	Assertion/Negation-Same timing as MAn High impedance-Same timing as MAn
MACT_B	O	Activate. Active-low SDRAM address multiplexing signal. Asserted for activate commands	
		State Meaning	Asserted-Indicates that a valid SDRAM row address is on the address bus for read and write transactions. Refer to the JEDEC DDR SDRAM specifications for more information on the states required on MACT_B for various other SDRAM commands. Negated-The row address is not guaranteed to be valid.
		Timing	Assertion/Negation - Assertion and negation timing is directed by the values described in DDR SDRAM timing configuration 0 (TIMING_CFG_0) , DDR SDRAM timing configuration 1 (TIMING_CFG_1), DDR SDRAM timing configuration 2 (TIMING_CFG_2), and DDR SDRAM timing configuration 3 (TIMING_CFG_3) High impedance-MACT_B is always driven unless the memory controller is disabled.
MCAS_B/MA15	O	Column address strobe. Active-low SDRAM address multiplexing signal. MCAS_B is asserted for read or write transactions and for mode register set, refresh, self-refrsh and precharge commands.	

Table continues on the next page...

Table 12-3. Memory Interface Signals-Detailed Signal Descriptions (continued)

Signal	I/O	Description	
		This signal is multiplexed with MA15. During activate command assertion, MA15 is driven on this pin; see MAn for details of state meaning and timing. At all other times, the pin is driven as MCAS_B, with state meaning and timing as shown below.	
		State Meaning	Asserted - Indicates that a valid SDRAM column address is on the address bus for read and write transactions. Refer to the JEDEC DDR SDRAM specifications for more information on the states required on MCAS_B for various other SDRAM commands. Negated - The column address is not guaranteed to be valid.
		Timing	Assertion/Negation - Assertion and negation timing is directed by the values described in DDR SDRAM timing configuration 0 (TIMING_CFG_0), DDR SDRAM timing configuration 1 (TIMING_CFG_1), DDR SDRAM timing configuration 2 (TIMING_CFG_2), and DDR SDRAM timing configuration 3 (TIMING_CFG_3) High impedance - MCAS_B is always driven unless the memory controller is disabled.
MRAS_B/MA16	O	Row address strobe. Active-low SDRAM address multiplexing signal. Asserted for activate, refresh, self-refresh and precharge commands. This signal is multiplexed with MA16. During activate command assertion, MA16 is driven on this pin; see MAn for details of state meaning and timing. At all other times, the pin is driven as MRAS_B, with state meaning and timing as shown below.	
		State Meaning	Asserted - Indicates that a valid SDRAM row address is on the address bus for read and write transactions. Refer to the JEDEC DDR SDRAM specifications for more information on the states required on MRAS_B for various other SDRAM commands. Negated - The row address is not guaranteed to be valid.
		Timing	Assertion/Negation - Assertion and negation timing is directed by the values described in DDR SDRAM timing configuration 0 (TIMING_CFG_0) , DDR SDRAM timing configuration 1 (TIMING_CFG_1), DDR SDRAM timing configuration 2 (TIMING_CFG_2), and DDR SDRAM timing configuration 3 (TIMING_CFG_3) High impedance - MRAS_B is always driven unless the memory controller is disabled.
MCS[0:3]_B	O	Chip selects. Four chip selects supported by the memory controller.	
		State Meaning	Asserted - Selects a physical SDRAM bank to perform a memory operation as described in Chip select a memory bounds (CS0_BNDS - CS3_BNDS), and Chip select a configuration (CS0_CONFIG - CS3_CONFIG). The DDR controller asserts one of the MCS_B[0:3]signals to begin a memory cycle. Negated - Indicates no SDRAM action during the current cycle.
		Timing	Assertion/Negation - Asserted to signal any new transaction to the SDRAM. The transaction must adhere to the timing constraints set in TIMING_CFG_0-TIMING_CFG_3. High impedance - Always driven unless the memory controller is disabled.
MWE_B/MA14	O	Write enable. Asserted when a write transaction is issued to the SDRAM. This is also used for mode register set commands and precharge commands. This signal is multiplexed with MA14. During activate command assertion, MA14 is driven on this pin; see MAn for details of state meaning and timing. At all other times, the pin is driven as MWE_B, with state meaning and timing as shown below.	
		State Meaning	Asserted - Indicates a memory write operation. Refer to the JEDEC DDR SDRAM specifications for more information on the states required on MWE_B for various other SDRAM commands. Negated - Indicates a memory read operation.
		Timing	Assertion/Negation - Similar timing as MRAS_B and MCAS_B. Used for write commands. High impedance - MWE_B is always driven unless the memory controller is disabled.

Table continues on the next page...

Table 12-3. Memory Interface Signals-Detailed Signal Descriptions (continued)

Signal	I/O	Description	
MDM[0:8]/ MDBI[0:8]	I/O	These signals are multiplexed in DDR4. They can be either DDR SDRAM data output masks or data bus inversion signals. (See DDR SDRAM control configuration 3 (DDR_SDRAM_CFG_3) for more information.) .As data masks MDMn, these signals mask unwanted bytes of data transferred during a write. They are needed to support sub-burst-size transactions (such as single-byte writes) on SDRAM where all I/O occurs in multi-byte bursts. MDM0 corresponds to the most significant byte (MSB) and MDM7 corresponds to the LSB, while MDM8 corresponds to the ECC byte. DDR SDRAM Interface Operation shows byte lane encodings. As DDR SDRAM data bus inversion signals DBIn_n, these signals indicate that all the data bits of the byte lane are inverted on the data bus during a read or write transaction. When used for this purpose, the signals are I/Os.	
	O	As outputs, these signals operate as described below.	
		State Meaning	Asserted-As masks, prevent writing to DDR SDRAM. Asserted when data is written to DRAM if the corresponding byte(s) should be masked for the write. Note that the MDM*n* signals are active-high for the DDR controller. MDM*n* is part of the DDR command encoding. As DBIs, indicate bus signal inversion per byte lane; asserted by the controller during a write cycle. Note that the DBIn_n signals are active-low for both the DRAM and DDR controller. Negated-As masks, allow the corresponding byte to be read from or written to the SDRAM. As DBIs, signal DRAMs or the DDR controller to invert all DQx in that byte lane before storing data.
		Timing	Assertion/Negation-Same timing as MDQx as outputs. High impedance-Always driven unless the memory controller is disabled.
	I	As inputs, these signals operate as described below.	
		State Meaning	Asserted-As DBIs, indicate bus signal inversion per byte lane; asserted by the DRAMs during a read cycle. Note that the DBIn_n signals are active-low for both the DRAM and DDR controller. Negated-As DBIs, signal DRAMs or the DDR controller to invert all DQx in that byte lane before storing data.
		Timing	Assertion/Negation-Same timing as MDQx as inputs. High impedance-Always driven unless the memory controller is disabled.
MODT[0:1]	O	On-Die termination. Memory controller outputs for the ODT to the DRAM. MODT represents the on-die termination for the associated data, data masks, ECC, and data strobes. The MODT signal should be connected to the same rank of memory as the corresponding MCK/MCK_B pair.	
		State Meaning	Asserted/Negated - Represents the ODT driven by the DDR memory controller.
		Timing	Assertion/Negation - Driven in accordance with JEDEC DRAM specifications for on-die termination timings. It is configured through the CS*n* _CONFIG[ODT_RD_CFG] and CS*n* _CONFIG[ODT_WR_CFG] fields. High impedance - Always driven.
MDIC[0:1]	I/O	Driver impedance calibration. The proper value and connection of the MDIC resistor are specified in the corresponding device data sheetdocument. See DDR Control Driver Register 1 (DDRCDR_1) for more information on these signals.	
		State Meaning	Used for automatic calibration of the DDR IOs
		Timing	Driven for four DRAM cycles at a time while the DDR controller is executing the automatic driver compensation

Table continues on the next page...

Table 12-3. Memory Interface Signals-Detailed Signal Descriptions (continued)

Signal	I/O	Description	
MALERT_B	I	Address parity error. Reflects whether an address parity error has been detected by the DRAM. This signal is active low.	
		State Meaning	Asserted - An error has been detected. Negated - An error has not been detected.
		Timing	Assertion/Negation- Driven by DRAM ALERT_n pin a period of time after parity error has been detected by the DRAM.
MAPAR	O	Address parity out. Driven by the memory controller as the parity bit calculated across the address and command bits. Even parity is used, and parity is not calculated for the MCKE[0:3], MODT[0:3], or MCS_B[0:3] signals. .	
		State Meaning	Asserted-The parity bit is high. Negated-The parity bit is low.
		Timing	Assertion/Negation- Will be issued at the same cycle as the chip select.

12.3.2.2 Clock Interface Signals

This table contains the detailed descriptions of the clock signals of the DDR controller.

Table 12-4. Clock Signals - Detailed Signal Descriptions

Signal	I/O	Description	
MCK[0:1], MCK_B[0:1]	O	DRAM clock output and complement.	
		State Meaning	Asserted/Negated - The JEDEC DDR SDRAM specifications require true and complement clocks. A clock edge is seen by the SDRAM when the true and complement cross.
		Timing	Assertion/Negation - Timing is controlled by the DDR_CLK_CNTL register at offset 0x130.
MCKE[0:1]	O	Clock enable. Output signal used as the clock enable to the SDRAM. MCKEn can be negated to stop clocking the DDR SDRAM. The MCKE signal should be connected to the same rank of memory as the corresponding MCK/MCK_B pair.	
		State Meaning	Asserted - Clocking to the SDRAM is enabled. Negated - Clocking to the SDRAM is disabled and the SDRAM should ignore signal transitions on MCK or MCK_B. MCK/MCK_B are don't cares while MCKEn is negated.
		Timing	Assertion/Negation - Asserted when DDR_SDRAM_CFG[MEM_EN] is set. Can be negated when entering dynamic power management or self refresh. Will be asserted again when exiting dynamic power management or self refresh. High impedance-Always driven.

12.4 DDR register descriptions

This table shows the register memory map for the DDR memory controller.

12.4.1 DDR memory map

DDR base address: 108_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Chip select 0 memory bounds (CS0_BNDS)	32	RW	0000_0000h
8h	Chip select 1 memory bounds (CS1_BNDS)	32	RW	0000_0000h
10h	Chip select 2 memory bounds (CS2_BNDS)	32	RW	0000_0000h
18h	Chip select 3 memory bounds (CS3_BNDS)	32	RW	0000_0000h
80h - 8Ch	Chip select a configuration (CS0_CONFIG - CS3_CONFIG)	32	RW	0000_0000h
100h	DDR SDRAM timing configuration 3 (TIMING_CFG_3)	32	RW	0000_0000h
104h	DDR SDRAM timing configuration 0 (TIMING_CFG_0)	32	RW	0011_0005h
108h	DDR SDRAM timing configuration 1 (TIMING_CFG_1)	32	RW	0000_0000h
10Ch	DDR SDRAM timing configuration 2 (TIMING_CFG_2)	32	RW	0000_0000h
110h	DDR SDRAM control configuration (DDR_SDRAM_CFG)	32	RW	0700_0000h
114h	DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)	32	RW	0000_0000h
118h	DDR SDRAM mode configuration (DDR_SDRAM_MODE)	32	RW	0000_0000h
11Ch	DDR SDRAM mode configuration 2 (DDR_SDRAM_MODE_2)	32	RW	0000_0000h
120h	DDR SDRAM mode control (DDR_SDRAM_MD_CNTL)	32	RW	0000_0000h
124h	DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL)	32	RW	0000_0000h
128h	DDR SDRAM data initialization (DDR_DATA_INIT)	32	RW	0000_0000h
130h	DDR SDRAM clock control (DDR_SDRAM_CLK_CNTL)	32	RW	0200_0000h
148h	DDR training initialization address (DDR_INIT_ADDR)	32	RW	0000_0000h
14Ch	DDR training initialization extended address (DDR_INIT_EXT_ADDRESS)	32	RW	0000_0000h
160h	DDR SDRAM timing configuration 4 (TIMING_CFG_4)	32	RW	0000_0000h
164h	DDR SDRAM timing configuration 5 (TIMING_CFG_5)	32	RW	0000_0000h
16Ch	DDR SDRAM timing configuration 7 (TIMING_CFG_7)	32	RW	0000_0000h
170h	DDR ZQ calibration control (DDR_ZQ_CNTL)	32	RW	0000_0000h
174h	DDR write leveling control (DDR_WRLVL_CNTL)	32	RW	0000_0000h
17Ch	DDR Self Refresh Counter (DDR_SR_CNTR)	32	RW	0000_0000h
180h	DDR Register Control Words 1 (DDR_SDRAM_RCW_1)	32	RW	0000_0000h
184h	DDR Register Control Words 2 (DDR_SDRAM_RCW_2)	32	RW	0000_0000h
190h	DDR write leveling control 2 (DDR_WRLVL_CNTL_2)	32	RW	0000_0000h
194h	DDR write leveling control 3 (DDR_WRLVL_CNTL_3)	32	RW	0000_0000h
1A0h	DDR Register Control Words 3 (DDR_SDRAM_RCW_3)	32	RW	0000_0000h
1A4h	DDR Register Control Words 4 (DDR_SDRAM_RCW_4)	32	RW	0000_0000h
1A8h	DDR Register Control Words 5 (DDR_SDRAM_RCW_5)	32	RW	0000_0000h
1ACh	DDR Register Control Words 6 (DDR_SDRAM_RCW_6)	32	RW	0000_0000h
200h	DDR SDRAM mode configuration 3 (DDR_SDRAM_MODE_3)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
204h	DDR SDRAM mode configuration 4 (DDR_SDRAM_MODE_4)	32	RW	0000_0000h
208h	DDR SDRAM mode configuration 5 (DDR_SDRAM_MODE_5)	32	RW	0000_0000h
20Ch	DDR SDRAM mode configuration 6 (DDR_SDRAM_MODE_6)	32	RW	0000_0000h
210h	DDR SDRAM mode configuration 7 (DDR_SDRAM_MODE_7)	32	RW	0000_0000h
214h	DDR SDRAM mode configuration 8 (DDR_SDRAM_MODE_8)	32	RW	0000_0000h
220h	DDR SDRAM mode configuration 9 (DDR_SDRAM_MODE_9)	32	RW	0000_0000h
224h	DDR SDRAM mode configuration 10 (DDR_SDRAM_MODE_10)	32	RW	0000_0000h
228h	DDR SDRAM mode configuration 11 (DDR_SDRAM_MODE_11)	32	RW	0000_0000h
22Ch	DDR SDRAM mode configuration 12 (DDR_SDRAM_MODE_12)	32	RW	0000_0000h
230h	DDR SDRAM mode configuration 13 (DDR_SDRAM_MODE_13)	32	RW	0000_0000h
234h	DDR SDRAM mode configuration 14 (DDR_SDRAM_MODE_14)	32	RW	0000_0000h
238h	DDR SDRAM mode configuration 15 (DDR_SDRAM_MODE_15)	32	RW	0000_0000h
23Ch	DDR SDRAM mode configuration 16 (DDR_SDRAM_MODE_16)	32	RW	0000_0000h
250h	DDR SDRAM timing configuration 8 (TIMING_CFG_8)	32	RW	0000_0000h
260h	DDR SDRAM control configuration 3 (DDR_SDRAM_CFG_3)	32	RW	0000_0000h
400h	DQ mapping register 0 (DDR_DQ_MAP0)	32	RW	0000_0000h
404h	DQ mapping register 1 (DDR_DQ_MAP1)	32	RW	0000_0000h
408h	DQ mapping register 2 (DDR_DQ_MAP2)	32	RW	0000_0000h
40Ch	DQ mapping register 3 (DDR_DQ_MAP3)	32	RW	0000_0000h
B20h	DDR Debug Status Register 1 (DDRDSR_1)	32	RO	0000_8080h
B24h	DDR Debug Status Register 2 (DDRDSR_2)	32	RW	8000_0000h
B28h	DDR Control Driver Register 1 (DDRCDR_1)	32	RW	0000_0000h
B2Ch	DDR Control Driver Register 2 (DDRCDR_2)	32	RW	0000_0000h
BF8h	DDR IP block revision 1 (DDR_IP_REV1)	32	RO	0002_0502h
BFCh	DDR IP block revision 2 (DDR_IP_REV2)	32	RO	0000_0000h
D00h	DDR Memory Test Control Register (DDR_MTCR)	32	RW	0000_0000h
D20h - D44h	DDR Memory Test Pattern n Register (DDR_MTP0 - DDR_MTP9)	32	RW	0000_0000h
D60h	DDR Memory Test Start Extended Address (DDR_MT_ST_EXT_ADDR)	32	RW	0000_0000h
D64h	DDR Memory Test Start Address (DDR_MT_ST_ADDR)	32	RW	0000_0000h
D68h	DDR Memory Test End Extended Address (DDR_MT_END_EXT_ADDR)	32	RW	0000_0000h
D6Ch	DDR Memory Test End Address (DDR_MT_END_ADDR)	32	RW	0000_0000h
E00h	Memory data path error injection mask high (DATA_ERR_INJECT_HI)	32	RW	0000_0000h
E04h	Memory data path error injection mask low (DATA_ERR_INJECT_LO)	32	RW	0000_0000h
E08h	Memory data path error injection mask ECC (ECC_ERR_INJECT)	32	RW	0000_0000h
E20h	Memory data path read capture high (CAPTURE_DATA_HI)	32	RW	0000_0000h
E24h	Memory data path read capture low (CAPTURE_DATA_LO)	32	RW	0000_0000h
E28h	Memory data path read capture ECC (CAPTURE_ECC)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
E40h	Memory error detect (ERR_DETECT)	32	W1C	0000_0000h
E44h	Memory error disable (ERR_DISABLE)	32	RW	0000_0000h
E48h	Memory error interrupt enable (ERR_INT_EN)	32	RW	0000_0000h
E4Ch	Memory error attributes capture (CAPTURE_ATTRIBUTES)	32	RW	0000_0000h
E50h	Memory error address capture (CAPTURE_ADDRESS)	32	RW	0000_0000h
E54h	Memory error extended address capture (CAPTURE_EXT_ADDRESS)	32	RW	0000_0000h
E58h	Single-Bit ECC memory error management (ERR_SBE)	32	RW	0000_0000h

12.4.2 Chip select a memory bounds (CS0_BNDS - CS3_BNDS)

12.4.2.1 Offset

Register	Offset
CS0_BNDS	0h
CS1_BNDS	8h
CS2_BNDS	10h
CS3_BNDS	18h

12.4.2.2 Function

The chip select bounds registers (CS*n* _BNDS) define the starting and ending address of the memory space that corresponds to the individual chip selects. Note that the size specified in CS*n* _BNDS should equal the size of physical DRAM. Also, note that EA*n* must be greater than or equal to SA*n*.

If chip select interleaving is enabled, all fields in the lower interleaved chip select will be used, and the other chip selects' bounds registers will be unused. For example, if chip selects 0 and 1 are interleaved, all fields in CS0_BNDS will be used, and all fields in CS1_BNDS will be unused.

12.4.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.2.4 Fields

Field	Function
31-16 SA	Starting Address. Starting address for chip select (bank) *n*.This value is compared against the 16 msbs of the 40-bit address.
15-0 EA	Ending Address. Ending address for chip select (bank)*n*. This value is compared against the 16 msbs of the 40-bit address.

12.4.3 Chip select a configuration (CS0_CONFIG - CS3_CONFIG)

12.4.3.1 Offset

Register	Offset
CS0_CONFIG	80h
CS1_CONFIG	84h
CS2_CONFIG	88h
CS3_CONFIG	8Ch

12.4.3.2 Function

The chip select configuration (CS*n* _CONFIG) registers enable the DDR chip selects and set the number of row and column bits used for each chip select. These registers should be loaded with the correct number of row and column bits for each SDRAM. Because CS*n* _CONFIG[ROW_BITS_CS_*n* , COL_BITS_CS_*n*] establish address multiplexing, the user should take great care to set these values correctly.

If chip select interleaving is enabled, then all fields in the lower interleaved chip select will be used, and the other registers' fields will be unused, with the exception of the ODT_RD_CFG and ODT_WR_CFG fields. For example, if chip selects 0 and 1 are interleaved, all fields in CS0_CONFIG will be used, but only the ODT_RD_CFG and ODT_WR_CFG fields in CS1_CONFIG will be used.

12.4.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CS_EN	Reserved							AP_EN	ODT_RD_CFG			Reserved	ODT_WR_CFG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BA_BITS_CS		Reserved			ROW_BITS_CS			Reserved		BG_BITS_CS		Reserved	COL_BITS_CS		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.3.4 Fields

Field	Function
31 CS_EN	Chip Select Enable. Chip select *n*enable 0b - Chip select *n*is not active 1b - Chip select *n*is active and assumes the state set in CS*n* _BNDS.
30-24	Reserved

Table continues on the next page...

Field	Function
—	
23 AP_EN	Auto Precharge Enable. Chip select *n*auto-precharge enable 0b - Chip select will only be auto-precharged if global auto-precharge mode is enabled (DDR_SDRAM_INTERVAL[BSTOPRE] = 0). 1b - Chip select will always issue an auto-precharge for read and write transactions.
22-20 ODT_RD_CFG	On-Die Termination Read Config. ODT for reads configuration. Note that CAS latency plus additive latency must be at least 3 cycles for ODT_RD_CFG to be enabled. 000b - Never assert ODT for reads 001b - Assert ODT only during reads to CS*a* 010b - Assert ODT only during reads to other chip selects 011b - Assert ODT only during reads to other DIMM modules. It is assumed that CS0 and CS1 are on the same DIMM module, whereas CS2 and CS3 are on a separate DIMM module. 100b - Assert ODT for all reads 101b - Assert ODT only during transactions to same DIMM 110b - Assert ODT only during transactions to own CS and other DIMM. 111b - Assert ODT only during transactions to other CS in same DIMM.
19 —	Reserved
18-16 ODT_WR_CFG	On-Die Termination Write Config. ODT for writes configuration. Note that write latency plus additive latency must be at least 3 cycles for ODT _WR_CFG to be enabled. 000b - Never assert ODT for writes 001b - Assert ODT only during writes to CSn 010b - Assert ODT only during writes to other chip selects 011b - Assert ODT only during writes to other DIMM modules. It is assumed that CS0 and CS1 are on the same DIMM module, whereas CS2 and CS3 are on a separate DIMM module. 100b - Assert ODT for all writes 101b - Assert ODT only during transactions to same DIMM 110b - Assert ODT only during transactions to own CS and other DIMM. 111b - Assert ODT only during transactions to other CS in same DIMM.
15-14 BA_BITS_CS	Bank Address Bits. Number of bank bits for SDRAM on chip select*n*. These bits correspond to the sub-bank bits driven on MBA*n*. Note that if DDR4 is used, this must be set to 00, as 8 sub-banks are not supported when also using bank groups. 00b - 2 logical bank bits 01b - Reserved 10b - Reserved 11b - Reserved
13-11 —	Reserved
10-8 ROW_BITS_CS	Row Bits. Number of row bits for SDRAM on chip select*n*. 000b - 12 row bits 001b - 13 row bits 010b - 14 row bits 011b - 15 row bits

Table continues on the next page...

Field	Function
	100b - 16 row bits 101b - 17 row bits 110b - 18 row bits 111b - Reserved
7-6 —	Reserved
5-4 BG_BITS_CS	Bank Group Bits. Number of bank group bits for SDRAM on chip select*n*. In addition, it is illegal to use BA_BITS_CS_*n* equal to 01 if DDR4 bank groups are used. 00b - 0 bank group bits 01b - 1 bank group bit 10b - 2 bank group bits 11b - Reserved
3 —	Reserved
2-0 COL_BITS_CS	Column Bits. Number of column bits for SDRAM on chip select*n.* The decoding is as follows: 000b - 8 column bits 001b - 9 column bits 010b - 10 column bits 011b - 11 column bits 100b - Reserved 101b - Reserved 110b - Reserved 111b - Reserved

12.4.4 DDR SDRAM timing configuration 3 (TIMING_CFG_3)

12.4.4.1 Offset

Register	Offset
TIMING_CFG_3	100h

12.4.4.2 Function

DDR SDRAM timing configuration register 3 sets the extended refresh recovery time, which is combined with TIMING_CFG_1[REFREC] to determine the full refresh recovery time.

12.4.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			EXT_PRETOACT	Reserved		EXT_ACTTOPRE		Reserved	EXT_ACTTORW	EXT_REFREC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		EXT_CASLAT		Reserved	EXT_ADD_LAT	Reserved	EXT_WRREC	Reserved					CNTL_ADJ		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.4.4 Fields

Field	Function
31-29 —	Reserved
28 EXT_PRETOACT	Extended Precharge to Activate. Extended precharge-to-activate interval (t_{RP}). Determines the number of clock cycles from a precharge command until an activate or refresh command is allowed. This field is concatenated with TIMING_CFG_1[PRETOACT] to obtain a 5-bit value for the total precharge to activate time. 0b - 0 clocks 1b - 16 clocks
27-26 —	Reserved
25-24 EXT_ACTTOPRE	Extended Activate to Precharge. Extended Activate to precharge interval (t_{RAS}). Determines the number of clock cycles from an activate command until a precharge command is allowed. This field is concatenated with TIMING_CFG_1[ACTTOPRE] to obtain a 6-bit value for the total activate to precharge. Note that a 6-bit value of 00_0000 is the same as a 6-bit value of 01_0000. Both values represent 16 cycles. 00b - 0 clocks 01b - 16 clocks 10b - 32 clocks 11b - 48 clocks
23 —	Reserved

Table continues on the next page...

Field	Function
22 EXT_ACTTORW	Extended Activate to Read/Write. Extended activate to read/write interval for SDRAM (t_{RCD}). Controls the number of clock cycles from an activate command until a read or write command is allowed. This field is concatenated with TIMING_CFG_1[ACTTORW] to obtain a 5-bit value for the total activate to read/write time.
21-16 EXT_REFREC	Extended Refresh Recovery. Extended refresh recovery time (t_{RFC}). Controls the number of clock cycles from a refresh command until an activate command is allowed. This field is concatenated with TIMING_CFG_1[REFREC] to obtain a 10-bit value for the total refresh recovery. Note that hardware adds an additional 8 clock cycles to the final, 10-bit value of the refresh recovery, such that t_{RFC} is calculated as follows: t_{RFC} = {EXT_REFREC || REFREC} + 8. (Settings greater than 101111b are reserved.) All values of less than 0b110000 are legal; in this case the number of cycles is [setting] × 16. Sample values are shown below. Settings of 0b110000 or greater are reserved. 000000b - 0 clocks 000001b - 16 clocks 000010b - 32 clocks 101110b - 736 clocks 101111b - 752 clocks 110000-111111b - Reserved
15-14 —	Reserved
13-12 EXT_CASLAT	Extended CAS Latency. Number of clock cycles between registration of a READ command by the SDRAM and the availability of the first output data. If a READ command is registered at clock edge *n* and the latency is *m* clocks, data is available nominally coincident with clock edge *n* + *m*. This field is concatenated with TIMING_CFG_1[CASLAT] to obtain a 5-bit value for the total CAS latency. Note that the value of this field is added to the programmed value in TIMING_CFG_1[CASLAT]. The largest total CAS latency supported is 20 clocks. 00b - 0 clocks 01b - 8 clocks 10b - 16 clocks 11b - Reserved
11 —	Reserved
10 EXT_ADD_LAT	Extended Additive Latency. The additive latency must be set to a value less than TIMING_CFG_1[ACTTORW]. Note that the value of this field is added to the programmed value in TIMING_CFG_2[ADD_LAT]. The largest total additive latency supported is 19 clocks. 0b - 0 clocks 1b - 16 clocks
9 —	Reserved
8 EXT_WRREC	Extended Write Recovery. Extended last data to precharge minimum interval (t_{WR}). Determines the number of clock cycles from the last data associated with a write command until a precharge command is allowed. If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this field needs to be programmed to (t_{WR} + 2 cycles). This field is concatenated with TIMING_CFG_1[WRREC] to obtain a 5-bit value for the total write recovery time. 0b - 0 clocks

Table continues on the next page...

Field	Function
	1b - 16 clocks
7-3 —	Reserved
2-0 CNTL_ADJ	Control Adjust. Controls the amount of delay to add to the lightly loaded control signals with respect to all other DRAM address and command signals. The signals affected by this field are MODTn, MCSn_B, and MCKEn. 000b - MODTn, MCSn_B, and MCKEn will be launched aligned with the other DRAM address and control signals. 001b - MODTn, MCSn_B, and MCKEn will be launched 1/4 DDR clock cycle later than the other DRAM address and control signals. 010b - MODTn, MCSn_B, and MCKEn will be launched 1/2 DDR clock cycle later than the other DRAM address and control signals. 011b - MODTn, MCSn_B, and MCKEn will be launched 3/4 DDR clock cycle later than the other DRAM address and control signals. 100b - MODTn, MCSn_B, and MCKEn will be launched 1 DDR clock cycle later than the other DRAM address and control signals. 101b - MODTn, MCSn_B, and MCKEn will be launched 5/4 DDR clock cycles later than the other DRAM address and control signals. 110b - Reserved 111b - Reserved

12.4.5 DDR SDRAM timing configuration 0 (TIMING_CFG_0)

12.4.5.1 Offset

Register	Offset
TIMING_CFG_0	104h

12.4.5.2 Function

DDR SDRAM timing configuration register 0 sets the number of clock cycles between various SDRAM control commands.

12.4.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RWT		WRT		RRT		WWT		ACT_PD_EXIT				PRE_PD_EXIT			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXT_PRE_PD_EXIT		Reserved									MRS_CYC				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

12.4.5.4 Fields

Field	Function
31-30 RWT	Read-to-write Turnaround. Specifies how many extra cycles will be added between a read to write turnaround. If 0 clocks is chosen, then the DDR controller will use a fixed number based on the CAS latency and write latency. Choosing a value other than 0 adds extra cycles past this default calculation. As a default the DDR controller will determine the read-to-write turnaround as CL - WL + BL/2 + 2. In this equation, CL is the CAS latency rounded up to the next integer, WL is the programmed write latency, and BL is the burst length. This field is concatenated with TIMING_CFG_4[EXT_RWT] to obtain a 4-bit value for the total read-to-write turnaround 00b - 0 clocks 01b - 1 clock 10b - 2 clocks 11b - 3 clocks
29-28 WRT	Write-to-read Turnaround. Specifies how many extra cycles will be added between a write to read turnaround. If 0 clocks is chosen, then the DDR controller will use a fixed number based on the, read latency, and write latency. Choosing a value other than 0 adds extra cycles past this default calculation. As a default, the DDR controller will determine the write-to-read turnaround as WL - CL + BL/2 + 1. In this equation, CL is the CAS latency rounded down to the next integer, WL is the programmed write latency, and BL is the burst length. This field is concatenated with TIMING_CFG_4[EXT_WRT] to obtain a 3-bit value for the total write-to-read turnaround 00b - 0 clocks 01b - 1 clock 10b - 2 clocks 11b - 3 clocks
27-26 RRT	Read-to-read Turnaround.

Table continues on the next page...

Field	Function
	Specifies how many extra cycles will be added between reads to different chip selects. As a default, 3 cycles will be required between read commands to different chip selects. Extra cycles may be added with this field. Note: If 8-beat bursts are enabled, then 5 cycles will be the default. This field is concatenated with TIMING_CFG_4[EXT_RRT] to obtain a 3-bit value for the total read-to-read turnaround. 00b - 0 clocks 01b - 1 clock 10b - 2 clocks 11b - 3 clocks
25-24 WWT	Write-to-write Turnaround. Specifies how many extra cycles will be added between writes to different chip selects. As a default, 2 cycles will be required between write commands to different chip selects. Extra cycles may be added with this field. Note: If 8-beat bursts are enabled, then 4 cycles will be the default. This field is concatenated with TIMING_CFG_4[EXT_WWT] to obtain a 3-bit value for the total write-to-write turnaround 00b - 0 clocks 01b - 1 clock 10b - 2 clocks 11b - 3 clocks
23-20 ACT_PD_EXIT	Active Powerdown Exit. Active powerdown exit timing (t_{XP}). Specifies how many clock cycles to wait after exiting active powerdown before issuing any command. 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
19-16 PRE_PD_EXIT	Precharge Powerdown Exit. Precharge powerdown exit timing (t_{XP}). Specifies how many clock cycles to wait after exiting precharge powerdown before issuing any command. This field is concatenated with TIMING_CFG_0[EXT_PRE_PD_EXIT] to obtain a 6-bit value for the total precharge powerdown exit timing. 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks

Table continues on the next page...

Field	Function
	1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
15-14 EXT_PRE_PD_EXIT	Extended Precharge Powerdown Exit. Extended precharge powerdown exit timing (t_{XP}). Specifies how many clock cycles to wait after exiting precharge powerdown before issuing any command. Note the decoding for this field is not a straight decode. This field is concatenated with TIMING_CFG_0[PRE_PD_EXIT] to obtain a 6-bit value for the total precharge powerdown exit timing. 00b - 0 clocks 01b - 16 clocks 10b - 32 clocks 11b - 48 clocks
13-5 —	
4-0 MRS_CYC	Mode Register Set Cycle Time. Mode register set cycle time (t_{MRD}, t_{MOD}). Specifies the number of cycles that must pass after a Mode Register Set command until any other command. This should be set to the greater of $t_{MRD\ and\ tMOD}$). If command/address latency (CAL) mode is used for DDR4, then this field should be set to the greater of $t_{MRD_CAL\ and\ tMOD_CAL}$). In addition, this field should be programmed higher than TIMING_CFG_7[CS_TO_CMD] if using CAL mode. 00000b - Reserved 00001b - 1 clock 00010b - 2 clocks 00011b - 3 clocks 00100b - 4 clocks 00101b - 5 clocks 00110b - 6 clocks 00111b - 7 clocks 01000b - 8 clocks 01001b - 9 clocks 01010b - 10 clocks 01011b - 11 clocks 01100b - 12 clocks 01101b - 13 clocks 01110b - 14 clocks 01111b - 15 clocks 10000b - 16 clocks 10001b - 17 clocks 10010b - 18 clocks 10011b - 19 clocks 10100b - 20 clocks 10101b - 21 clocks 10110b - 22 clocks 10111b - 23 clocks 11000b - 24 clocks 11001b - 25 clocks 11010b - 26 clocks 11011b - 27 clocks 11100b - 28 clocks 11101b - 29 clocks 11110b - 30 clocks 11111b - 31 clocks

12.4.6 DDR SDRAM timing configuration 1 (TIMING_CFG_1)

12.4.6.1 Offset

Register	Offset
TIMING_CFG_1	108h

12.4.6.2 Function

DDR SDRAM timing configuration register 1 sets the number of clock cycles between various SDRAM control commands.

12.4.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRETOACT				ACTTOPRE				ACTTORW				CASLAT			RSRVD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REFREC				WRREC				ACTTOACT				WRTORD			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.6.4 Fields

Field	Function
31-28 PRETOACT	Precharge-to-Activate. Precharge-to-activate interval (t_{RP}). Determines the number of clock cycles from a precharge command until an activate or refresh command is allowed. This field is concatenated with TIMING_CFG_3[EXT_PRETOACT] to obtain a 5-bit value for the total precharge to activate time. 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks

Table continues on the next page...

Field	Function
	0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
27-24 ACTTOPRE	Activate-to-Precharge. Activate to precharge interval (t_{RAS}). Determines the number of clock cycles from an activate command until a precharge command is allowed. This field is concatenated with TIMING_CFG_3[EXT_ACTTOPRE] to obtain a 6-bit value for the total activate to precharge time. Note that the decode of 0000-0011 is equal to 16-19 clocks when TIMING_CFG_3[EXT_ACTTOPRE] = 00, but it is equal to 0-3 clocks when TIMING_CFG_3[EXT_ACTTOPRE] = 01, 10, or 11. 0000b - 16 clocks 0001b - 17 clocks 0010b - 18 clocks 0011b - 19 clocks 0100b - 4 clock 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
23-20 ACTTORW	Activate-to-Read/Write. Activate to read/write interval for SDRAM (t_{RCD}). Controls the number of clock cycles from an activate command until a read or write command is allowed. This field is concatenated with TIMING_CFG_3[EXT_ACTTORW] to obtain a 5-bit value for the total activate to read/write time. 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks

Table continues on the next page...

Field	Function
	1111b - 15 clocks
19-17 CASLAT	CAS Latency. Number of clock cycles between registration of a READ command by the SDRAM and the availability of the first output data. If a READ command is registered at clock edge *n* and the latency is *m* clocks, data is available nominally coincident with clock edge *n* + *m*. This field is concatenated with TIMING_CFG_3[EXT_CASLAT] to obtain a 5-bit value for the total CAS latency. This value must be programmed at initialization as described in DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)) Note that the largest total CAS latency supported is 20 clocks. 000b - 1 clock 001b - 2 clocks 010b - 3 clocks 011b - 4 clocks 100b - 5 clocks 101b - 6 clocks 110b - 7 clocks 111b - 8 clocks
16 RSRVD	Reserved This bit is reserved, but it is readable and writeable.
15-12 REFREC	Refresh recovery. Refresh recovery time (t_{RFC}). Controls the number of clock cycles from a refresh command until an activate command is allowed. This field is concatenated with TIMING_CFG_3[EXTREFREC] to obtain a 10-bit value for the total refresh recovery. Note that hardware adds an additional 8 clock cycles to the final, 10-bit value of the refresh recovery, such that t_{RFC} is calculated as follows: t_{RFC} = {EXT_REFREC || REFREC} + 8. 0000b - 8 clocks 0001b - 9 clocks 0010b - 10 clocks 0011b - 11 clocks 1111b - 23 clocks
11-8 WRREC	Write recovery. Last data to precharge minimum interval (t_{WR}). Determines the number of clock cycles from the last data associated with a write command until a precharge command is allowed. If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this field needs to be programmed to (t_{WR} + 2 cycles). This field is concatenated with TIMING_CFG_3[EXT_WRREC] to obtain a 5-bit value for the total write recovery time. 0000b - Reserved when DDR_TIMING_CFG_3[EXT_WRREC] = 0; otherwise 0 clocks 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
7-4	Activate-to-activate.

Table continues on the next page...

Field	Function
ACTTOACT	Activate-to-activate interval (t_{RRD}). Number of clock cycles from an activate command until another activate command is allowed for a different logical bank in the same physical bank (chip select). 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
3-0 WRTORD	Write-to-read. Last write data pair to read command issue interval (t_{WTR}). Number of clock cycles between the last write data pair and the subsequent read command to the same physical bank. If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this field needs to be programmed to (t_{WTR} + 2 cycles). 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks

12.4.7 DDR SDRAM timing configuration 2 (TIMING_CFG_2)

12.4.7.1 Offset

Register	Offset
TIMING_CFG_2	10Ch

12.4.7.2 Function

DDR SDRAM timing configuration 2 sets the clock delay to data for writes.

12.4.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADD_LAT				Reserved					WR_LAT				EXT_WR_LAT	Reserved	RD_TO_PRE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_TO_PRE			Reserved				CKE_PLS			FOUR_ACT					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.7.4 Fields

Field	Function
31-28 ADD_LAT	Additive Latency. The additive latency must be set to a value less than TIMING_CFG_1[ACTTORW]. This field is added to TIMING_CFG_3]EXT_ADD_LAT]. The maximum total additive latency supported is 19 clocks. 0000b - 0 clocks 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - Reserved
27-23	Reserved

Table continues on the next page...

Field	Function
—	
22-19 WR_LAT	Write Latency. Note that the total write latency is equal to WR_LAT + ADD_LAT. Note that the total write latency must be at least 6 cycles if using unbuffered DIMMs in 1T timing mode. Note that this field is added to TIMING_CFG_2[EXT_WR_LAT]. The maximum write latency supported before adding the additive latency is 18 clocks. Note that values of 0b0000 to 0b0101 are reserved unless TIMING_CFG_2[EXT_WR_LAT] is programmed to a non-zero value. 0000b - 0 clocks 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
18 EXT_WR_LAT	Extended Write Latency. Note that the value of this field is added to the programmed value in TIMING_CFG_2[WR_LAT]. The largest total write latency supported before adding the additive latency is 18 clocks. 0b - 0 clocks 1b - 16 clocks
17 —	Reserved
16-13 RD_TO_PRE	Read-to-Precharge. Read to precharge (t_{RTP}). If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this field needs to be programmed to (t_{RTP} + 2 cycles). 0000b - Reserved 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
12-9 Reserved	Write Data Delay.

Table continues on the next page...

Field	Function
	Write command to write data strobe timing adjustment. Controls the amount of delay applied to the data and data strobes for writes. See DDR SDRAM Write Timing Adjustments for details. The write preamble will be driven high for 1/2 DRAM cycle, and then it will be driven low for 1/2 DRAM cycle. 0000b - 0 clock delay 0001b - 2 clock delay 0010b - 1/4 clock delay 0011b - 9/4 clock delay 0100b - 1/2 clock delay 0101b - 5/2 clock delay 0110b - 3/4 clock delay 0111b - Reserved 1000b - 1 clock delay 1001b - Reserved 1010b - 5/4 clock delay 1011b - Reserved 1100b - 3/2 clock delay 1101b - Reserved 1110b - 7/4 clock delay 1111b - Reserved
8-6 CKE_PLS	CKE Pulse. Minimum CKE pulse width (t_{CKE}). This field is concatenated with TIMING_CFG_3[EXT_CKE_PLS] to obtain a 4-bit value for the total minimum CKE pulse width. 000b - 8 clocks 001b - 1 clock 010b - 2 clocks 011b - 3 clocks 100b - 4 clocks 101b - 5 clocks 110b - 6 clocks 111b - 7 clocks
5-0 FOUR_ACT	Four Activate. Window for four activates (t_{FAW}). **NOTE:** If tFAW requires FOUR_ACT to be set higher than the supported values, then TIMING_CFG_1[ACTTOACT] should be programmed to avoid a t_{FAW} violation. This can be done by programming -> TIMING_CFG_1[ACTTOACT] = rounded_up[max(tRRD{_S}, tFAW/4)]. 000000b - Reserved 000001b - 1 cycle 000010b - 2 cycles 000011b - 3 cycles 000100b - 4 cycles 011111b - 31 cycles 100000b - 32 cycles

12.4.8 DDR SDRAM control configuration (DDR_SDRAM_CFG)

12.4.8.1 Offset

Register	Offset
DDR_SDRAM_CFG	110h

12.4.8.2 Function

The DDR SDRAM control configuration register enables the interface logic and specifies certain operating features such as self refreshing, error checking and correcting, registered DIMMs, and dynamic power management.

12.4.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MEM_EN	SREN	ECC_EN	RD_EN	Reserved	SDRAM_TYPE			Reserved		DYN_PWR	DBW		BE_8	Reserved	T3_EN
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	T2_EN	BA_INTLV_CTL							Reserved				HSE	ACC_ECC_EN	MEM_HALT	BI
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.8.4 Fields

Field	Function
31 MEM_EN	Memory Controller Enable. 0b - SDRAM interface logic is disabled. 1b - SDRAM interface logic is enabled. Must not be set until all other memory configuration parameters have been appropriately configured by initialization code.
30 SREN	Self Refresh Enable.

Table continues on the next page...

Field	Function
	0b - SDRAM self refresh is disabled during sleep. Whenever self-refresh is disabled, the system is responsible for preserving the integrity of SDRAM during sleep. 1b - SDRAM self refresh is enabled during sleep.
29 ECC_EN	ECC Enable. Note that uncorrectable read errors may cause an interrupt. 0b - No ECC errors are reported. No ECC interrupts are generated. 1b - ECC is enabled.
28 RD_EN	Registered DIMM Enable. Specifies the type of DIMM used in the system. Note that RD_EN and T2_EN must not both be set at the same time. See DDR SDRAM Registered DIMM Mode for restrictions that apply to the DDR controller when using DDR4 registered DIMMs. If using DDR4 registered DIMMs at a frequency that requires a latency adder in RC0F, then the DDR_SDRAM_WRLVL_CNTL_*[WRLVL_START] values must have an extra cycle for each latency adder cycle added by adding a value of 0x8 to each of the programmed settings. 0b - Indicates unbuffered DIMMs. 1b - Indicates registered DIMMs.
27 —	Reserved
26-24 SDRAM_TYPE	SDRAM Type. Type of SDRAM device to be used. This field will be used when issuing the automatic hardware initialization sequence to DRAM via Mode Register Set and Extended Mode Register Set commands. Default value is 111. 000-100 Reserved 101 DDR4 SDRAM 110 Reserved 111 Reserved
23-22 —	Reserved
21 DYN_PWR	Dynamic Power Management. 0b - Dynamic power management mode is disabled. 1b - Dynamic power management mode is enabled. If there is no ongoing memory activity, the SDRAM CKE signal is negated.
20-19 DBW	DRAM Data Bus Width. 00b - 64-bit bus is used. 01b - 32-bit bus is used. 10b - Reserved 11b - Reserved
18 BE_8	8-Beat Burst Enable. 0b - 4-beat bursts are used on the DRAM interface. This is only supported if DDR_SDRAM_CFG_2[OBC_CFG] is also set. 1b - 8-beat bursts are used on the DRAM interface.
17 —	Reserved
16	3T Timing Enable.

Table continues on the next page...

Field	Function
T3_EN	This field cannot be set if DDR_SDRAM_CFG[T2_EN] is also set. This field cannot be used with a 32-bit bus if 4-beat bursts are used. **NOTE:** 3T timing may not be used with 4-beat bursts, unless DDR_SDRAM_CFG_2[OBC_CFG] is set. 0b - 1T timing is enabled if T2_EN is cleared. The DRAM command/address are held for only 1 cycle on the DRAM bus. 1b - 3T timing is enabled. The DRAM command/address are held for 3 full cycles on the DRAM bus for every DRAM transaction. However, the chip select is only held for the third cycle.
15 T2_EN	2T Timing Enable. This field should not be set if DDR_SDRAM_CFG[T3_EN] is set. Note that RD_EN and T2_EN must not both be set at the same time. 0b - 1T timing is enabled if T3_EN is cleared. The DRAM command/address are held for only 1 cycle on the DRAM bus. 1b - 2T timing is enabled. The DRAM command/address are held for 2 full cycles on the DRAM bus for every DRAM transaction. However, the chip select is only held for the second cycle.
14-8 BA_INTLV_CTL	Bank (chip select) interleaving control. Set this field only when using bank interleaving. ('x' denotes a don't care bit value. All unlisted field values are reserved.) 0000000 No external memory banks are interleaved 1000000 External memory banks 0 and 1 are interleaved 0100000 External memory banks 2 and 3 are interleaved 1100000 External memory banks 0 and 1 are interleaved together and banks 2 and 3 are interleaved together xx00100 External memory banks 0 through 3 are all interleaved together
7-4 —	Reserved
3 HSE	Half-Strength Enable. Sets I/O driver impedance to calibrate to half strength. This calibrated impedance will be used by the MDIC, address/command, data, and clock impedance values, but only if automatic hardware calibration is enabled and the corresponding group's software override is disabled in the DDR control driver register(s) described in DDR Control Driver Register 1 (DDRCDR_1) and DDR Control Driver Register 2 (DDRCDR_2) 0b - I/O driver impedance will be calibrated to full strength. 1b - I/O driver impedance will be calibrated to half strength.
2 ACC_ECC_EN	Accumulated ECC enable. This can be used to save ECC pins/wires when using a DDR data bus width smaller than 64-bits. In this mode, the 8-bit ECC code will be transmitted/received across several beats, and it will be used to check 64-bits of data once 8-bits of ECC are accumulated. Note that using this mode guarantees that all single-bit ECC errors are corrected and detected for every 64-bits of data, and every 2-bit ECC error is detected for 64-bits of data. Unused ECC bits are driven high by the memory controller during write bursts if ACC_ECC_EN is set. 0b - Accumulated ECC is disabled 1b - Accumulated ECC is enabled
1 MEM_HALT	Memory Controller Halt. DDR memory controller halt. When this bit is set, the memory controller will not accept any new data read/write transactions to DDR SDRAM until the bit is cleared again. This can be used when bypassing initialization and forcing MODE REGISTER SET commands through software.

Table continues on the next page...

Field	Function
	0b - DDR controller will accept new transactions. 1b - DDR controller will finish any remaining transactions, and then it will remain halted until this bit is cleared by software.
0 BI	Bypass Initialization. See DDR training initialization address (DDR_INIT_ADDR) for details on avoiding ECC errors in this mode. 0b - DDR controller will cycle through initialization routine based on SDRAM_TYPE 1b - Initialization routine will be bypassed. Software is responsible for initializing memory through DDR_SDRAM_MD_CNTL register. If software is initializing memory, then the MEM_HALT bit can be set to prevent the DDR controller from issuing transactions during the initialization sequence. Note that the DDR controller will not issue a DLL reset to the DRAMs when bypassing the initialization routine, regardless of the value of DDR_SDRAM_CFG[DLL_RST_DIS]. If a DLL reset is required, then the controller should be forced to enter and exit self refresh after the controller is enabled.

12.4.9 DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)

12.4.9.1 Offset

Register	Offset
DDR_SDRAM_CFG_2	114h

12.4.9.2 Function

The DDR SDRAM control configuration register 2 provides more control configuration for the DDR controller.

12.4.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FRC_SR	Reserved	Reserved							ODT_CFG		Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_PR				DDR_SLOW	Reserved	QD_EN	UNQ_MRS_EN	Reserved	OBC_CFG	AP_EN	D_INIT	SPARE_CNFG	RCW_EN	CD_DIS	MD_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.9.4 Fields

Field	Function
31 FRC_SR	Force Self Refresh. 0b - DDR controller will operate in normal mode. 1b - DDR controller will enter self-refresh mode.
30 —	Reserved
29-23 —	Reserved
22-21 ODT_CFG	ODT configuration. This field defines how ODT will be driven to the on-chip IOs. See DDR Control Driver Register 1 (DDRCDR_1) and DDR Control Driver Register 2 (DDRCDR_2) which define the termination value that will be used. 00b - Never assert ODT to internal IOs 01b - Reserved 10b - Assert ODT to internal IOs only during reads to DRAM 11b - Reserved
20-16 —	Reserved
15-12 NUM_PR	Number of posted refreshes. This will determine how many posted refreshes, if any, can be issued at one time. Note that if posted refreshes are used, then this field, along with DDR_SDRAM_INTERVAL[REFINT], must be programmed such that the maximum t_{ras} specification cannot be violated. Patterns not shown are reserved. 0000b - Reserved

Table continues on the next page...

Field	Function
	0001b - 1 refresh will be issued at a time 0010b - 2 refreshes will be issued at a time 0011b - 3 refreshes will be issued at a time 1000b - 8 refreshes will be issued at a time
11 DDR_SLOW	DDR Slow Frequency. Indicates to the controller if it will be run at a lower frequency. 0b - The DDR controller will be run at data rates of 1250 MT/s or higher. 1b - The DDR controller will be run at data rates of less than 1250 MT/s.
10 —	Reserved
9 QD_EN	Quad-Rank Enable. Determines if a quad-ranked DIMM is used. This bit should also be set if quad-stacked discrete memory chips are used. 0b - Quad-ranked DIMMs are not used. 1b - Quad-ranked DIMMs are used.
8 UNQ_MRS_EN	Unique MRS Enable. Determines if the DDR_SDRAM_MODE_{3:8} and DDR_SDRAM_MODE_{11:16} registers will be used when initializing the memories for chip selects 1, 2, and 3. These can be used to provide unique values to the Mode Registers of the DRAM to allow different termination values for each rank.
7 —	
6 OBC_CFG	On-The-Fly Burst Chop Configuration. Determines if on-the-fly Burst Chop will be used. If on-the-fly Burst Chop mode is not used, then 8-beat burst mode should be used. DDR_SDRAM_CFG[BE_8] should be cleared for on-the-fly Burst Chop mode. 0b - On-the-fly Burst Chop mode is disabled. Fixed burst lengths as defined in DDR_SDRAM_CFG[BE_8] are used. 1b - On-the-fly Burst Chop mode will be used. DDR_SDRAM_CFG[BE_8] should be cleared for on-the-fly Burst Chop mode. DDR_SDRAM_CFG[DBW] should also be programmed for 64-bit bus or 32-bit bus.
5 AP_EN	Address Parity Enable. Determines if address parity will be generated and checked for the address and control signals when using registered DIMMs. If address parity is used, the MAPAR_OUT and MAPAR_ERR_B pins will be used to drive the parity bit and to receive errors from the open-drain parity error signal. Even parity will be used, and parity will be generated for the MA[15:0], MBA[2:0], MRAS_B, MCAS_B, MWE_B signals. Parity will not be generated for the MODTn, MCSn_B, and MCKEn signals. Note that address parity should not be used for non-zero values of TIMING_CFG_3[CNTL_ADJ]. 0b - Address parity will not be used 1b - Address parity will be used
4 D_INIT	DRAM data initialization. This bit is set by software, and it is cleared by hardware. If software sets this bit before the memory controller is enabled, the controller will automatically initialize DRAM after it is enabled. This bit will be automatically cleared by hardware once the initialization is completed. This data initialization bit should only be set when the controller is idle. 0b - There is not data initialization in progress, and no data initialization is scheduled 1b - The memory controller will initialize memory once it is enabled. This bit will remain asserted until the initialization is complete. The value in DDR_DATA_INIT register will be used to initialize memory.

Table continues on the next page...

Field	Function
3 SPARE_CNFG	Spare Config Bits. This field is currently unused.
2 RCW_EN	Register Control Word Enable. If registered DIMMs are used, it may be necessary to write the register control words before issuing commands to DRAM. If this bit is set, the controller will write the register control words after DDR_SDRAM_CFG[MEM_EN] is set, unless DDR_SDRAM_CFG[BI] is set. The register control words will be written with the values in DDR_SDRAM_RCW_1 and DDR_SDRAM_RCW_2. 0b - Register control words will not be automatically written during DRAM initialization 1b - Register control words will be automatically written during DRAM initialization. This bit should only be set if registered DIMMs are used, and the default settings need to be modified.
1 CD_DIS	Corrupted Data Disable. If this bit is set, then the corrupted data feature will be disabled. When the corrupted data feature is enabled, the DDR controller will inverted the generated ECC code for any beat of data which is known to have corrupted data. When a read to the corrupted data is later generated, the *ERR_DETECT[CDE]* error will be set if error reporting is enabled. 0b - Corrupted data is enabled 1b - Corrupted data is disabled
0 MD_EN	Mirrored DIMM Enable. Some DIMMs will be mirrored, where certain MA and MBA pins are mirrored on one side of the DIMM. When this bit is set, the controller will know to swap these signals before transmitting to the DRAM. The controller will assume that CS1 and CS3 are the 'mirrored' ranks of memory. The following signals are mirrored for DDR4: • MBA[0] vs. MBA[1] • MA[3] vs. MA[4] • MA[5] vs. MA[6] • MA[7] vs. MA[8] • MA[11] vs. MA[13] • MBG0 vs. MBG1 (if 2 bank group bits are used) Note that MBG0 is the same as MBA[2] from the DDR controller. If using mirrored DIMMs for DDR4 mode, then CS0 and CS1 must be programmed to use the same number of bank group bits in CS*a*_CONFIG[BG_BITS_CS*a*]. CS2 and CS3 must also be programmed to use the same number of bank group bits. 0b - Mirrored DIMMs are not used 1b - Mirrored DIMMs are used

12.4.10 DDR SDRAM mode configuration (DDR_SDRAM_MODE)

12.4.10.1 Offset

Register	Offset
DDR_SDRAM_MODE	118h

12.4.10.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers.

12.4.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.10.4 Fields

Field	Function
31-16 ESDMODE	Extended SDRAM mode. Specifies the initial value loaded into the DDR SDRAM extended mode register. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb of ESDMODE, which, in the big-endian convention shown here, corresponds to ESDMODE[15]. The msb of the SDRAM extended mode register value must be stored at ESDMODE[0].
15-0 SDMODE	SDRAM mode. Specifies the initial value loaded into the DDR SDRAM mode register. The range of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of SDMODE, which, in the big-endian convention, corresponds to SDMODE[15]. The msb of the SDRAM mode register value must be stored at SDMODE[0]. Because the memory controller forces SDMODE[7] to certain values depending on the state of the initialization sequence, (for resetting the SDRAM's DLL) the corresponding bits of this field are ignored by the memory controller. Note that SDMODE[7] is mapped to MA[8].

12.4.11 DDR SDRAM mode configuration 2 (DDR_SDRAM_MODE_2)

12.4.11.1 Offset

Register	Offset
DDR_SDRAM_MODE_2	11Ch

12.4.11.2 Function

The DDR SDRAM mode 2 configuration register sets the values loaded into the DDR's extended mode 2 and 3 registers.

12.4.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.11.4 Fields

Field	Function
31-16 ESDMODE2	Extended SDRAM mode 2. Specifies the initial value loaded into the DDR SDRAM extended 2 mode register. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE2, which, in the big-endian convention shown here, corresponds to ESDMODE2[15]. The msb of the SDRAM extended mode 2 register value must be stored at ESDMODE2[0].
15-0	Extended SDRAM mode 3.

Field	Function
ESDMODE3	Specifies the initial value loaded into the DDR SDRAM extended 3 mode register. The range of legal values of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of ESDMODE3, which, in the big-endian convention shown here, corresponds to ESDMODE3[15]. The msb of the SDRAM extended mode 3 register value must be stored at ESDMODE3[0].

12.4.12 DDR SDRAM mode control (DDR_SDRAM_MD_CNTL)

12.4.12.1 Offset

Register	Offset
DDR_SDRAM_MD_CNTL	120h

12.4.12.2 Function

The DDR SDRAM mode control register allows the user to carry out the following tasks:

Issue a mode register set command to a particular chip select

Issue an immediate refresh to a particular chip select

Issue an immediate precharge or precharge all command to a particular chip select

Force the CKE signals to a specific value

Note that MD_EN, SET_REF, and SET_PRE are mutually exclusive; only one of these fields can be set at a time.

This table shows how DDR_SDRAM_MD_CNTL fields should be set for each of the tasks described above.

Table 12-5. Settings of DDR_SDRAM_MD_CNTL Fields

Field	Mode Register Set	Refresh	Precharge	Clock Enable Signals Control
MD_EN	1	0	0	-
SET_REF	0	1	0	-
SET_PRE	0	0	1	-
CS_SEL	Chooses chip select (CS)			-

Table continues on the next page...

Table 12-5. Settings of DDR_SDRAM_MD_CNTL Fields (continued)

Field	Mode Register Set	Refresh	Precharge	Clock Enable Signals Control
MD_SEL	Select mode register.	-	Selects logical bank and bank group	-
MD_VALUE	Value written to mode register	-	Only bit 7 is significant.	-
CKE_CNTL	0	0	0	

12.4.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MD_EN	CS_SEL			MD_SEL				SET_REF	SET_PRE	CKE_CNTL		Reserved	Reserved	MD_VALUE	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MD_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.12.4 Fields

Field	Function
31 MD_EN	Mode enable. Setting this bit specifies that valid data in MD_VALUE is ready to be written to DRAM as one of the following commands: • MODE REGISTER SET • EXTENDED MODE REGISTER SET • EXTENDED MODE REGISTER SET 2 • EXTENDED MODE REGISTER SET 3 The specific command to be executed is selected by setting MD_SEL. In addition, the chip select must be chosen by setting CS_SEL.MD_EN is set by software and cleared by hardware once the command has been issued. 0b - Indicates that no mode register set command needs to be issued. 1b - Indicates that valid data contained in the register is ready to be issued as a mode register set command.
30-28 CS_SEL	Select chip select.

Table continues on the next page...

Field	Function
	Specifies the chip select that will be driven active due to any command forced by software in DDR_SDRAM_MD_CNTL. 000b - Chip select 0 is active 001b - Chip select 1 is active 010b - Chip select 2 is active 011b - Chip select 3 is active 100b - Chip select 0 and chip select 1 are active 101b - Chip select 2and chip select 3 are active 110b - Reserved 111b - Reserved
27-24 MD_SEL	Mode register select. MD_SEL specifies one of the following: • During a mode select command, selects the SDRAM mode register to be changed • During a precharge command, selects the SDRAM logical bank to be precharged. A precharge all command ignores this field. • During a refresh command, this field is ignored. In DDR4 mode, MD_SEL[0:1] will represent the value on MBG[1:0], and MD_SEL[2:3] will represent the value on MBA[1:0]. 0000b - MR 0001b - EMR 0010b - EMR2 0011b - EMR3
23 SET_REF	Set refresh. Forces an immediate refresh to be issued to the chip select specified by DDR_SDRAM_MD_CNTL[CS_SEL]. This bit will be set by software and cleared by hardware once the command has been issued. 0b - Indicates that no refresh command needs to be issued. 1b - Indicates that a refresh command is ready to be issued.
22 SET_PRE	Set precharge. Forces a precharge or precharge all to be issued to the chip select specified by DDR_SDRAM_MD_CNTL[CS_SEL]. This bit will be set by software and cleared by hardware once the command has been issued. 0b - Indicates that no precharge all command needs to be issued. 1b - Indicates that a precharge all command is ready to be issued.
21-20 CKE_CNTL	Clock enable control. Allows software to globally clear or set the all CKE signals issued to DRAM. Once software has forced the value driven on CKE, that value will continue to be forced until software clears the CKE_CNTL bits. At that time, the DDR controller will continue to drive the CKE signals to the same value forced by software until another event causes the CKE signals to change (that is, self refresh entry/exit, power down entry/exit). 00b - CKE signals are not forced by software. 01b - CKE signals are forced to a low value by software. 10b - CKE signals are forced to a high value by software. 11b - Reserved
19 —	Reserved
18 —	Reserved

Table continues on the next page...

Field	Function
17-0 MD_VALUE	Mode register value. This field, which specifies the value that will be presented on the memory address pins of the DDR controller during a mode register set command, is significant only when this register is used to issue a mode register set command or a precharge or precharge all command. Note that the 2 most significant bits of this register are implemented for future use, but they will not currently affect the MRS commands to the DRAM, as A[17:16] of the DRAMs MR registers are reserved. For a mode register set command, this field contains the data to be written to the selected mode register. For a precharge command, only bit five is significant: 000000000000000000b - Issue a precharge command; MD_SEL selects the logical bank to be precharged 000000000000000001b - Issue a precharge all command; all logical banks are precharged

12.4.13 DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL)

12.4.13.1 Offset

Register	Offset
DDR_SDRAM_INTERVAL	124h

12.4.13.2 Function

The DDR SDRAM interval configuration register sets the number of DRAM clock cycles between bank refreshes issued to the DDR SDRAMs. In addition, the number of DRAM cycles that a page is maintained after it is accessed is provided here.

12.4.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REFINT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		BSTOPRE													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.13.4 Fields

Field	Function
31-16 REFINT	Refresh interval. Represents the number of memory bus clock cycles between refresh cycles. Depending on DDR_SDRAM_CFG_2[NUM_PR], some number of rows are refreshed in each DDR SDRAM physical bank during each refresh cycle. The value for REFINT depends on the specific SDRAMs used and the interface clock frequency. Refreshes will not be issued when the REFINT is set to all 0s. This field is concatenated with TIMING_CFG_4[EXT_REFINT] to obtain a 17-bit value for the total refresh interval.
15-14 —	Reserved
13-0 BSTOPRE	Precharge interval. Sets the duration (in memory bus clocks) that a page is retained after a DDR SDRAM access. If BSTOPRE is zero, the DDR memory controller uses auto-precharge read and write commands rather than operating in page mode. This is called global auto-precharge mode.

12.4.14 DDR SDRAM data initialization (DDR_DATA_INIT)

12.4.14.1 Offset

Register	Offset
DDR_DATA_INIT	128h

12.4.14.2 Function

The DDR SDRAM data initialization register provides the value that will be used to initialize memory if DDR_SDRAM_CFG2[D_INIT] is set.

12.4.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_VALUE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.14.4 Fields

Field	Function
31-0 INIT_VALUE	Initialization value. Represents the value that DRAM will be initialized with if DDR_SDRAM_CFG2[D_INIT] is set.

12.4.15 DDR SDRAM clock control (DDR_SDRAM_CLK_CNTL)

12.4.15.1 Offset

Register	Offset
DDR_SDRAM_CLK_CNTL	130h

12.4.15.2 Function

The DDR SDRAM clock control configuration register provides a 1/8-cycle clock adjustment.

12.4.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					CLK_ADJUST					Reserved					
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.15.4 Fields

Field	Function
31-27 —	Reserved
26-22 CLK_ADJUST	Clock Adjust. 00000 Clock is launched aligned with address/command 00001 Clock is launched 1/16 applied cycle after address/command 00010 Clock is launched 1/8 applied cycle after address/command 00011 Clock is launched 3/16 applied cycle after address/command 00100 Clock is launched 1/4 applied cycle after address/command 00101 Clock is launched 5/16 applied cycle after address/command 00110 Clock is launched 3/8 applied cycle after address/command 00111 Clock is launched 7/16 applied cycle after address/command 01000 Clock is launched 1/2 applied cycle after address/command 01001 Clock is launched 9/16 applied cycle after address/command 01010 Clock is launched 5/8 applied cycle after address/command 01011 Clock is launched 11/16 applied cycle after address/command 01100 Clock is launched 3/4 applied cycle after address/command 01101 Clock is launched 13/16 applied cycle after address/command

Table continues on the next page...

Field	Function
	01110 Clock is launched 7/8 applied cycle after address/command 01111 Clock is launched 15/16 applied cycle after address/command 10000 Clock is launched 1 applied cycle after address/command 10010-11110 Reserved
21-0 —	Reserved

12.4.16 DDR training initialization address (DDR_INIT_ADDR)

12.4.16.1 Offset

Register	Offset
DDR_INIT_ADDR	148h

12.4.16.2 Function

The DDR SDRAM initialization address register provides the address that will be used for the data strobe to data skew adjustment and automatic CAS_B to preamble calibration after POR.

When the default value is used (that is, address 0x0), all chip selects are considered for the training. If DDR_INIT_ADDR is set to any value other than the default value of address zero, then only the first chip select will be trained. When multiple chip selects are used and DQS/DQ skew is not common between chip selects/ranks, then the default address value of 0x0 is be recommended to obtain the best timing margins.

After the skew adjustment, this address will contain bad ECC data. This is not important at POR, as all of memory should be subsequently initialized if ECC is enabled (either by software or through the use of DDR_SDRAM_CFG_2[D_INIT]).

If an HRESET_B has been issued after the DRAM is in self-refresh mode, however, memory is not initialized, so this address should be written to using an 8- or 32-byte transaction to avoid possible ECC errors if this address could later be accessed.

12.4.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.16.4 Fields

Field	Function
31-0 INIT_ADDR	Initialization address. Represents the address that will be used for the data strobe to data skew adjustment and automatic CAS to preamble calibration at POR. This address will be written to during the initialization sequence.

12.4.17 DDR training initialization extended address (DDR_INIT_EXT_ADDRESS)

12.4.17.1 Offset

Register	Offset
DDR_INIT_EXT_ADDRESS	14Ch

12.4.17.2 Function

The DDR SDRAM initialization extended address register provides the extended address that will be used for the data strobe to data skew adjustment and automatic CAS_B to preamble calibration after POR.

12.4.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UIA	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								INIT_EXT_ADDR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.17.4 Fields

Field	Function
31 UIA	Use initialization address. 0b - Use the default address for training sequence as calculated by the controller. This will be the first valid address in each enabled chip select. 1b - Use the initialization address programmed in DDR_INIT_ADDR and DDR_INIT_EXT_ADDR.
30-8 —	Reserved
7-0 INIT_EXT_ADDR	Initialization extended address. Represents the extended address that will be used for the data strobe to data skew adjustment and automatic CAS_B to preamble calibration at POR. This extended address will be written to during the initialization sequence.

12.4.18 DDR SDRAM timing configuration 4 (TIMING_CFG_4)

12.4.18.1 Offset

Register	Offset
TIMING_CFG_4	160h

12.4.18.2 Function

The DDR SDRAM timing configuration 4 register provides additional timing fields.

12.4.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RWT				WRT				RRT				WWT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXT_RWT		Reserved	EXT_WRT	Reserved	EXT_RRT	Reserved	EXT_WWT	Reserved			EXT_REFINT	Reserved		DLL_LOCK	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.18.4 Fields

Field	Function
31-28 RWT	Read-to-write turnaround for same chip select. Specifies how many cycles will be added between a read to write turnaround for transactions to the same chip select. If a value of 0000 is chosen, then the DDR controller will use the value used for transactions to different chip selects, as defined in TIMING_CFG_0[RWT]. This field can be used to improve performance when operating in burst-chop mode by forcing transactions to the same chip select to use extra cycles, while transaction to different chip selects can utilize the tri-state time on the DRAM interface. Regardless of the value that is set in this field, the value defined by TIMING_CFG_0[RWT] will also be met before issuing a write command. 0000b - Default 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
27-24 WRT	Write-to-read turnaround for same chip select. Specifies how many cycles will be added between a write to read turnaround for transactions to the same chip select. If a value of 0000 is chosen, then the DDR controller will use the value used for transactions to different chip selects, as defined in TIMING_CFG_0[WRT]. This field can be used to improve

Table continues on the next page...

Field	Function
	performance when operating in burst-chop mode by forcing transactions to the same chip select to use extra cycles, while transaction to different chip selects can utilize the tri-state time on the DRAM interface. Regardless of the value that is set in this field, the value defined by TIMING_CFG_0[WRT] will also be met before issuing a read command. 0000b - Default 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
23-20 RRT	Read-to-read turnaround for same chip select. Specifies how many cycles will be added between reads to the same chip select. If a value of 0000 is chosen, then 2 cycles will be required between read commands to the same chip select if 4-beat bursts are used (4 cycles will be required if 8-beat bursts are used). Note that fixed 4-beat bursts are not supported. However, this field may be used to add extra cycles when burst-chop mode is used, and the DDR controller must wait 4 cycles for read-to-read transactions to the same chip select. 0000b - BL/2 clocks 0001b - BL/2 + 1 clock 0010b - BL/2 + 2 clocks 0011b - BL/2 + 3 clocks 0100b - BL/2 + 4 clocks 0101b - BL/2 + 5 clocks 0110b - BL/2 + 6 clocks 0111b - BL/2 + 7 clocks 1000b - BL/2 + 8 clocks 1001b - BL/2 + 9 clocks 1010b - BL/2 + 10 clocks 1011b - BL/2 + 11 clocks 1100b - BL/2 + 12 clocks 1101b - BL/2 + 13 clocks 1110b - BL/2 + 14 clocks 1111b - BL/2 + 15 clocks
19-16 WWT	Write-to-write turnaround for same chip select. Specifies how many cycles will be added between writes to the same chip select. If a value of 0000 is chosen, then 2 cycles will be required between write commands to the same chip select if 4-beat bursts are used (4 cycles will be required if 8-beat bursts are used). Note that fixed 4-beat bursts are not supported. However, this field may be used to add extra cycles when burst-chop mode is used, and the DDR controller must wait 4 cycles for write-to-write transactions to the same chip select. 0000b - BL/2 clocks 0001b - BL/2 + 1 clock 0010b - BL/2 + 2 clocks 0011b - BL/2 + 3 clocks 0100b - BL/2 + 4 clocks 0101b - BL/2 + 5 clocks

Table continues on the next page...

Field	Function
	0110b - BL/2 + 6 clocks 0111b - BL/2 + 7 clocks 1000b - BL/2 + 8 clocks 1001b - BL/2 + 9 clocks 1010b - BL/2 + 10 clocks 1011b - BL/2 + 11 clocks 1100b - BL/2 + 12 clocks 1101b - BL/2 + 13 clocks 1110b - BL/2 + 14 clocks 1111b - BL/2 + 15 clocks
15-14 EXT_RWT	Extended read-to-write turnaround (tRTW). Specifies how many extra cycles will be added between a read to write turnaround. If 0 clocks is chosen, then the DDR controller will use a fixed number based on the CAS latency and write latency. Choosing a value other than 0 adds extra cycles past this default calculation. As a default the DDR controller will determine the read-to-write turnaround as CL - WL + BL/2 + 2. In this equation, CL is the CAS latency rounded up to the next integer, WL is the programmed write latency, and BL is the burst length. This field is concatenated with TIMING_CFG_0[RWT] to obtain a 4-bit value for the total read-to-write turnaround 00b - 0 clocks 01b - 1 clock 10b - 2 clocks 11b - 3 clocks
13 —	Reserved
12 EXT_WRT	Extended write-to-read turnaround. Specifies how many extra cycles will be added between a write to read turnaround. If 0 clocks is chosen, then the DDR controller will use a fixed number based on the, read latency, and write latency. Choosing a value other than 0 adds extra cycles past this default calculation. As a default, the DDR controller will determine the write-to-read turnaround as WL - CL + BL/2 + 1. In this equation, CL is the CAS latency rounded down to the next integer, WL is the programmed write latency, and BL is the burst length. This field is concatenated with TIMING_CFG_0[WRT] to obtain a 3-bit value for the total write-to-read turnaround 0b - 0 clocks 1b - 4 clocks
11 —	Reserved
10 EXT_RRT	Extended read-to-read turnaround. Specifies how many extra cycles will be added between reads to different chip selects. As a default, 3 cycles will be required between read commands to different chip selects. Extra cycles may be added with this field. Note: If 8-beat bursts are enabled, then 5 cycles will be the default. This field is concatenated with TIMING_CFG_0[RRT] to obtain a 3-bit value for the total read-to-read turnaround 0b - 0 clocks 1b - 4 clocks
9 —	Reserved
8 EXT_WWT	Extended write-to-write turnaround. Specifies how many extra cycles will be added between writes to different chip selects. As a default, 2 cycles will be required between write commands to different chip selects. Extra cycles may be added with this field. Note: If 8-beat bursts are enabled, then 4 cycles will be the default. This field is concatenated with TIMING_CFG_0[WWT] to obtain a 3-bit value for the total write-to-write turnaround

Table continues on the next page...

Field	Function
	0b - 0 clocks 1b - 4 clocks
7-5 —	Reserved
4 EXT_REFINT	Extended Refresh Interval. Represents the number of memory bus clock cycles between refresh cycles. Depending on DDR_SDRAM_CFG_2[NUM_PR], some number of rows are refreshed in each DDR SDRAM physical bank during each refresh cycle. The value for REFINT depends on the specific SDRAMs used and the interface clock frequency. Refreshes will not be issued when the REFINT is set to all 0s. This field is concatenated with DDR_SDRAM_INTERVAL[REFINT] to obtain a 17-bit value for the total refresh interval. 0b - 0 clocks 1b - 65,536 clocks
3-2 —	Reserved
1-0 DLL_LOCK	DDR SDRAM DLL Lock Time. This provides the number of cycles that it will take for the DRAMs DLL to lock at POR and after exiting self refresh. The controller will wait the specified number of cycles before issuing any commands after exiting POR or self refresh. 00b - 200 clocks 01b - 512 clocks 10b - 1024 clocks 11b - Reserved

12.4.19 DDR SDRAM timing configuration 5 (TIMING_CFG_5)

12.4.19.1 Offset

Register	Offset
TIMING_CFG_5	164h

12.4.19.2 Function

The DDR SDRAM timing configuration 5 register provides additional timing fields.

12.4.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			RODT_ON					Reserved	RODT_OFF			Reserved			WODT_ON
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WODT_ON				Reserved	WODT_OFF			Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.19.4 Fields

Field	Function
31-29 —	Reserved
28-24 RODT_ON	Read to ODT on. Specifies the number of cycles that will pass from when a read command is placed on the DRAM bus until the assertion of the relevant ODT signal(s). The default case (00000) provides a decode of [CASLAT - WR_LAT] to provide the expected default. If 2T timing is used, an extra cycle will automatically be added to the value selected in this field. Patterns not shown are reserved. 00000b - CASLAT - WR_LAT 00001b - 0 clocks 00010b - 1 clock 00011b - 2 clocks 01100b - 11 clocks
23 —	Reserved
22-20 RODT_OFF	Read to ODT off. Specifies the number of cycles that the relevant ODT signal(s) will remain asserted for each read transaction. The default case (000) will leave the ODT signal(s) asserted for 4 DRAM cycles. 000b - 4 clocks 001b - 1 clock 010b - 2 clocks 011b - 3 clocks 100b - 4 clocks 101b - 5 clocks 110b - 6 clocks

Table continues on the next page...

Field	Function
	111b - 7 clocks
19-17 —	Reserved
16-12 WODT_ON	Write to ODT on. Specifies the number of cycles that will pass from when a write command is placed on the DRAM bus until the assertion of the relevant ODT signal(s). The default case (00000) provides a decode of 0 cycles to provide the expected default. If 2T timing is used, an extra cycle will automatically be added to the value selected in this field. Patterns not shown are reserved. 00000b - 0 clocks 00001b - 0 clocks 00010b - 1 clock 00011b - 2 clocks 00110b - 5 clocks
11 —	Reserved
10-8 WODT_OFF	Write to ODT off. Specifies the number of cycles that the relevant ODT signal(s) will remain asserted for each write transaction. The default case (000) will leave the ODT signal(s) asserted for 4 DRAM cycles. **NOTE:** Value of 4 clock cycles must be selected for this field. 000b - 4 clocks 001b - 1 clock 010b - 2 clocks 011b - 3 clocks 100b - 4 clocks 101b - 5 clocks 110b - 6 clocks 111b - 7 clocks
7-0 —	Reserved

12.4.20 DDR SDRAM timing configuration 7 (TIMING_CFG_7)

12.4.20.1 Offset

Register	Offset
TIMING_CFG_7	16Ch

12.4.20.2 Function

The DDR SDRAM timing configuration 7 register provides additional timing fields required.

12.4.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		CKE_RST		CKSRE				CKSRX				PAR_LAT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CS_TO_CMD				Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.20.4 Fields

Field	Function
31-30 —	Reserved
29-28 CKE_RST	CKE Reset Time. CKE reset time (t_{XPR}). Specifies the number of cycles the DDR controller must wait after asserting CKE after RESET until the first MRS command. This field is also used to determine t_{XS} when exiting self refresh. Therefore, this should be programmed to the maximum of (t_{XPR}, t_{XS}) if they are different for a particular memory. 00b - 200 clocks 01b - 256 clocks 10b - 512 clocks 11b - 1024 clocks
27-24 CKSRE	Clock After Self Refresh Entry. Valid clock after Self Refresh entry (t_{CKSRE}). Specifies the number of cycles the DDR controller must drive valid MCK/MCK_B after entering self refresh before the clocks are allowed to stop. TIMING_CFG_7[CS_TO_CMD] should also be added to t_{CKSRE} to obtain the final value for this field. 0000b - 15 clocks 0001b - 6 clocks 0010b - 7 clocks 0011b - 8 clocks 0100b - 9 clocks 0101b - 10 clocks

Table continues on the next page...

Field	Function
	0110b - 11 clocks 0111b - 12 clocks 1000b - 13 clocks 1001b - 14 clocks 1010b - 15 clocks 1011b - 16 clocks 1100b - 17 clocks 1101b - 18 clocks 1110b - 19 clocks 1111b - 32 clocks
23-20 CKSRX	Clock After Self Refresh Exit. Valid clock after Self Refresh exit (t_{CKSRX}). Specifies the number of cycles the DDR controller must drive valid MCK/MCK_B after exiting self refresh before the clocks are allowed to stop. 0000b - 15 clocks 0001b - 6 clocks 0010b - 7 clocks 0011b - 8 clocks 0100b - 9 clocks 0101b - 10 clocks 0110b - 11 clocks 0111b - 12 clocks 1000b - 13 clocks 1001b - 14 clocks 1010b - 15 clocks 1011b - 16 clocks 1100b - 17 clocks 1101b - 18 clocks 1110b - 19 clocks 1111b - 27 clocks
19-16 PAR_LAT	Parity latency. Specifies the number of cycles to be used for the parity latency for DDR4 memories. Patterns not shown are reserved. 0000b - Disabled 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks
15-8 —	Reserved
7-4 CS_TO_CMD	Chip select to command latency. Specifies the number of cycles from a chip select until the command is launched. This should only be used with DDR4. The DDR controller will automatically apply this latency after it has written the MR4 register during DDR4 SDRAM initialization. It will also be enabled when the controller is enabled if DDR_SDRAM_CFG[BI] is set. However, if software is going to use the DDR_SDRAM_MD_CNTL register to intialize the DRAM's MR registers instead of allowing for automatic calibration, then this field cannot be set until software has properly programmed the DDR4 MR4 register to enable CAL mode. Patterns not shown are reserved.

Table continues on the next page...

Field	Function
	0000b - Disabled 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks
3-0 —	Reserved

12.4.21 DDR ZQ calibration control (DDR_ZQ_CNTL)

12.4.21.1 Offset

Register	Offset
DDR_ZQ_CNTL	170h

12.4.21.2 Function

The DDR ZQ Calibration Control register provides the enable and controls required for ZQ calibration.

There is a limitation for various DRAM timing parameters when ZQ calibration is used. The factors involved in this limitation are DDR_ZQ_CNTL[ZQOPER], DDR_ZQ_CNTL[ZQCS], TIMING_CFG_1[PRETOACT], TIMING_CFG_1[REFREC], DDR_SDRAM_INTERVAL[REFINT], and the number of chip selects enabled. If the following condition is true:

[((DDR_ZQ_CNTL[ZQOPER] + DDR_ZQ_CNTL[ZQCS])* (# enabled chip selects)) +

TIMING_CFG_1[PRETOACT] +

TIMING_CFG_1[REFREC] + $2t_{CK}$] > (DDR_SDRAM_INTERVAL[REFINT]),

then it is possible that one refresh will be skipped when the controller is exiting self refresh. If this is an issue, then posted refreshes could be used to extend the refresh interval. Another alternative is to use the DDR_SDRAM_MD_CNTL register to force an extra refresh to each chip select after exiting self refresh mode. However, timing parameters for most devices/frequencies will not allow for a refresh to be missed.

12.4.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ZQ_EN	Reserved			ZQINIT				Reserved				ZQOPER			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				ZQCS				Reserved				ZQCS_INT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.21.4 Fields

Field	Function
31 ZQ_EN	ZQ Calibration Enable. This bit determines if ZQ calibrating will be used. 0b - ZQ Calibration will not be used. 1b - ZQ Calibration will be used. A ZQCL command will be issued by the DDR controller after POR and anytime the DDR controller is exiting self refresh. A ZQCS command will be issued every 32 refresh sequences to account for VT variations.
30-28 —	Reserved
27-24 ZQINIT	ZQ Calibration Initialization Time. POR ZQ Calibration Time (t_{ZQinit}). Determines the number of cycles that must be allowed for DRAM ZQ calibration at POR. Each chip select will be calibrated separately, and this time must elapse after the ZQCL command is issued for each chip select before a separate command may be issued. Patterns not shown are reserved. 0111b - 128 clocks 1000b - 256 clocks 1001b - 512 clocks 1010b - 1024 clocks
23-20	Reserved

Table continues on the next page...

Field	Function
—	
19-16 ZQOPER	ZQ Calibration Operation Time. Normal Operation Full Calibration Time (t_{ZQoper}). Determines the number of cycles that must be allowed for DRAM ZQ calibration when exiting self refresh. Each chip select will be calibrated separately, and this time must elapse after the ZQCL command is issued for each chip select before a separate command may be issued. Patterns not shown are reserved. 0111b - 128 clocks 1000b - 256 clocks 1001b - 512 clocks 1010b - 1024 clocks
15-12 —	Reserved
11-8 ZQCS	ZQ Calibration Short Time. Normal Operation Short Calibration Time (t_{ZQCS}). Determines the number of cycles that must be allowed for DRAM ZQ calibration during dynamic calibration which is issued every *ZQCS_INT* refresh sequences. Each chip select will be calibrated separately, and this time must elapse after the ZQCS command is issued for each chip select before a separate command may be issued. Patterns not shown are reserved. 0000b - 1 clock 0001b - 2 clocks 0010b - 4 clocks 0011b - 8 clocks 0100b - 16 clocks 0101b - 32 clocks 0110b - 64 clocks 0111b - 128 clocks 1000b - 256 clocks 1001b - 512 clocks
7-4 —	Reserved
3-0 ZQCS_INT	ZQCS Interval. Determines the number of refresh sequences that will pass between each ZQCS calibration. 0000b - 32 refresh sequences 0001b - 64 refresh sequences 0010b - 128 refresh sequences 0011b - 256 refresh sequences 0100b - 512 refresh sequences 0101b - 1024 refresh sequences 0110b - 2048 refresh sequences 0111b - 4096 refresh sequences 1000b - 8192 refresh sequences 1001b - 16384 refresh sequences 1010b - 32768 refresh sequences 1011b - Reserved 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - ZQCS calibration disabled

12.4.22 DDR write leveling control (DDR_WRLVL_CNTL)

12.4.22.1 Offset

Register	Offset
DDR_WRLVL_CNTL	174h

12.4.22.2 Function

The DDR Write Leveling Control register provides controls for write leveling.

12.4.22.3 Diagram

12.4.22.4 Fields

Field	Function
31 WRLVL_EN	Write Leveling Enable.

Table continues on the next page...

Field	Function
	This bit determines if write leveling will be used. If this bit is set, then the DDR controller will perform write leveling immediately after initializing the DRAM. 0b - Write leveling will not be used 1b - Write leveling will be used
30 WRLVL_DONE	Write Leveling Done. This bit will be set by hardware once write leveling has been completed. This is a read-only bit, and it will only clear after a reset to the part has been issued. 0b - Write leveling has not completed 1b - Write leveling has completed
29-27 —	Reserved
26-24 WRLVL_MRD	Write Leveling MRD. First DQS pulse rising edge after margining mode is programmed (t_{WL_MRD}). Determines how many cycles to wait after margining mode has been programmed before the first DQS pulse may be issued. This field is only relevant when DDR_WRLVL_CNTL[WRLVL_EN] is set. 000b - 1 clock 001b - 2 clocks 010b - 4 clocks 011b - 8 clocks 100b - 16 clocks 101b - 32 clocks 110b - 64 clocks 111b - 128 clocks
23 —	Reserved
22-20 WRLVL_ODTEN	Write Leveling ODT Enable. ODT delay after margining mode is programmed (t_{WL_ODTEN}). Determines how many cycles to wait after margining mode has been programmed.until ODT may be asserted.This field is only relevant when DDR_WRLVL_CNTL[WRLVL_EN] is set. 000b - 1 clock 001b - 2 clocks 010b - 4 clocks 011b - 8 clocks 100b - 16 clocks 101b - 32 clocks 110b - 64 clocks 111b - 128 clocks
19 —	Reserved
18-16 WRLVL_DQSEN	Write Leveling DQS Enable. DQS/DQS_B delay after margining mode is programmed (t_{WL_DQSEN}). Determines how many cycles to wait after margining mode has been programmed.until DQS may be actively driven. This field is only relevant when DDR_WRLVL_CNTL[WRLVL_EN] is set. 000b - 1 clock 001b - 2 clocks 010b - 4 clocks 011b - 8 clocks 100b - 16 clocks

Table continues on the next page...

Field	Function
	101b - 32 clocks 110b - 64 clocks 111b - 128 clocks
15-12 WRLVL_SMPL	Write leveling sample time. Determines the number of cycles that must pass before the data signals are sampled after a DQS pulse during margining mode. This field should be programmed at least 6 cycles higher than t_{WLO} to allow enough time for propagation delay and sampling of the prime data bits. This field is only relevant when DDR_WRLVL_CNTL[WRLVL_EN] is set. 0000b - 32 clocks 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
11 —	Reserved
10-8 WRLVL_WLR	Write leveling repetition time. Determines the number of cycles that must pass between DQS pulses during write leveling. This field is only relevant when DDR_WRLVL_CNTL[WRLVL_EN] is set. 000b - 1 clock 001b - 2 clocks 010b - 4 clocks 011b - 8 clocks 100b - 16 clocks 101b - 32 clocks 110b - 64 clocks 111b - 128 clocks
7-5 —	Reserved
4-0 WRLVL_START	Write Leveling Start for DQS[0]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - 0 clock delay 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay

Field	Function
	01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay

12.4.23 DDR Self Refresh Counter (DDR_SR_CNTR)

12.4.23.1 Offset

Register	Offset
DDR_SR_CNTR	17Ch

12.4.23.2 Function

The DDR Self Refresh Counter register can be programmed to force the DDR controller to enter self refresh after a predefined period of idle time.

12.4.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												SR_IT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.23.4 Fields

Field	Function
31-20 —	Reserved
19-16 SR_IT	Self Refresh Idle Threshold. Defines the number of DRAM cycles that must pass while the DDR controller is idle before it will enter self refresh. Anytime a transaction is issued to the DDR controller, it will reset its internal counter. When a new transaction is received by the DDR controller, it will exit self refresh and reset its internal counter. If this field is zero, then the described power savings feature will be disabled. In addition, if a non-zero value is programmed into this field, then the DDR controller will exit self refresh anytime a transaction is issued to the DDR controller, regardless of the reason self refresh was initially entered. Patterns not shown are reserved. 0000b - Automatic self refresh entry disabled 0001b - 2^10 DRAM clocks 0010b - 2^12 DRAM clocks 0011b - 2^14 DRAM clocks 0100b - 2^16 DRAM clocks 0101b - 2^18 DRAM clocks 0110b - 2^20 DRAM clocks 0111b - 2^22 DRAM clocks 1000b - 2^24 DRAM clocks 1001b - 2^26 DRAM clocks 1010b - 2^28 DRAM clocks 1011b - 2^30 DRAM clocks
15-0 —	Reserved

12.4.24 DDR Register Control Words 1 (DDR_SDRAM_RCW_1)

12.4.24.1 Offset

Register	Offset
DDR_SDRAM_RCW_1	180h

12.4.24.2 Function

The DDR Register Control Word 1 register should be programmed with the intended values of the register control words if DDR_SDRAM_CFG[RCW_EN] is set. Each 4-bit field represents the value that will be placed on MA[3:0] in DDR4 mode during register control word writes.

12.4.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCW0				RCW1				RCW2				RCW3			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCW4				RCW5				RCW6				RCW7			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.24.4 Fields

Field	Function
31-28 RCW0	Register Control Word 0. Represents the value that will be used during writes to register control word 0.
27-24 RCW1	Register Control Word 1. Represents the value that will be used during writes to register control word 1.
23-20 RCW2	Register Control Word 2. Represents the value that will be used during writes to register control word 2.
19-16 RCW3	Register Control Word 3. Represents the value that will be used during writes to register control word 3.
15-12	Register Control Word 4.

Table continues on the next page...

Field	Function
RCW4	Represents the value that will be used during writes to register control word 4.
11-8 RCW5	Register Control Word 5. Represents the value that will be used during writes to register control word 5.
7-4 RCW6	Register Control Word 6. Represents the value that will be used during writes to register control word 6.
3-0 RCW7	Register Control Word 7. Represents the value that will be used during writes to register control word 7.

12.4.25 DDR Register Control Words 2 (DDR_SDRAM_RCW_2)

12.4.25.1 Offset

Register	Offset
DDR_SDRAM_RCW_2	184h

12.4.25.2 Function

The DDR Register Control Word 2 register should be programmed with the intended values of the register control words if DDR_SDRAM_CFG[RCW_EN] is set. Each 4-bit field represents the value that will be placed on MA[3:0] in DDR4 mode during register control word writes.

t

12.4.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCW8				RCW9				RCW10				RCW11			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCW12				RCW13				RCW14				RCW15			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.25.4 Fields

Field	Function
31-28 RCW8	Register Control Word 8. Represents the value that will be used during writes to register control word 8.
27-24 RCW9	Register Control Word 9. Represents the value that will be used during writes to register control word 9.
23-20 RCW10	Register Control Word 10. Represents the value that will be used during writes to register control word 10.
19-16 RCW11	Register Control Word 11. Represents the value that will be used during writes to register control word 11.
15-12 RCW12	Register Control Word 12. Represents the value that will be used during writes to register control word 12.
11-8 RCW13	Register Control Word 13. Represents the value that will be used during writes to register control word 13.
7-4 RCW14	Register Control Word 14. Represents the value that will be used during writes to register control word 14.
3-0 RCW15	Register Control Word 15. Represents the value that will be used during writes to register control word 15.

12.4.26 DDR write leveling control 2 (DDR_WRLVL_CNTL_2)

12.4.26.1 Offset

Register	Offset
DDR_WRLVL_CNTL_2	190h

12.4.26.2 Function

The DDR Write Leveling Control 2 register provides controls for write leveling. This register specifically defines the starting points for the individual data strobes.

Note: For each field WRLVL_START_n, a setting of 0000 is not recommended; it disables the per-byte write level start time selection. It is recommended to select proper individual delay for each byte lane based on board skews.

12.4.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			WRLVL_START_1					Reserved			WRLVL_START_2				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			WRLVL_START_3					Reserved			WRLVL_START_4				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.26.4 Fields

Field	Function
31-29 —	Reserved
28-24 WRLVL_START_1	Write leveling start time for DQS[1]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay

Table continues on the next page...

Field	Function
	10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay
23-21 —	Reserved
20-16 WRLVL_START_2	Write leveling start time for DQS[2]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay
15-13 —	Reserved
12-8 WRLVL_START_3	Write leveling start time for DQS[3]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay

Table continues on the next page...

Field	Function
	00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay
7-5 —	Reserved
4-0 WRLVL_START_4	Write leveling start time for DQS[4]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay

Field	Function
	10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay

12.4.27 DDR write leveling control 3 (DDR_WRLVL_CNTL_3)

12.4.27.1 Offset

Register	Offset
DDR_WRLVL_CNTL_3	194h

12.4.27.2 Function

The DDR Write Leveling Control 3 register provides controls for write leveling. This register specifically defines the starting points for the individual data strobes.

Note: For each field WRLVL_START_n, a setting of 0000 is not recommended; it disables the per-byte write level start time selection. It is recommended to select proper individual delay for each byte lane based on board skews.

12.4.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			WRLVL_START_5					Reserved			WRLVL_START_6				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			WRLVL_START_7					Reserved			WRLVL_START_8				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.27.4 Fields

Field	Function
31-29 —	Reserved
28-24 WRLVL_START_5	Write leveling start time for DQS[5]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay
23-21 —	Reserved
20-16 WRLVL_START_6	Write leveling start time for DQS[6]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay

Table continues on the next page...

Field	Function
	00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay
15-13 —	Reserved
12-8 WRLVL_START_7	Write leveling start time for DQS[7]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay

Table continues on the next page...

Field	Function
	11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay
7-5 —	Reserved
4-0 WRLVL_START_8	Write leveling start time for DQS[8]. Determines the value to use for the DQS_ADJUST for the first sample when write leveling is enabled. 00000b - Use value from DDR_WRLVL_CNTL[WRLVL_START] 00001b - 1/8 clock delay 00010b - 1/4 clock delay 00011b - 3/8 clock delay 00100b - 1/2 clock delay 00101b - 5/8 clock delay 00110b - 3/4 clock delay 00111b - 7/8 c'lock delay 01000b - 1 clock delay 01001b - 9/8 clock delay 01010b - 5/4 clock delay 01011b - 11/8 clock delay 01100b - 3/2 clock delay 01101b - 13/8 clock delay 01110b - 7/4 clock delay 01111b - 15/8 clock delay 10000b - 2 clock delay 10001b - 17/8 clock delay 10010b - 9/4 clock delay 10011b - 19/8 clock delay 10100b - 5/2 clock delay 10101b - 21/8 clock delay 10110b - 16/4 clock delay 10111b - 23/8 clock delay 11000b - 3 clock delay 11001b - 25/8 clock delay 11010b - 13/4 clock delay 11011b - 27/8 clock delay 11100b - 7/2 clock delay 11101b - 29/8 clock delay 11110b - 15/4 clock delay 11111b - 31/8 clock delay

12.4.28 DDR Register Control Words 3 (DDR_SDRAM_RCW_3)

12.4.28.1 Offset

Register	Offset
DDR_SDRAM_RCW_3	1A0h

12.4.28.2 Function

The DDR Register Control Word 3 register should be programmed with the intended values of the register control words if DDR_SDRAM_CFG[RCW_EN] is set. Each 8-bit field represents the value that will be placed on MA[7:0], during register control word writes. This register is only used for DDR4 mode.

12.4.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCW1X								RCW2X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCW3X								RCW4X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.28.4 Fields

Field	Function
31-24 RCW1X	Register Control Word 1X. Represents the value that will be placed on MA[7:0] during writes to register control word 1X for DDR4.
23-16 RCW2X	Register Control Word 2X. Represents the value that will be placed on MA[7:0] during writes to register control word 2X for DDR4.
15-8 RCW3X	Register Control Word 3X. Represents the value that will be placed on MA[7:0] during writes to register control word 3X for DDR4.
7-0 RCW4X	Register Control Word 4X. Represents the value that will be placed on MA[7:0] during writes to register control word 4X for DDR4.

12.4.29 DDR Register Control Words 4 (DDR_SDRAM_RCW_4)

12.4.29.1 Offset

Register	Offset
DDR_SDRAM_RCW_4	1A4h

12.4.29.2 Function

The DDR Register Control Word 4 register should be programmed with the intended values of the register control words if DDR_SDRAM_CFG[RCW_EN] is set. Each 8-bit field represents the value that will be placed on MA[7:0], during register control word writes. This register is only used for DDR4 mode.

12.4.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCW5X								RCW6X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCW7X								RCW8X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.29.4 Fields

Field	Function
31-24 RCW5X	Register Control Word 5X. Represents the value that will be placed on MA[7:0] during writes to register control word 5X for DDR4.
23-16 RCW6X	Register Control Word 6X. Represents the value that will be placed on MA[7:0] during writes to register control word 6X for DDR4.

Table continues on the next page...

Field	Function
15-8 RCW7X	Register Control Word 7X. Represents the value that will be placed on MA[7:0] during writes to register control word 7X for DDR4.
7-0 RCW8X	Register Control Word 8X. Represents the value that will be placed on MA[7:0] during writes to register control word 8X for DDR4.

12.4.30 DDR Register Control Words 5 (DDR_SDRAM_RCW_5)

12.4.30.1 Offset

Register	Offset
DDR_SDRAM_RCW_5	1A8h

12.4.30.2 Function

The DDR Register Control Word 5 register should be programmed with the intended values of the register control words if DDR_SDRAM_CFG[RCW_EN] is set. Each 8-bit field represents the value that will be placed on MA[7:0], during register control word writes. This register is only used for DDR4 mode.

t

12.4.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCW9X								RCW10X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCW11X								RCW12X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.30.4 Fields

Field	Function
31-24 RCW9X	Register Control Word 9X. Represents the value that will be placed on MA[7:0] during writes to register control word 9X for DDR4.
23-16 RCW10X	Register Control Word 10X. Represents the value that will be placed on MA[7:0] during writes to register control word 10X for DDR4.
15-8 RCW11X	Register Control Word 11X. Represents the value that will be placed on MA[7:0] during writes to register control word 11X for DDR4.
7-0 RCW12X	Register Control Word 12X. Represents the value that will be placed on MA[7:0] during writes to register control word 12X for DDR4.

12.4.31 DDR Register Control Words 6 (DDR_SDRAM_RCW_6)

12.4.31.1 Offset

Register	Offset
DDR_SDRAM_RCW_6	1ACh

12.4.31.2 Function

The DDR Register Control Word 6 register should be programmed with the intended values of the register control words if DDR_SDRAM_CFG[RCW_EN] is set. Each 8-bit field represents the value that will be placed on MA[7:0], during register control word writes. This register is only used for DDR4 mode.

12.4.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCW13X								RCW14X							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCW15X								SPARE_CNFG							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.31.4 Fields

Field	Function
31-24 RCW13X	Register Control Word 13X. Represents the value that will be placed on MA[7:0] during writes to register control word 13X for DDR4.
23-16 RCW14X	Register Control Word 14X. Represents the value that will be placed on MA[7:0] during writes to register control word 14X for DDR4.
15-8 RCW15X	Register Control Word 15X. Represents the value that will be placed on MA[7:0] during writes to register control word 15X for DDR4.
7-0 SPARE_CNFG	Spare Config Bits. This field is currently unused.

12.4.32 DDR SDRAM mode configuration 3 (DDR_SDRAM_MODE_3)

12.4.32.1 Offset

Register	Offset
DDR_SDRAM_MODE_3	200h

12.4.32.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers. This register is used specifically for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.32.4 Fields

Field	Function
31-16 ESDMODE	Extended SDRAM mode. Specifies the initial value loaded into the DDR SDRAM extended mode register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb of ESDMODE, which, in the big-endian convention shown here, corresponds to ESDMODE[15]. The msb of the SDRAM extended mode register value must be stored at ESDMODE[0]. The value programmed into this field is also used for writing MR1 during write leveling, although the bits specifically related to the write leveling scheme are handled automatically by the DDR controller. Even if DDR_SDRAM_CFG[BI] is set, this field is still used during write leveling. The write leveling enable bit should be cleared by software in this field.
15-0 SDMODE	SDRAM mode. Specifies the initial value loaded into the DDR SDRAM mode register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of SDMODE, which, in the big-endian convention shown here corresponds to SDMODE[15]. The msb of the SDRAM mode register value must be stored at SDMODE[0]. Because the memory controller forces SDMODE[7] to certain values depending on the state of the initialization sequence, (for resetting the SDRAM's DLL) the corresponding bits of this field are ignored by the memory controller. Note that SDMODE[7] is mapped to MA[8].

12.4.33 DDR SDRAM mode configuration 4 (DDR_SDRAM_MODE_4)

12.4.33.1 Offset

Register	Offset
DDR_SDRAM_MODE_4	204h

12.4.33.2 Function

The DDR SDRAM mode 4 configuration register sets the values loaded into the DDR's extended mode 2 and 3 registers. This register is used specifically for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.33.4 Fields

Field	Function
31-16 ESDMODE2	Extended SDRAM mode 2. Specifies the initial value loaded into the DDR SDRAM extended 2 mode register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range and meaning of legal values is specified by the DDR SDRAM manufacturer.

Table continues on the next page...

Field	Function
	When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE2, which, in the big-endian convention shown here, corresponds to ESDMODE2[15]. The msb of the SDRAM extended mode 2 register value must be stored at ESDMODE2[0].
15-0 ESDMODE3	Extended SDRAM mode 3. Specifies the initial value loaded into the DDR SDRAM extended 3 mode register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range of legal values of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of ESDMODE3, which, in the big-endian convention shown here, corresponds to ESDMODE3[15]. The msb of the SDRAM extended mode 3 register value must be stored at ESDMODE3[0].

12.4.34 DDR SDRAM mode configuration 5 (DDR_SDRAM_MODE_5)

12.4.34.1 Offset

Register	Offset
DDR_SDRAM_MODE_5	208h

12.4.34.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers. This register is used specifically for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.34.4 Fields

Field	Function
31-16 ESDMODE	Extended SDRAM mode. Specifies the initial value loaded into the DDR SDRAM extended mode register for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb of ESDMODE, which, in the big-endian convention, corresponds to ESDMODE[15]. The msb of the SDRAM extended mode register value must be stored at ESDMODE[0]. The value programmed into this field is also used for writing MR1 during write leveling, although the bits specifically related to the write leveling scheme are handled automatically by the DDR controller. Even if DDR_SDRAM_CFG[BI] is set, this field is still used during write leveling. The write leveling enable bit should be cleared by software in this field.
15-0 SDMODE	SDRAM mode. Specifies the initial value loaded into the DDR SDRAM mode register for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of SDMODE, which, in the big-endian convention, corresponds to SDMODE[15]. The msb of the SDRAM mode register value must be stored at SDMODE[0]. Because the memory controller forces SDMODE[7] to certain values depending on the state of the initialization sequence, (for resetting the SDRAM's DLL) the corresponding bits of this field are ignored by the memory controller. Note that SDMODE[7] is mapped to MA[8].

12.4.35 DDR SDRAM mode configuration 6 (DDR_SDRAM_MODE_6)

12.4.35.1 Offset

Register	Offset
DDR_SDRAM_MODE_6	20Ch

12.4.35.2 Function

The DDR SDRAM mode 6 configuration register sets the values loaded into the DDR's extended mode 2 and 3 registers. This register is used specifically for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.35.4 Fields

Field	Function
31-16 ESDMODE2	Extended SDRAM mode 2. Specifies the initial value loaded into the DDR SDRAM extended 2 mode register for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE2, which, in the big-endian convention, corresponds to ESDMODE2[15]. The msb of the SDRAM extended mode 2 register value must be stored at ESDMODE2[0].
15-0 ESDMODE3	Extended SDRAM mode 3. Specifies the initial value loaded into the DDR SDRAM extended 3 mode register for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range of legal values of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of ESDMODE3, which, in the big-endian convention, corresponds to ESDMODE3[15]. The msb of the SDRAM extended mode 3 register value must be stored at ESDMODE3[0].

12.4.36 DDR SDRAM mode configuration 7 (DDR_SDRAM_MODE_7)

12.4.36.1 Offset

Register	Offset
DDR_SDRAM_MODE_7	210h

12.4.36.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers. This register is used specifically for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDMODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.36.4 Fields

Field	Function
31-16 ESDMODE	Extended SDRAM mode. Specifies the initial value loaded into the DDR SDRAM extended mode register for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb of ESDMODE, which, in the big-endian convention, corresponds to ESDMODE[15]. The msb of the SDRAM extended mode register value must be stored at ESDMODE[0]. The value programmed into this field is also used for writing MR1 during write leveling, although the bits specifically related to the write leveling scheme are handled automatically by the DDR controller. Even if DDR_SDRAM_CFG[BI] is set, this field is still used during write leveling. The write leveling enable bit should be cleared by software in this field.
15-0 SDMODE	SDRAM mode. Specifies the initial value loaded into the DDR SDRAM mode register for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of SDMODE, which, in the big-endian convention, corresponds to SDMODE[15]. The msb of the SDRAM mode register value must be stored at SDMODE[0]. Because the memory controller forces

Field	Function
	SDMODE[7] to certain values depending on the state of the initialization sequence, (for resetting the SDRAM's DLL) the corresponding bits of this field are ignored by the memory controller. Note that SDMODE[7] is mapped to MA[8].

12.4.37 DDR SDRAM mode configuration 8 (DDR_SDRAM_MODE_8)

12.4.37.1 Offset

Register	Offset
DDR_SDRAM_MODE_8	214h

12.4.37.2 Function

The DDR SDRAM mode 8 configuration register sets the values loaded into the DDR's extended mode 2 and 3 registers. This register is used specifically for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.37.4 Fields

Field	Function
31-16 ESDMODE2	Extended SDRAM mode 2. Specifies the initial value loaded into the DDR SDRAM extended 2 mode register for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE2, which, in the big-endian convention, corresponds to ESDMODE2[15]. The msb of the SDRAM extended mode 2 register value must be stored at ESDMODE2[0].
15-0 ESDMODE3	Extended SDRAM mode 3. Specifies the initial value loaded into the DDR SDRAM extended 3 mode register for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. The range of legal values of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during DDR SDRAM initialization), MA[0] presents the lsb of ESDMODE3, which, in the big-endian convention, corresponds to ESDMODE3[15]. The msb of the SDRAM extended mode 3 register value must be stored at ESDMODE3[0].

12.4.38 DDR SDRAM mode configuration 9 (DDR_SDRAM_MODE_9)

12.4.38.1 Offset

Register	Offset
DDR_SDRAM_MODE_9	220h

12.4.38.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers.

12.4.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.38.4 Fields

Field	Function
31-16 ESDMODE4	Extended SDRAM mode 4. Specifies the initial value loaded into the DDR SDRAM MR4 register if using DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE4, which, in the big-endian convention, corresponds to ESDMODE4[15]. The msb of the SDRAM MR4 value must be stored at ESDMODE4[0].
15-0 ESDMODE5	Extended SDRAM mode 5. Specifies the initial value loaded into the DDR SDRAM MR5 register if using DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE5, which, in the big-endian convention, corresponds to ESDMODE5[15]. The msb of the SDRAM MR5 value must be stored at ESDMODE5[0].

12.4.39 DDR SDRAM mode configuration 10 (DDR_SDRAM_MODE_10)

12.4.39.1 Offset

Register	Offset
DDR_SDRAM_MODE_10	224h

12.4.39.2 Function

The DDR SDRAM mode 10 configuration register sets the values loaded into the DDR's extended mode 6and 7 registers.

12.4.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.39.4 Fields

Field	Function
31-16 ESDMODE6	Extended SDRAM mode 6. Specifies the initial value loaded into the DDR SDRAM MR6 register if using DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE6, which, in the big-endian convention, corresponds to ESDMODE6[15]. The msb of the SDRAM MR6 value must be stored at ESDMODE6[0].
15-0 ESDMODE7	Extended SDRAM mode 7. Specifies the initial value loaded into the DDR SDRAM MR7 register if using DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE7, which, in the big-endian convention, corresponds to ESDMODE7[15]. The msb of the SDRAM MR7 value must be stored at ESDMODE7[0].

12.4.40 DDR SDRAM mode configuration 11 (DDR_SDRAM_MODE_11)

12.4.40.1 Offset

Register	Offset
DDR_SDRAM_MODE_11	228h

12.4.40.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers. This register is used specifically for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.40.4 Fields

Field	Function
31-16 ESDMODE4	Extended SDRAM mode 4. Specifies the initial value loaded into the DDR SDRAM MR4 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE4, which, in the big-endian convention, corresponds to ESDMODE4[15]. The msb of the SDRAM MR4 value must be stored at ESDMODE4[0].
15-0 ESDMODE5	Extended SDRAM mode 5. Specifies the initial value loaded into the DDR SDRAM MR5 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto

Field	Function
	the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE5, which, in the big-endian convention, corresponds to ESDMODE5[15]. The msb of the SDRAM MR5 value must be stored at ESDMODE5[0].

12.4.41 DDR SDRAM mode configuration 12 (DDR_SDRAM_MODE_12)

12.4.41.1 Offset

Register	Offset
DDR_SDRAM_MODE_12	22Ch

12.4.41.2 Function

The DDR SDRAM mode 12 configuration register sets the values loaded into the DDR's extended mode 6 and 7 registers. This register is used specifically for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.41.4 Fields

Field	Function
31-16 ESDMODE6	Extended SDRAM mode 6. Specifies the initial value loaded into the DDR SDRAM MR6 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE6, which, in the big-endian convention, corresponds to ESDMODE6[15]. The msb of the SDRAM MR6 value must be stored at ESDMODE6[0].
15-0 ESDMODE7	Extended SDRAM mode 7. Specifies the initial value loaded into the DDR SDRAM MR7 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE7, which, in the big-endian convention, corresponds to ESDMODE7[15]. The msb of the SDRAM MR7 value must be stored at ESDMODE7[0].

12.4.42 DDR SDRAM mode configuration 13 (DDR_SDRAM_MODE_13)

12.4.42.1 Offset

Register	Offset
DDR_SDRAM_MODE_13	230h

12.4.42.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers. This register is used specifically for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.42.4 Fields

Field	Function
31-16 ESDMODE4	Extended SDRAM mode 4. Specifies the initial value loaded into the DDR SDRAM MR4 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE4, which, in the big-endian convention corresponds to ESDMODE4[15]. The msb of the SDRAM MR4 value must be stored at ESDMODE4[0].
15-0 ESDMODE5	Extended SDRAM mode 5. Specifies the initial value loaded into the DDR SDRAM MR5 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE5, which, in the big-endian convention corresponds to ESDMODE5[15]. The msb of the SDRAM MR5 value must be stored at ESDMODE5[0].

12.4.43 DDR SDRAM mode configuration 14 (DDR_SDRAM_MODE_14)

12.4.43.1 Offset

Register	Offset
DDR_SDRAM_MODE_14	234h

12.4.43.2 Function

The DDR SDRAM mode 14 configuration register sets the values loaded into the DDR's extended mode 6 and 7 registers. This register is used specifically for chip select 2 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.43.4 Fields

Field	Function
31-16 ESDMODE6	Extended SDRAM mode 6. Specifies the initial value loaded into the DDR SDRAM MR6 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE6, which, in the big-endian convention, corresponds to ESDMODE6[15]. The msb of the SDRAM MR6 value must be stored at ESDMODE6[0].
15-0 ESDMODE7	Extended SDRAM mode 7. Specifies the initial value loaded into the DDR SDRAM MR7 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE7, which, in the big-endian convention, corresponds to ESDMODE7[15]. The msb of the SDRAM MR7 value must be stored at ESDMODE7[0].

12.4.44 DDR SDRAM mode configuration 15 (DDR_SDRAM_MODE_15)

12.4.44.1 Offset

Register	Offset
DDR_SDRAM_MODE_15	238h

12.4.44.2 Function

The DDR SDRAM mode configuration register sets the values loaded into the DDR's mode registers. This register is used specifically for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.44.4 Fields

Field	Function
31-16 ESDMODE4	Extended SDRAM mode 4. Specifies the initial value loaded into the DDR SDRAM MR4 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE4, which, in the big-endian convention, corresponds to ESDMODE4[15]. The msb of the SDRAM MR4 value must be stored at ESDMODE4[0].

Table continues on the next page...

Field	Function
15-0 ESDMODE5	Extended SDRAM mode 5. Specifies the initial value loaded into the DDR SDRAM MR5 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE5, which, in the big-endian convention, corresponds to ESDMODE5[15]. The msb of the SDRAM MR5 value must be stored at ESDMODE5[0].

12.4.45 DDR SDRAM mode configuration 16 (DDR_SDRAM_MODE_16)

12.4.45.1 Offset

Register	Offset
DDR_SDRAM_MODE_16	23Ch

12.4.45.2 Function

The DDR SDRAM mode 16 configuration register sets the values loaded into the DDR's extended mode 6 and 7 registers. This register is used specifically for chip select 3 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set.

12.4.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ESDMODE6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESDMODE7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.45.4 Fields

Field	Function
31-16 ESDMODE6	Extended SDRAM mode 6. Specifies the initial value loaded into the DDR SDRAM MR6 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE6, which, in the big-endian convention, corresponds to ESDMODE6[15]. The msb of the SDRAM MR6 value must be stored at ESDMODE6[0].
15-0 ESDMODE7	Extended SDRAM mode 7. Specifies the initial value loaded into the DDR SDRAM MR7 register for chip select 1 if DDR_SDRAM_CFG_2[UNQ_MRS_EN] is set. This is only used with DDR4 memories. The range and meaning of legal values is specified by the DDR SDRAM manufacturer. When this value is driven onto the address bus (during the DDR SDRAM initialization sequence), MA[0] presents the lsb bit of ESDMODE7, which, in the big-endian convention, corresponds to ESDMODE7[15]. The msb of the SDRAM MR7 value must be stored at ESDMODE7[0].

12.4.46 DDR SDRAM timing configuration 8 (TIMING_CFG_8)

12.4.46.1 Offset

Register	Offset
TIMING_CFG_8	250h

12.4.46.2 Function

The DDR SDRAM timing configuration 8 register provides additional timing fields required to support DDR4 memories.

12.4.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RWT_BG				WRT_BG				RRT_BG				WWT_BG			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ACTTOACT_BG				WRTORD_BG				EXT_WRTORD_BG	Reserved		PRE_ALL_REC				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.46.4 Fields

Field	Function
31-28 RWT_BG	Read-to-write turnaround for same chip select and same bank group. Specifies how many cycles will be added between a read to write turnaround for transactions to the same bank group. If a value of 0000 is chosen, then the DDR controller will use the value used for transactions to different chip selects, as defined in TIMING_CFG_0[RWT]. This field can be used to improve performance when operating in burst-chop mode by forcing transactions to the same chip select to use extra cycles, while transaction to different chip selects can utilize the tri-state time on the DRAM interface. Regardless of the value that is set in this field, the value defined by TIMING_CFG_0[RWT] will also be met before issuing a write command. 0000b - Default 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
27-24 WRT_BG	Write-to-read turnaround for same chip select and same bank group.

Table continues on the next page...

Field	Function
	Specifies how many cycles will be added between a write to read turnaround for transactions to the same bank group. If a value of 0000 is chosen, then the DDR controller will use the value used for transactions to different chip selects, as defined in TIMING_CFG_0[WRT]. This field can be used to improve performance when operating in burst-chop mode by forcing transactions to the same chip select to use extra cycles, while transaction to different chip selects can utilize the tri-state time on the DRAM interface. Regardless of the value that is set in this field, the value defined by TIMING_CFG_0[WRT] will also be met before issuing a read command. 0000b - Default 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
23-20 RRT_BG	Read-to-read turnaround for same chip select and same bank group. Specifies how many cycles will be added between reads to the same bank group. If a value of 0000 is chosen, then 2 cycles will be required between read commands to the same chip select if 4-beat bursts are used (4 cycles will be required if 8-beat bursts are used). Note that fixed 4-beat bursts are not supported. However, this field may be used to add extra cycles when burst-chop mode is used, and the DDR controller must wait 4 cycles for read-to-read transactions to the same chip select. 0000b - BL/2 clocks 0001b - BL/2 + 1 clock 0010b - BL/2 + 2 clocks 0011b - BL/2 + 3 clocks 0100b - BL/2 + 4 clocks 0101b - BL/2 + 5 clocks 0110b - BL/2 + 6 clocks 0111b - BL/2 + 7 clocks 1000b - BL/2 + 8 clocks 1001b - BL/2 + 9 clocks 1010b - BL/2 + 10 clocks 1011b - BL/2 + 11 clocks 1100b - BL/2 + 12 clocks 1101b - BL/2 + 13 clocks 1110b - BL/2 + 14 clocks 1111b - BL/2 + 15 clocks
19-16 WWT_BG	Write-to-write turnaround for same chip select and same bank group. Specifies how many cycles will be added between writes to the same bank group. If a value of 0000 is chosen, then 2 cycles will be required between write commands to the same bank group if 4-beat bursts are used (4 cycles will be required if 8-beat bursts are used). Note that fixed 4-beat bursts are not supported. However, this field may be used to add extra cycles when burst-chop mode is used, and the DDR controller must wait 4 cycles for write-to-write transactions to the same chip select. 0000b - BL/2 clocks 0001b - BL/2 + 1 clock 0010b - BL/2 + 2 clocks

Table continues on the next page...

Field	Function
	0011b - BL/2 + 3 clocks 0100b - BL/2 + 4 clocks 0101b - BL/2 + 5 clocks 0110b - BL/2 + 6 clocks 0111b - BL/2 + 7 clocks 1000b - BL/2 + 8 clocks 1001b - BL/2 + 9 clocks 1010b - BL/2 + 10 clocks 1011b - BL/2 + 11 clocks 1100b - BL/2 + 12 clocks 1101b - BL/2 + 13 clocks 1110b - BL/2 + 14 clocks 1111b - BL/2 + 15 clocks
15-12 ACTTOACT_BG	Activate-To-Activate Same Bank Group. Activate-to-activate interval for the same bank group(t_{RRD_L}). Number of clock cycles from an activate command until another activate command is allowed for a different logical bank in the same physical bank (chip select). Bank groups are only used for DDR4. 0000b - ACTTOACT_BG is unused 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks
11-8 WRTORD_BG	Write-To-Read Same Bank Group. Last write data pair to read command issue interval for the same bank group(t_{WTR_L}). Number of clock cycles between the last write data pair and the subsequent read command to the same physical bank. If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this field needs to be programmed to (t_{WTR} + 2 cycles). Bank groups are only used for DDR4. 0000b - WRTORD_BG timing is unused 0001b - 1 clock 0010b - 2 clocks 0011b - 3 clocks 0100b - 4 clocks 0101b - 5 clocks 0110b - 6 clocks 0111b - 7 clocks 1000b - 8 clocks 1001b - 9 clocks 1010b - 10 clocks 1011b - 11 clocks 1100b - 12 clocks 1101b - 13 clocks 1110b - 14 clocks 1111b - 15 clocks

Table continues on the next page...

Field	Function
7 EXT_WRTORD_BG	Extended Write-To-Read Same Bank Group. Extended bit field for last write data pair to read command issue interval for the same bank group(t_{WTR_L}). This field is concatenated as {EXT_WRTORD_BG, WRTORD_BG} to determine the total number of clock cycles between last write data pair and the subsequent read command to the same physical bank. 0b - Do not add additional cycles to the value programmed at WRTORD_BG 1b - 16 clock cycles are added to the value programmed at WRTORD_BG
6-5 —	Reserved
4-0 PRE_ALL_REC	Precharge all-to-activate interval. Determines the number of clock cycles from a precharge all command until an activate or refresh command is allowed. 00000b - Use TIMING_CFG_1[PRETOACT] 00001b - 1 clock 00010b - 2 clocks 00011b - 3 clocks 00100b - 4 clocks 00101b - 5 clocks 00110b - 6 clocks 00111b - 7 clocks 01000b - 8 clocks 01001b - 9 clocks 01010b - 10 clocks 01011b - 11 clocks 01100b - 12 clocks 01101b - 13 clocks 01110b - 14 clocks 01111b - 15 clocks 10000b - 16 clocks 10001b - 17 clocks 10010b - 18 clocks 10011b - 19 clocks 10100b - 20 clocks 10101b - 21 clocks 10110b - 22 clocks 10111b - 23 clocks 11000b - 24 clocks 11001b - 25 clocks 11010b - 26 clocks 11011b - 27 clocks 11100b - 28 clocks 11101b - 29 clocks 11110b - 30 clocks 11111b - 31 clocks

12.4.47 DDR SDRAM control configuration 3 (DDR_SDRAM_CFG_3)

12.4.47.1 Offset

Register	Offset
DDR_SDRAM_CFG_3	260h

12.4.47.2 Function

The DDR SDRAM control configuration register 3 provides more control configuration for the DDR controller.

12.4.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	DDRC_RST	ECC_FIX_EN	Reserved		ECC_SCRUB_INT				Reserved				Reserved	Reserved	WR_PRE	RD_PRE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		DM_CFG		Reserved		REF_MODE		Reserved	Reserved	Reserved		Reserved	Reserved	Reserved	DIS_MRS_PAR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.47.4 Fields

Field	Function
31 DDRC_RST	DDR controller reset. This bit should be set by software if the DDR controller should be reset. This will reset all state machines and internal FIFOs. This bit will not reset the DDR controller's configuration registers. Before setting this bit, DDR_SDRAM_CFG[MEM_EN] should be cleared. Note that this bit is self-clearing. Software should poll for this bit to clear before re-enabling the DDR controllerr. 0b - DDR controller is operating in normal mode.

Table continues on the next page...

Field	Function
	1b - DDR controller is reset.
30 ECC_FIX_EN	ECC fixing enable. The DDR controller supports ECC fixing in memory. In this mode, the DDR controller will automatically fix any detected single-bit errors by issuing a new transaction to read the address with the failing bit, correcting the bit, and writing the data back to memory. The single-bit error will still be counted in the ERR_SBE register for this case, but the controller will automatically fix the error. Note that during the 'read back', the single-bit error will not be double counted in the ERR_SBE register. In addition, the DDR controller will periodically issue a read to memory at the interval defined by ECC_SCRUB_INT. If a single-bit error is detected during a periodic read, it will be fixed. In this case, the error will be reported as an SSBE in the ERR_SBE register. If a multi-bit eror is detected, then it will be reported in the ERR_DETECT register. Also note that if a subsequent single-bit error is detected at the same address while a first error is being fixed, then the second error will not be reported. Also, after a first SBE is detected, no other SBEs will be fixed until the first SBE has been fixed in memory.This bit should only be set if DDR_SDRAM_CFG[ECC_EN] is also set. **NOTE:** Scrubbing cannot be enabled until after the controller has cleared DDR_SDRAM_CFG_2[D_INIT]. 0b - ECC scrubbing is disabled. 1b - ECC scrubbing is enabled.
29-28 —	Reserved
27-24 ECC_SCRUB_INT	ECC scrubbing interval. This field defines how frequent the DDR controller will inject reads to test ECC if ECC_FIX_EN is set. Note that reads will only be injected immediately after a refresh sequence. If a single-bit error is detected, then the controller will fix the error in memory. If a multi-bit error is detected, then the controller will report the error in the ERR_DETECT register. When issuing reads, the DDR controller will move sequentially throughout the DRAM address space by automatically incrementing an internal address counter. Note that the maximum amount memory supported would be 4 ranks of 16 Gbit x8 devices utilizing a 64-bit bus, which provides a total of 64 GBytes of memory. Since 64-bytes of data are scrubbed for each read, this will require 1G total reads to scrub the entire memory contents. Assuming a refresh interval of 7.8 microseconds, this will take approximately 2.35 hours to cycle through the entire memory array. If less memory is used, then this time will take less. The scrubbing interval can be increased to improve performance so the reads are not issued as frequently. **NOTE:** Scrubbing cannot be enabled until after the controller has cleared DDR_SDRAM_CFG_2[D_INIT]. 0000b - Periodic reads for ECC scrubbing will not be issued. 0001b - A read will be issued every refresh sequence. 0010b - A read will be issued every 2 refresh sequences. 0011b - A read will be issued every 4 refresh sequences. 0100b - A read will be issued every 8 refresh sequences. 0101b - A read will be issued every 16 refresh sequences. 0110b - A read will be issued every 32 refresh sequences. 0111b - A read will be issued every 64 refresh sequences. 1000b - A read will be issued every 128 refresh sequences. 1001b - A read will be issued every 256 refresh sequences. 1010b - A read will be issued every 512 refresh sequences. 1011b - A read will be issued every 1024 refresh sequences. 1100b - A read will be issued every 2048 refresh sequences. 1101b - A read will be issued every 4096 refresh sequences. 1110b - A read will be issued every 8192 refresh sequences. 1111b - A read will be issued every 16,384 refresh sequences.
23-20 —	Reserved

Table continues on the next page...

Field	Function
19 —	Reserved
18 —	Reserved
17 WR_PRE	Write preamble. The DDR controller supports a programmable write preamble for DDR4 memories. Extra turnaround cycles will need to be programmed if using 2 tCK preamble. In addition, either Rtt should be parked, or the system needs to tolerate enabling the termination at the DRAM at the same time the preamble is driven active when using a 2 tCK write preamble. **NOTE:** When 2-cycle preambles are used, TIMING_CFG_0[WWT,RRT,WRT,RWT] must all be set greater than 0. 0b - A 1t_{CK} preamble is used 1b - A 2 t_{CK} preamble is used
16 RD_PRE	Read preamble. The DDR controller supports a programmable read preamble for DDR4 memories. Extra turnaround cycles will need to be programmed if using 2 tCK preamble. **NOTE:** When 2-cycle preambles are used, TIMING_CFG_0[WWT,RRT,WRT,RWT] must all be set greater than 0. 0b - A 1t_{CK} preamble is used 1b - A 2 t_{CK} preamble is used
15-14 —	Reserved
13-12 DM_CFG	Data mask config. Determines how the MDM pins will be utilized for the DDR controller. 00b - Normal data masks will be used based on DDR_SDRAM_CFG[SDRAM_TYPE]. 01b - Reserved. 10b - Data bus inversion (DBI) will be used by utilizing the MDM pins. 11b - Neither data mask or data bus inversion will be used. The MDM pins will be held high for DDR4 by the controller.
11-10 —	Reserved
9-8 REF_MODE	Refresh Mode. This field allows programming for fine granularity refresh defined by DDR4 specifications. The fine granularity refresh mode is supported by allowing full programmability of the DDR_SDRAM_CFG_2[NUM_PR] and DDR_SDRAM_INTERVAL[REFINT] timings. However, this field is required to notify the controller how many refreshes to issue when exitting self refresh. If fine granularity refresh mode will be used, then it must enabled after the DDR controller has been enabled, per the DRAM requirement to use 1x mode during initial MRS commands. Therefore, software must issue the required MRS commands via DDR_SDRAM_MD_CNTL register to allow the DRAM to enter fine granularity refresh mode. On-the-fly fine granularity refresh mode is not supported. 00b - Fine granularity refresh disabled. 01b - 2x fine granularity refresh mode. 10b - 4x fine granularity refresh mode. 11b - Reserved
7 —	Reserved
6	Reserved

Table continues on the next page...

Field	Function
—	
5-4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 DIS_MRS_PAR	Disable MRS on Parity Error. This bit controls the automatic MR command that may be issued upon a parity error. This bit should be set if using parity. If using parity with discrete devices, MR5 should be set to automatically re-enable parity checking after an error. If registered DIMMs are used, then RC0E of the register should be programmed to automatically re-enable parity checking after an error. 0b - Issue an MR command to clear the parity error in the DRAM when a parity error is observed. 1b - Do not issue an MR command to clear the parity error on the DRAM. Software is responsible for clearing parity errors if required via the DDR_SDRAM_MD_CNTL register.

12.4.48 DQ mapping register 0 (DDR_DQ_MAP0)

12.4.48.1 Offset

Register	Offset
DDR_DQ_MAP0	400h

12.4.48.2 Function

DQ_MAP0-DQ_MAP3 should be programmed to reflect how the DQ bits are swizzled on the DIMMs. Note that if a board has further swizzling within the nibbles of the DQ signals, then that needs to be accounted for when programming these fields. In addition, if multiple modules are used in the system, they must use the same raw card design (such that the swizzling is identical between DIMMs).

For each of the DQ mapping fields, the encodings are shown in the accompanying table. These bitfields apply to nibbles of data; therefore, they must normally be set in pairs. (For example, DQ_0_3 and DQ_4_7 must be set together.)

Table 12-6. Bitfield settings for each byte lane

Setting	DQ0	DQ1	DQ2	DQ3
	DQ4	DQ5	DQ6	DQ7
0x00	0	1	2	3
0x01	0	1	2	3
0x02	0	1	3	2
0x03	0	2	1	3
0x04	0	2	3	1
0x05	0	3	1	2
0x06	0	3	2	1
0x07	1	0	2	3
0x08	1	0	3	2
0x09	1	2	0	3
0x0a	1	2	3	0
0x0b	1	3	0	2
0x0c	1	3	2	0
0x0d	2	0	1	3
0x0e	2	0	3	1
0x0f	2	1	0	3
0x10	2	1	3	0
0x11	2	3	0	1
0x12	2	3	1	0
0x13	3	0	1	2
0x14	3	0	2	1
0x15	3	1	0	2
0x16	3	1	2	0
0x17	3	2	0	1
0x18	3	2	1	0
0x21	4	5	6	7
0x22	4	5	7	6
0x23	4	6	5	7
0x24	4	6	7	5
0x25	4	7	5	6
0x26	4	7	6	5
0x27	5	4	6	7
0x28	5	4	7	6
0x29	5	6	4	7
0x2a	5	6	7	4

Table continues on the next page...

Table 12-6. Bitfield settings for each byte lane (continued)

Setting	DQ0	DQ1	DQ2	DQ3
	DQ4	DQ5	DQ6	DQ7
0x2b	5	7	4	6
0x2c	5	7	6	4
0x2d	6	4	5	7
0x2e	6	4	7	5
0x2f	6	5	4	7
0x30	6	5	7	4
0x31	6	7	4	5
0x32	6	7	5	4
0x33	7	4	5	6
0x34	7	4	6	5
0x35	7	5	4	6
0x36	7	5	6	4
0x37	7	6	4	5
0x38	7	6	5	4

12.4.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DQ_0_3						DQ_4_7						DQ_8_11			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DQ_8_11		DQ_12_15						DQ_16_19						RESERVED	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.48.4 Fields

Field	Function
31-26 DQ_0_3	DQ[0:3] Mapping. See Table 12-6 and the text describing this register for settings.
25-20	DQ[4:7] Mapping.

Table continues on the next page...

Field	Function
DQ_4_7	See Table 12-6 and the text describing this register for settings.
19-14 DQ_8_11	DQ[8:11] Mapping. See Table 12-6 and the text describing this register for settings.
13-8 DQ_12_15	DQ[12:15] Mapping. See Table 12-6 and the text describing this register for settings.
7-2 DQ_16_19	DQ[16:19] Mapping. See Table 12-6 and the text describing this register for settings.
1-0 RESERVED	Reserved These bits are writeable, but they are unused.

12.4.49 DQ mapping register 1 (DDR_DQ_MAP1)

12.4.49.1 Offset

Register	Offset
DDR_DQ_MAP1	404h

12.4.49.2 Function

DQ_MAP0-DQ_MAP3 should be programmed to reflect how the DQ bits are swizzled on the DIMMs. See further details at DQ mapping register 0 (DDR_DQ_MAP0).

12.4.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DQ_20_23						DQ_24_27						DQ_28_31			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DQ_28_31		DQ_32_35						DQ_36_39						RESERVED	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.49.4 Fields

Field	Function
31-26 DQ_20_23	DQ[20:23] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
25-20 DQ_24_27	DQ[24:27] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
19-14 DQ_28_31	DQ[28:31] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
13-8 DQ_32_35	DQ[32:35] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
7-2 DQ_36_39	DQ[36:39] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
1-0 RESERVED	Reserved These bits are writeable, but they are unused.

12.4.50 DQ mapping register 2 (DDR_DQ_MAP2)

12.4.50.1 Offset

Register	Offset
DDR_DQ_MAP2	408h

12.4.50.2 Function

DQ_MAP0-DQ_MAP3 should be programmed to reflect how the DQ bits are swizzled on the DIMMs. See further details at DQ mapping register 0 (DDR_DQ_MAP0).

12.4.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DQ_40_43						DQ_44_47						DQ_48_51			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DQ_48_51		DQ_52_55						DQ_56_59						RESERVED	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.50.4 Fields

Field	Function
31-26 DQ_40_43	DQ[40:43] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
25-20 DQ_44_47	DQ[44:47] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
19-14 DQ_48_51	DQ[48:51] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
13-8 DQ_52_55	DQ[52:55] Maping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
7-2 DQ_56_59	DQ[56:59] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
1-0 RESERVED	Reserved These bits are writeable, but they are unused.

12.4.51 DQ mapping register 3 (DDR_DQ_MAP3)

12.4.51.1 Offset

Register	Offset
DDR_DQ_MAP3	40Ch

12.4.51.2 Function

DQ_MAP0-DQ_MAP3 should be programmed to reflect how the DQ bits are swizzled on the DIMMs. See further details at DQ mapping register 0 (DDR_DQ_MAP0).

12.4.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DQ_60_63						ECC_0_3						ECC_4_7			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECC_4_7		Reserved													ORS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.51.4 Fields

Field	Function
31-26 DQ_60_63	DQ[60:63] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
25-20 ECC_0_3	ECC[0:3] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
19-14 ECC_4_7	ECC[4:7] Mapping. See the table and description in DQ mapping register 0 (DDR_DQ_MAP0) for settings.
13-1 —	Reserved
0 ORS	Odd rank swizzle. If this bit is cleared, then CS1 and CS3 follow the same swizzling as CS0 and CS2. If this bit is set, then CS1 and CS3 are further swizzled by swapping bits 0 vs 1, 2 vs 3, 4 vs 5, and 6 vs 7 within each byte.

12.4.52 DDR Debug Status Register 1 (DDRDSR_1)

12.4.52.1 Offset

Register	Offset
DDRDSR_1	B20h

12.4.52.2 Function

The DDRDSR_1 register contains the current settings of the driver impedance for the command/control and data.

12.4.52.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CZ	Reserved							DZ	Reserved						
W																
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

12.4.52.4 Fields

Field	Function
31-16 —	Reserved
15 CZ	Command Impedance. Current setting of driver command impedance
14-8 —	Reserved
7 DZ	Data Impedance. Current setting of driver data impedance
6-0 —	Reserved

12.4.53 DDR Debug Status Register 2 (DDRDSR_2)

12.4.53.1 Offset

Register	Offset
DDRDSR_2	B24h

12.4.53.2 Function

The DDRDSR_2 register contains the current settings of the driver impedance for the DDR drivers for clocks.

12.4.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CLKZ	Reserved														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														RPD_ST	RPD_END
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.53.4 Fields

Field	Function
31 CLKZ	Clock Impedance. Current setting of driver clock impedance

Table continues on the next page...

Field	Function
30-2 —	Reserved
1 RPD_ST	Rapid clear of memory start. See DDR Rapid Clear of Memory for more information. 0b - The rapid clear of memory function has not been started. 1b - The rapid clear of memory function has started. During the rapid clear of memory, writes to the DDR memory mapped registers will not affect the register contents. This bit is cleared by software writing a 1.
0 RPD_END	Rapid clear of memory end. See DDR Rapid Clear of Memory for more information. 0b - The rapid clear of memory function has not ended. 1b - The rapid clear of memory function has completed. This bit is cleared by software writing a 1.

12.4.54 DDR Control Driver Register 1 (DDRCDR_1)

12.4.54.1 Offset

Register	Offset
DDRCDR_1	B28h

12.4.54.2 Function

DDRCDR_1 sets the hardware DDR driver calibration enable, the ODT termination value for IOs, and the software override enables for address/command and data bus signals. The combined DDRCDR_1[ODT] and DDRCDR_2[ODT] set the on-die-termination values in the memory controller for the data bus signals. Hardware DDR driver calibration must be enabled by setting DDRCDR_1[DHC_EN]. MDIC pins are required to be connected to proper calibration resistors. The proper value and connection of the MDIC resistor are specified in the corresponding device data sheet document.

The global half-strength override field DDR_SDRAM_CFG[HSE] or software overrides in DDRCDR_1 or DDRCDR_2 registers determine whether the DDR drivers calibrate to full-strength or half-strength. The software overrides in DDRCDR_1 or DDRCDR_2 registers take precedence over the DDR_SDRAM_CFG[HSE] setting.

NOTE

All driver calibration related registers should be set 500 µs before the DDR controller is enabled (that is, before DDR_SDRAM_CFG[MEM_EN] is set).

This table lists the valid impedance override values. Note that the drivers may be calibrated to either full-strength or half-strength.

Table 12-7. Valid Impedance Override Values

Driver impedance	Impedance Override Value
Highest	0
Lowest	1

12.4.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	DHC_EN	V0PT9_EN	Reserved										ODT		DSO_C_EN	DSO_D_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DSO_CZ	Reserved							DSO_DZ	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.54.4 Fields

Field	Function
31 DHC_EN	Driver Hardware Calibration Enable. DDR driver hardware compensation enable. If this bit is set to enable automatic hardware compensation, then 200 microseconds should pass after setting this bit, and before setting DDR_SDRAM_CFG[MEM_EN].
30 V0PT9_EN	Enable if using 0.9V Vdd
29-20 —	Reserved These bits are writeable, but they are unused.

Table continues on the next page...

Field	Function
19-18 ODT	On-Die-Termination. ODT termination value for IOs. This is combined with DDRCDR_2[ODT] to determine the termination value. Below is the termination based on concatenating these two fields. Note that the order of concatenation is from left to right DDRCDR_1[ODT], DDRCDR_2[ODT] Bits [0:0] of the following values are defined at DDRCDR_2.ODT. 000b - Termination is disabled 001b - 100 Ω (DDR4) 010b - 120 Ω (DDR4) 011b - 80 Ω (DDR4) 100b - 60 Ω (DDR4) 101b - 40 Ω (DDR4) 110b - 50 Ω (DDR4) 111b - 30 Ω (DDR4)
17 DSO_C_EN	Override Enable for Command Impedance. Driver software override enable for address/command
16 DSO_D_EN	Override Enable for Data Impedance. Driver software override enable for data
15 DSO_CZ	Override for Command Impedance. DDR driver software command impedance override
14-8 —	Reserved These bits are writeable, but they are unused.
7 DSO_DZ	Override For Data Impedance. Driver software data impedance override
6-0 —	Reserved These bits are writeable, but they are unused.

12.4.55 DDR Control Driver Register 2 (DDRCDR_2)

12.4.55.1 Offset

Register	Offset
DDRCDR_2	B2Ch

12.4.55.2 Function

The DDRCDR_2 sets the driver software override enable for clocks, and the DDR clocks driver P/N impedance.

12.4.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DSO_CLK_EN	Reserved			DSO_CLKZ	Reserved							Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VREF_OVRD_EN	Reserved	VREF_OVRD_VAL						VREF_TRAIN_EN	VREF_DRAM_RANGE	Reserved					ODT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.55.4 Fields

Field	Function
31 DSO_CLK_EN	Override Enable for Clock Impedance. Driver software override enable for clocks
30-28 —	Reserved
27 DSO_CLKZ	Override Value for Clock Impedance. Driver software clocks impedance override
26-20 —	Reserved
19-16 —	Reserved
15	Internal VRef Override Enable.

Table continues on the next page...

Field	Function
VREF_OVRD_EN	If DDR4 mode is used, then an internal VRef is generated an used for the data bus. By default, this internal VRef value will be trained. However, the training can be disabled by this override. 0b - Internal VRef will be trained if DDR4 mode is used. 1b - Internal VRef will be defined by the VREF_OVRD_VAL field of this register.
14 —	Reserved
13-8 VREF_OVRD_VAL	Internal VRef Override Value. This field defines the override value to use for the internal VRef if VREF_OVRD_EN is set. The values below are targetted percentages of GVdd for the internal VRef. However, it is highly recommended to leave the VREF_OVRD_EN bit cleared to allow VRef to be trained for the best read margins. 000000b - 37% 000001b - 38% 000010b - 39% 111110b - 99% 111111b - 100%
7 VREF_TRAIN_EN	DRAM VRef Train Enable In addition to the internal VRef training, the DRAM's VRef may also be trained. This bit can be set to automatically enable DRAM VRef training. 0b - DRAM VRef will not be trained. 1b - DRAM VRef will be trained.
6 VREF_DRAM_RANGE	VRef DRAM Range. If using DRAM VRef training, this bit will specify whether Range 1 or Range 2 will be trained. 0b - DRAM VRef Range 1 will be used for training. 1b - DRAM VRef Range 2 will be used for training.
5-1 —	Reserved
0 ODT	On-Die Termination ODT termination value for IOs. This is combined with DDRCDR_1[ODT] to determine the termination value. Below is the termination based on concatenating these two fields. Note that the order of concatenation is (from left to right) DDRCDR_1[ODT], DDRCDR_2[ODT] Bits [18:19] of the following values are defined at DDRCDR_1.ODT. 000b - Termination is disabled 001b - 100 Ω (DDR4) 010b - 120 Ω (DDR4) 011b - 80 Ω (DDR4) 100b - 60 Ω (DDR4) 101b - 40 Ω (DDR4) 110b - 50 Ω (DDR4) 111b - 30 Ω (DDR4)

12.4.56 DDR IP block revision 1 (DDR_IP_REV1)

12.4.56.1 Offset

Register	Offset
DDR_IP_REV1	BF8h

12.4.56.2 Function

The DDR IP block revision 1 register provides read-only fields with the IP block ID, along with major and minor revision information.

12.4.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IP_MJ								IP_MN							
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0

12.4.56.4 Fields

Field	Function
31-16 IP_ID	IP block ID. For the DDR controller, this value is 0x0002.
15-8 IP_MJ	Major revision. This is currently set to 0x05.
7-0 IP_MN	Minor revision. This is currently set to 8'd2.

12.4.57 DDR IP block revision 2 (DDR_IP_REV2)

12.4.57.1 Offset

Register	Offset
DDR_IP_REV2	BFCh

12.4.57.2 Function

The DDR IP block revision 2 register provides read-only fields with the IP block integration and configuration options.

12.4.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								IP_INT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								IP_CFG							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.57.4 Fields

Field	Function
31-24 —	Reserved
23-16 IP_INT	IP Block Integration Options.
15-8 —	Reserved
7-0 IP_CFG	IP Block Configuration Options.

12.4.58 DDR Memory Test Control Register (DDR_MTCR)

12.4.58.1 Offset

Register	Offset
DDR_MTCR	D00h

12.4.58.2 Function

The DDR Memory Test Control Register provides the enable and controls for an automatic memory test.

12.4.58.3 Diagram

12.4.58.4 Fields

Field	Function
31 MT_EN	Memory Test Enable. This bit can be set by software to enable the memory test. Based on the value of MT_TYP, the controller will either issue writes only, reads only, or it will issue writes and reads.The memory controller will issue transactions throughout all of memory, as defined by the CS*n*_CONFIG and CSn_BNDS registers. The memory controller will check the data during this test. In addition, the ERR_DETECT register should be

Table continues on the next page...

Field	Function
	read by software if ECC is enabled after the memory test to ensure there were no ECC errors. If an ECC error is detected, the error capture registers will hold the address, data, and attributes captured for the first fail. If there is no ECC error and the test failed (for a data miscompare), then the capture registers will hold information for the transaction that caused the first data miscompare. Note that transactions with ECC errors will take priority in the capture registers. Hardware will clear MT_EN after the memory test is complete. This bit can be set before DDR_SDRAM_CFG[MEM_EN] is set, or it can be set again after the memory controller has been enabled. 0b - Memory test has been disabled. 1b - Memory test has been enabled. Hardware will clear this bit when it is complete.
30-26 —	Reserved
25-24 MT_TYP	Memory Test Type. This field will determine if the memory test will issue writes only, reads only, or both writes and reads. Note that the 'read only' test should not be used unless memory has already been initialized. 00b - Memory test will issue writes and reads. 01b - Memory test will issue writes only. 10b - Memory test will issue reads only. 11b - Reserved
23-20 —	Reserved
19-16 MT_TRNARND	Memory Test Turnaround. This field determines how many writes will be issued during the memory test before the reads to the same addresses will be issued. This can be used to allow longer streams of writes/reads, and it can be used to test the write->read and read->write turnarounds in a stressful manner. This field is only relevant if MT_TYP is set to 2'b00. 0000b - Entire memory will be written before read transactions are issued. 0001b - Total write/read streams will be 1 transaction each. 0010b - Total write/read streams will be 2 transactions each. 0011b - Total write/read streams will be 4 transactions each. 0100-1111b - Reserved
15-10 —	Reserved
9 MT_ADDR_EN	Memory Test Address Range Enable. If this bit is set, then the address range defined in the DDR_MT_ST_EXT_ADDR, DDR_MT_ST_ADDR, DDR_MT_END_EXT_ADDR, and DDR-MT_END_ADDR registers will be used. Please note the following restrictions when using this field to limit the address range to be tested: For any individual test, neither the starting address (DDR_MT_ST_EXT_ADDR || DDR_MT_ST_ADDR) nor the ending address (DDR_MT_END_EXT_ADDR || DDR_MT_END_ADDR) can be less than that used during the previous test. In addition, this field cannot be set if MTCR[MT_TRNARND] is set to 0b0000. 0b - Full memory range defined by CS*a*_BNDS registers will be used for the memory test. 1b - Memory range defined by DDR_MT_ST_EXT_ADDR, DDR_MT_ST_ADDR, DDR_MT_END_EXT_ADDR, and DDR_MT_END_ADDR is used.
8-1 —	Reserved
0 MT_STAT	Memory Test Status. After hardware clears MT_EN, this bit will be set if there was a fail. If there is a fail during the memory test (that is, a data miscompare), then this bit will be set at the same time that MT_EN is cleared. Software can clear this bit after it has been set.

Field	Function
	0b - No fail has been detected. 1b - A data miscompare was detected during the memory test.

12.4.59 DDR Memory Test Pattern n Register (DDR_MTP0 - DDR_MTP9)

12.4.59.1 Offset

For a = 0 to 9:

Register	Offset
DDR_MTPa	D20h + (a × 4h)

12.4.59.2 Function

The DDR memory test pattern *n* register provides the data pattern that will be written during the nth set of 32-bits of each 40-byte memory test pattern. This is used when DDR_MTCR[MT_EN] is set to enable the memory write/read test.

NOTE

Memory test read compares data that is written in this register.

12.4.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DDR_PATT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DDR_PATT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.59.4 Fields

Field	Function
31-0 DDR_PATT	DDR Pattern. This 32-bit pattern will be used during the memory test if enabled via DDR_MTCR[MT_EN]. This memory test will write/read a programmable 40-byte pattern. The 10 DDR_MTP*n* registers will create the 40-byte pattern that is used.

12.4.60 DDR Memory Test Start Extended Address (DDR_MT_ST_EXT_ADDR)

12.4.60.1 Offset

Register	Offset
DDR_MT_ST_EXT_ADDR	D60h

12.4.60.2 Function

The DDR memory test start extended address register provides the extended address that will be used with DDR_MTCR[MT_ADDR_EN] is set.

12.4.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MT_ST_EXT_ADDR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.60.4 Fields

Field	Function
31-8 —	Reserved
7-0 MT_ST_EXT_ADDR	Memory Test Start Extended Address. This field represents the starting extended address that will be used when MTCR[MT_ADDR_EN] is set. Please note the following restriction when using DDR_MTCR[MT_ADDR_EN] to limit the address range to be tested: For any individual test, neither the starting address (DDR_MT_ST_EXT_ADDR || DDR_MT_ST_ADDR) nor the ending address (DDR_MT_END_EXT_ADDR || DDR_MT_END_ADDR) can be less than that used during the previous test.

12.4.61 DDR Memory Test Start Address (DDR_MT_ST_ADDR)

12.4.61.1 Offset

Register	Offset
DDR_MT_ST_ADDR	D64h

12.4.61.2 Function

The DDR memory test start address register provides the address that will be used with DDR_MTCR[MT_ADDR_EN] is set.

12.4.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MT_ST_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MT_ST_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.61.4 Fields

Field	Function
31-0 MT_ST_ADDR	Memory Test Start Address. This field represents the starting address that will be used when MTCR[MT_ADDR_EN] is set. Please note the following restriction when using DDR_MTCR[MT_ADDR_EN] to limit the address range to be tested: For any individual test, neither the starting address (DDR_MT_ST_EXT_ADDR || DDR_MT_ST_ADDR) nor the ending address (DDR_MT_END_EXT_ADDR || DDR_MT_END_ADDR) can be less than that used during the previous test.

12.4.62 DDR Memory Test End Extended Address (DDR_MT_END_EXT_ADDR)

12.4.62.1 Offset

Register	Offset
DDR_MT_END_EXT_ADDR	D68h

12.4.62.2 Function

The DDR memory test end extended address register provides the extended address that will be used with DDR_MTCR[MT_ADDR_EN] is set.

12.4.62.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MT_END_EXT_ADDR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.62.4 Fields

Field	Function
31-8 —	Reserved
7-0 MT_END_EXT_ADDR	Memory Test End Extended Address. This field represents the ending extended address that will be used when MTCR[MT_ADDR_EN] is set. Please note the following restriction when using DDR_MTCR[MT_ADDR_EN] to limit the address range to be tested: For any individual test, neither the starting address (DDR_MT_ST_EXT_ADDR || DDR_MT_ST_ADDR) nor the ending address (DDR_MT_END_EXT_ADDR || DDR_MT_END_ADDR) can be less than that used during the previous test.

12.4.63 DDR Memory Test End Address (DDR_MT_END_ADDR)

12.4.63.1 Offset

Register	Offset
DDR_MT_END_ADDR	D6Ch

12.4.63.2 Function

The DDR memory test end address register provides the address that will be used with DDR_MTCR[MT_ADDR_EN] is set.

12.4.63.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MT_END_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MT_END_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.63.4 Fields

Field	Function
31-0 MT_END_ADDR	Memory Test End Address. This field represents the ending address that will be used when MTCR[MT_ADDR_EN] is set. Please note the following restriction when using DDR_MTCR[MT_ADDR_EN] to limit the address range to be tested: For any individual test, neither the starting address (DDR_MT_ST_EXT_ADDR || DDR_MT_ST_ADDR) nor the ending address (DDR_MT_END_EXT_ADDR || DDR_MT_END_ADDR) can be less than that used during the previous test.

12.4.64 Memory data path error injection mask high (DATA_ERR_INJECT_HI)

12.4.64.1 Offset

Register	Offset
DATA_ERR_INJECT_HI	E00h

12.4.64.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EIMH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EIMH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.64.3 Fields

Field	Function
31-0 EIMH	Error injection mask high data path. Used to test ECC by forcing errors on the high word of the data path. Setting a bit causes the corresponding data path bit to be inverted on memory bus writes.

12.4.65 Memory data path error injection mask low (DATA_ERR_INJECT_LO)

12.4.65.1 Offset

Register	Offset
DATA_ERR_INJECT_LO	E04h

12.4.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EIML															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EIML															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.65.3 Fields

Field	Function
31-0 EIML	Error injection mask low data path. Used to test ECC by forcing errors on the low word of the data path. Setting a bit causes the corresponding data path bit to be inverted on memory bus writes.

12.4.66 Memory data path error injection mask ECC (ECC_ERR_INJECT)

12.4.66.1 Offset

Register	Offset
ECC_ERR_INJECT	E08h

12.4.66.2 Function

The memory data path error injection mask ECC register sets the ECC mask, enables errors to be written to ECC memory, and allows the ECC byte to mirror the most significant data byte. In addition, a single address parity error may be injected through this register.

12.4.66.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	Reserved			Reserved								Reserved			APIEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						EMB	EIEN	EEIM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.66.4 Fields

Field	Function
31 —	Reserved
30-28 —	Reserved
27-20 —	Reserved
19-17 —	Reserved
16 APIEN	Address parity error injection enable. This bit will be cleared by hardware after a single address parity error has been injected. 0b - Address parity error injection disabled. 1b - Address parity error injection enabled.
15-10 —	Reserved
9 EMB	ECC Mirror Byte. 0b - Mirror byte functionality disabled. 1b - Mirror the most significant data path byte onto the ECC byte.
8 EIEN	Error Injection Enable. 0b - Error injection disabled. 1b - Error injection enabled. This applies to the data mask bits, the ECC mask bits, and the ECC mirror bit. Note that error injection should not be enabled until the memory controller has been enabled via DDR_SDRAM_CFG[MEM_EN].
7-0 EEIM	ECC error injection mask. Setting a mask bit causes the corresponding ECC bit to be inverted on memory bus writes.

12.4.67 Memory data path read capture high (CAPTURE_DATA_HI)

12.4.67.1 Offset

Register	Offset
CAPTURE_DATA_HI	E20h

12.4.67.2 Function

The memory data path read capture high register stores the high word of the read data path during error capture.

12.4.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECHD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECHD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.67.4 Fields

Field	Function
31-0 ECHD	Error capture high data path. Captures the high word of the data path when errors are detected.

12.4.68 Memory data path read capture low (CAPTURE_DATA_LO)

12.4.68.1 Offset

Register	Offset
CAPTURE_DATA_LO	E24h

12.4.68.2 Function

The memory data path read capture low register stores the low word of the read data path during error capture.

12.4.68.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.68.4 Fields

Field	Function
31-0 ECLD	Error capture low data path. Captures the low word of the data path when errors are detected.

12.4.69 Memory data path read capture ECC (CAPTURE_ECC)

12.4.69.1 Offset

Register	Offset
CAPTURE_ECC	E28h

12.4.69.2 Function

The memory data path read capture ECC register stores the ECC syndrome bits that were on the data bus when an error was detected.

12.4.69.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.69.4 Fields

Field	Function
31-0 ECE	Error capture ECC. Captures the ECC bits on the data path whenever errors are detected. 64-bit mode: In 64-bit mode, only 24:31 should be used, although 0:7, 8:15, and 16:23 show the 8-bit ECC code replicated. • 0:7 should be ignored • 8:15 should be ignored • 16:23 should be ignored • 24:31 all 64 bits 32-bit mode: • 0:7 should be ignored • 8:15 8-bit ECC for the 32 bits in beats 0, 2, 4, 6 in 32-bit bus mode • 16:23 should be ignored • 24:31 8-bit ECC for the 32 bits in beats 1, 3, 5, 7 in 32-bit bus mode

12.4.70 Memory error detect (ERR_DETECT)

12.4.70.1 Offset

Register	Offset
ERR_DETECT	E40h

12.4.70.2 Function

The memory error detect register stores the detection bits for multiple memory errors, single- and multiple-bit ECC errors, and memory select errors. It is a read/write register. A bit can be cleared by writing a one to the bit. System software can determine the type of memory error by examining the contents of this register. If an error is disabled with ERR_DISABLE, the corresponding error is never detected or captured in ERR_DETECT.

12.4.70.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MME	Reserved														Reserved
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			SSBE	Reserved			APE	ACE	Reserved		CDE	MBE	SBE	Reserved	MSE
W				W1C				W1C	W1C			W1C	W1C	W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.70.4 Fields

Field	Function
31 MME	Multiple memory errors. This bit is cleared by software writing a 1. 0b - Multiple memory errors of the same type were not detected. 1b - Multiple memory errors of the same type were detected.
30-17 —	Reserved
16-13 —	Reserved
12 SSBE	Scrubbed single-bit ECC error. This bit is cleared by software writing a 1. 0b - The number of scrubbed single-bit ECC errors detected has not crossed the threshold set in ERR_SBE[SBET]. 1b - The number of scrubbed single-bit ECC errors detected crossed the threshold set in ERR_SBE[SBET].
11-9 —	Reserved
8 APE	Address parity error. This bit is cleared by software writing a 1. 0b - An address parity error has not been detected. 1b - An address parity error has been detected.
7 ACE	Automatic calibration error. This bit is cleared by software writing a 1. 0b - An automatic calibration error has not been detected. 1b - An automatic calibration error has been detected.
6-5 —	Reserved
4 CDE	Corrupted data error. This bit is cleared by software writing a 1. 0b - A corrupted data error has not been detected. 1b - A corrupted data error has been detected. This bit will be set if the actual ECC is inverted from the expected ECC during a read command. The memory controller will intentionally invert the ECC code if *DDR_SDRAM_CFG_2[CD_DIS]* is cleared when corrupted data is written to memory. The *ERR_DETECT[MBE]* bit will also be set when a corrupted data error is detected. Note it is also possible for a 2-bit data error to cause the ECC code to become inverted, which would either mask a corrupted data error or cause a normal multi-bit error to also appear as a corrupted data error.
3 MBE	Multiple-bit error. This bit is cleared by software writing a 1. 0b - A multiple-bit error has not been detected. 1b - A multiple-bit error has been detected.
2 SBE	Single-bit ECC error. This bit is cleared by software writing a 1.

Table continues on the next page...

Field	Function
	0b - The number of single-bit ECC errors detected has not crossed the threshold set in ERR_SBE[SBET]. 1b - The number of single-bit ECC errors detected crossed the threshold set in ERR_SBE[SBET].
1 —	Reserved
0 MSE	Memory select error. This bit is cleared by software writing a 1. 0b - A memory select error has not been detected. 1b - A memory select error has been detected.

12.4.71 Memory error disable (ERR_DISABLE)

12.4.71.1 Offset

Register	Offset
ERR_DISABLE	E44h

12.4.71.2 Function

The memory error disable register allows selective disabling of the DDR controller's error detection circuitry. Disabled errors are not detected or reported.

12.4.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			SSBED	Reserved			APED	ACED	Reserved		CDED	MBED	SBED	Reserved	MSED
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.71.4 Fields

Field	Function
31-17 —	Reserved
16-13 —	Reserved
12 SSBED	Scrubbed single-bit ECC error disable. Note that if this bit is set, then single-bit ECC errors will not be automatically fixed by hardware, even if ECC scrubbing is enabled. 0b - Scrubbed single-bit ECC errors are enabled. 1b - Scrubbed single-bit ECC errors are disabled.
11-9 —	Reserved
8 APED	Address parity error disable. 0b - Address parity errors are detected if DDR_SDRAM_CFG_2[AP_EN] is set. They are reported if ERR_INT_EN[APEE] is set. 1b - Address parity errors are not detected or reported.
7 ACED	Automatic calibration error disable. 0b - Automatic calibration errors are enabled. 1b - Automatic calibration errors are disabled.
6-5 —	Reserved
4 CDED	Corrupted data error disable. 0b - Corrupted data error checking is enabled. 1b - Corrupted data error checking is disabled.
3 MBED	Multiple-bit ECC error disable. 0b - Multiple-bit ECC errors are detected if DDR_SDRAM_CFG[ECC_EN] is set. They are reported if ERR_INT_EN[MBEE] is set. See Error Management for more information.MBED must be zero and ERR_INT_EN[MBEE] and ECC_EN must be one to ensure that an interrupt is generated. 1b - Multiple-bit ECC errors are not detected or reported.
2 SBED	Single-bit ECC error disable. Note that if this bit is set, then single-bit ECC errors will not be automatically fixed by hardware, even if ECC fixing is enabled. 0b - Single-bit ECC errors are enabled. 1b - Single-bit ECC errors are disabled.
1 —	Reserved
0 MSED	Memory select error disable. 0b - Memory select errors are enabled. 1b - Memory select errors are disabled.

12.4.72 Memory error interrupt enable (ERR_INT_EN)

12.4.72.1 Offset

Register	Offset
ERR_INT_EN	E48h

12.4.72.2 Function

The memory error interrupt enable register enables ECC interrupts or memory select error interrupts. When an enabled interrupt condition occurs, the internal int_B signal is asserted to the programmable interrupt controller (PIC).

12.4.72.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			SSBEE	Reserved			APEE	ACEE	Reserved		CDEE	MBEE	SBEE	Reserved	MSEE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.72.4 Fields

Field	Function
31-17 —	Reserved
16-13 —	Reserved
12	Scrubbed single-bit ECC error interrupt enable.

Table continues on the next page...

Field	Function
SSBEE	0b - Scrubbed single-bit ECC errors cannot generate interrupts. 1b - Scrubbed single-bit ECC errors generate interrupts.
11-9 —	Reserved
8 APEE	Address parity error interrupt enable. 0b - Address parity errors cannot generate interrupts. 1b - Address parity errors generate interrupts.
7 ACEE	Automatic calibration error interrupt enable. 0b - Automatic calibration errors cannot generate interrupts. 1b - Automatic calibration errors generate interrupts.
6-5 —	Reserved
4 CDEE	Corrupted data error interrupt enable. 0b - Corrupted data errors cannot generate interrupts. 1b - Corrupted data errors generate interrupts.
3 MBEE	Multiple-bit ECC error interrupt enable. See Error Management for more information. Note that uncorrectable read errors may cause an interrupt. ERR_DISABLE[MBED] must be zero and MBEE and DDR_SDRAM_CFG[ECC_EN] must be set to ensure that an interrupt is generated. 0b - Multiple-bit ECC errors cannot generate interrupts. 1b - Multiple-bit ECC errors generate interrupts.
2 SBEE	Single-bit ECC error interrupt enable. 0b - Single-bit ECC errors cannot generate interrupts. 1b - Single-bit ECC errors generate interrupts.
1 —	Reserved
0 MSEE	Memory select error interrupt enable. 0b - Memory select errors do not cause interrupts. 1b - Memory select errors generate interrupts.

12.4.73 Memory error attributes capture (CAPTURE_ATTRIBUTES)

12.4.73.1 Offset

Register	Offset
CAPTURE_ATTRIBUTES	E4Ch

12.4.73.2 Function

The memory error attributes capture register sets attributes for errors including type, size, source, and others.

12.4.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	BNUM			Reserved	TSIZ			TSRC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TTYP		Reserved											VLD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.73.4 Fields

Field	Function
31 —	Reserved
30-28 BNUM	Data beat number. Captures the doubleword number for the detected error. Relevant only for ECC errors.
27 —	Reserved
26-24 TSIZ	Transaction size for the error. Captures the transaction size in double words. 000b - 8 double words 001b - 1 double word 010b - 2 double words 011b - 3 double words 100b - 4 double words 101b - 5 double word 110b - 6 double words 111b - 7 double words
23-16 TSRC	Transaction source for the error. See the Global Source and Target IDs section in the Memory Map chapter for the defined encoding.

Table continues on the next page...

Field	Function
15-14 —	Reserved
13-12 TTYP	Transaction type for the error. 00b - Reserved 01b - Write 10b - Read 11b - Read-modify-write
11-1 —	Reserved
0 VLD	Valid. Set as soon as valid information is captured in the error capture registers.

12.4.74 Memory error address capture (CAPTURE_ADDRESS)

12.4.74.1 Offset

Register	Offset
CAPTURE_ADDRESS	E50h

12.4.74.2 Function

The memory error address capture register holds the 32 lsbs of a transaction when a DDR ECC error is detected.

12.4.74.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.74.4 Fields

Field	Function
31-0 CADDR	Captured address. Captures the 32 lsbs of the transaction address when an error is detected.

12.4.75 Memory error extended address capture (CAPTURE_EXT_ADDRESS)

12.4.75.1 Offset

Register	Offset
CAPTURE_EXT_ADDRESS	E54h

12.4.75.2 Function

The memory error extended address capture register holds the four most significant transaction bits when an error is detected.

12.4.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CEADDR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.75.4 Fields

Field	Function
31-8 —	Reserved
7-0 CEADDR	Captured extended address. Captures the 8 msbs of the transaction address when an error is detected

12.4.76 Single-Bit ECC memory error management (ERR_SBE)

12.4.76.1 Offset

Register	Offset
ERR_SBE	E58h

12.4.76.2 Function

The single-bit ECC memory error management register stores the threshold value for reporting single-bit errors and the number of single-bit errors counted since the last error report. When the counter field reaches the threshold, it wraps back to the reset value (0). If necessary, software must clear the counter after it has managed the error.

12.4.76.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SSBET								SBET							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SSBEC								SBEC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.76.4 Fields

Field	Function
31-24 SSBET	Scrubbed single-bit error threshold. Establishes the number of single-bit errors that must be detected and fixed during ECC scrubbing before an error condition is reported.
23-16 SBET	Single-bit error threshold. Establishes the number of single-bit errors that must be detected before an error condition is reported.
15-8 SSBEC	Scrubbed single-bit error counter. Indicates the number of scrubbed single-bit errors detected and fixed in memory since the last error report. This counter is incremented only if a single-bit error is detected during the ECC interval scrubbing. If single-bit error reporting is enabled, an error is reported and an interrupt is generated when this value equals SSBET. SSBEC is automatically cleared when the threshold value is reached.
7-0 SBEC	Single-bit error counter. Indicates the number of single-bit errors detected and corrected since the last error report. If single-bit error reporting is enabled, an error is reported and an interrupt is generated when this value equals SBET. SBEC is automatically cleared when the threshold value is reached.

12.5 DDR Functional Description

The DDR SDRAM controller controls processor and I/O interactions with system memory. The memory system allows a wide range of memory devices to be mapped to any arbitrary chip select, and support is provided for registered DIMMs and unbuffered DIMMs. However, registered DIMMs cannot be mixed with unbuffered DIMMs.

The figure below is a high-level block diagram of the DDR memory controller. Requests are received from the internal mastering device and the address is decoded to generate the physical bank, logical bank, row, and column addresses. The transaction is compared with values in the row open table to determine if the address maps to an open page. If the transaction does not map to an open page, an active command is issued.

Programmable parameters allow for a variety of memory organizations and timings. Optional error checking and correcting (ECC) protection is provided for the DDR SDRAM data bus. Using ECC, the DDR memory controller detects and corrects all single-bit errors within the data bus, detects all double-bit errors within the data bus, and detects all errors within a nibble. The controller allows as many as 64 pages to be open simultaneously. The amount of time (in clock cycles) the pages remain open is programmable with DDR_SDRAM_INTERVAL[BSTOPRE].

NOTE

When ECC DRAM is present ECC should be enabled.

Figure 12-2. DDR Memory Controller Block Diagram

Read and write accesses to memory are burst oriented; accesses start at a selected location and continue for a programmed number of higher locations (4 or 8) in a programmed sequence. Accesses to closed pages start with the registration of an ACTIVE command followed by a READ or WRITE. (Accessing open pages does not require an ACTIVE command.) The address bits registered coincident with the activate command specifies the logical bank and row to be accessed. The address coincident with the READ or WRITE command specify the logical bank and starting column for the burst access.

The data interface is source synchronous, meaning whatever sources the data also provides a clocking signal to synchronize data reception. These bidirectional data strobes (MDQS[0:8]) are inputs to the controller during reads and outputs during writes. The DDR SDRAM specification requires the data strobe signals to be centered within the data tenure during writes and to be offset by the controller to the center of the data tenure during reads. This delay is implemented in the controller for both reads and writes.

When ECC is enabled, 1 clock cycle is added to the read path to check ECC and correct single-bit errors. ECC generation does not add a cycle to the write path.

The address and command interface is also source synchronous, although 1/16 cycle adjustments are provided for adjusting the clock alignment.

This figure shows an example DDR SDRAM configuration with four physical banks each comprised of nine 8M x 8 DDR modules for a total of 256 Mbytes of system memory. One of the nine modules is used for the memory's ECC checking function. Certain address and control lines may require buffering. Analysis of the device's AC timing specifications, desired memory operating frequency, capacitive loads, and board routing loads can assist the system designer in deciding signal buffering requirements. The DDR memory controller drives 16 address pins, but in this example the DDR SDRAM devices use only 12 bits.

1. All signals are connected in common (in parallel) except for MCS_B[0:3], MCK[0:5], MDM[0:8], and the data bus signals.
2. Each of the MCS_B[0:3] signals correspond with a separate physical bank of memory.
3. Buffering may be needed if large memory arrays are used.
4. MCK[0:5] may be apportioned among all memory devices. Complementary bus is not shown.

Figure 12-3. Example 256-Mbyte DDR SDRAM Configuration

Error Management explains how the DDR memory controller handles errors.

12.5.1 DDR SDRAM Interface Operation

The DDR memory controller supports many different DDR SDRAM configurations; see Supported DDR SDRAM Organizations for details. SDRAMs with different sizes can be used in the same system. Sixteen multiplexed address signals and three logical bank select signals support numerous device densities. Four chip select (CS_B) signals support up to two DIMMs of memory. The DDR SDRAM physical banks can be built from standard memory modules or directly-attached memory devices. The data path to individual physical banks is 64 or 32 bits wide, 72 or 40 bits with ECC. The controller supports physical bank sizes of up to 16 Gbytes. The physical banks can be constructed using x 8, x16, or x32 (with 1 DQS per data byte) memory devices. Nine data qualifier (DQM) signals provide byte selection for memory accesses.

NOTE

An 8-bit DDR SDRAM device has a DQM signal and eight data signals (DQ[0:7]). A 16-bit DDR SDRAM device has two DQM signals associated with specific halves of the 16 data signals (DQ[0:7] and DQ[8:15]).

When ECC is enabled, all memory accesses are performed on double-word boundaries (that is, all DQM signals are set simultaneously). However, when ECC is disabled, the memory system uses the DQM signals for byte lane selection when using x8 or x16 devices.

This table shows the relationships between data byte lanes, MDMn, MDQSn, and MDQn when DDR SDRAM memories are used with x8 or x16 devices.

Table 12-8. Byte Lane to Data Relationship for x8 and x16 devices

Data Byte Lane	Data Bus Mask	Data Bus Strobe	Data Bus 64-Bit Mode
0 (MSB)	MDM[0]	MDQS[0]/MDQS_B[0]	MDQ[0:7]
1	MDM[1]	MDQS[1]/MDQS_B[1]	MDQ[8:15]
2	MDM[2]	MDQS[2]/MDQS_B[2]	MDQ[16:23]
3	MDM[3]	MDQS[3]/MDQS_B[3]	MDQ[24:31]
4	MDM[4]	MDQS[4]/MDQS_B[4]	MDQ[32:39]
5	MDM[5]	MDQS[5]/MDQS_B[5]	MDQ[40:47]
6	MDM[6]	MDQS[6]/MDQS_B[6]	MDQ[48:55]
7 (LSB)	MDM[7]	MDQS[7]/MDQS_B[7]	MDQ[56:63]
8 (ECC)	MDM8	MDQS8/MDQS8_B	MECC[0:7]

12.5.1.1 Supported DDR SDRAM Organizations

These tables describe DDR SDRAM device configurations supported by the DDR memory controller.

Table 12-9. Supported SDRAM Device Configurations, DDR4

SDRAM Device	Device Configuration	Row x Column x Sub-bank x Bank Group Bits	64-Bit Bank Size	Four Banks of Memory
2 Gbits	256 Mbits x 8	14 x 10 x 2 x 2	2 Gbytes	8 Gbytes
2 Gbits	128 Mbits x 16	14 x 10 x 2 x 1	1 Gbyte	4 Gbytes
4 Gbits	512 Mbits x 8	15 x 10 x 2 x 2	4 Gbytes	16 Gbytes
4 Gbits	256 Mbits x 16	15 x 10 x 2 x 1	2 Gbytes	8 Gbytes
8 Gbits	1 Gbit x 8	16 x 10 x 2 x 2	8 Gbytes	32 Gbytes
8 Gbits	512 Mbits x 16	16 x 10 x 2 x 1	4 Gbytes	16 Gbytes
16 Gbits	2 Gbits x 8	17 x 10 x 2 x 2	16 Gbytes	64 Gbytes
16 Gbits	1 Gbit x 16	17 x 10 x 2 x 1	8 Gbytes	32 Gbytes

If a transaction request is issued to the DDR memory controller and the address does not lie within any of the programmed address ranges for an enabled chip select, a memory select error is flagged. Errors are described in detail in Error Management

By using a memory-polling algorithm at power-on reset or by querying the JEDEC serial presence detect capability of memory modules, system firmware uses the memory-boundary registers to configure the DDR memory controller to map the size of each bank in memory. The memory controller uses its bank map to assert the appropriate MCS*n*_B signal for memory accesses according to the provided bank starting and ending addresses. The memory banks are not required to be mapped to a contiguous address space.

12.5.2 DDR SDRAM Address Multiplexing

The following tables show the address bit encodings for each DDR SDRAM configuration. The address presented at the memory controller signals MA[17:0] use MA[17] as the msb and MA[0] as the lsb. Also, MA[10] is used as the auto-precharge bit for reads and writes, so the column address can never use MA[10].

Table 12-10. DDR4 Address Multiplexing for 64-Bit Data Bus with no Interleaving

		msb	Address from Core Master																																		lsb
Row x Col		34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2-0
17 x 10 x 2 x 2	MACT_B		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	MBG																			1	0													
	MBA																					1	0											
	MCAS_B																							9	8	7	6	5	4	3	2	1	0	
16 x 10 x 2 x 2	MACT_B			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	MBG																			1	0													
	MBA																					1	0											
	MCAS_B																							9	8	7	6	5	4	3	2	1	0	
15 x 10 x 2 x 2	MACT_B				14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	MBG																			1	0													
	MBA																					1	0											
	MCAS_B																							9	8	7	6	5	4	3	2	1	0	
14 x 10 x 2 x 2	MRAS_B					13	12	11	10	9	8	7	6	5	4	3	2	1	0															
	MBG																			1	0													
	MBA																					1	0											
	MCAS_B																							9	8	7	6	5	4	3	2	1	0	

Chip select interleaving is supported for the memory controller, and is programmed in DDR_SDRAM_CFG[BA_INTLV_CTL]. Interleaving is supported between chip selects 0 and 1 or chip selects 2 and 3. In addition, interleaving between all four chip selects can be enabled. When interleaving is enabled, the chip selects being interleaved must use the same size of memory. If two chip selects are interleaved, then 1 extra bit in the address decode is used for the interleaving to determine which chip select to access. If four chip selects are interleaved, then two extra bits are required in the address decode.

The following table illustrates examples of address decode when interleaving between two chip selects.

Table 12-11. Example of Address Multiplexing for 32/64-Bit Data Bus Interleaving Between Two Banks

		msb	Address from Core Master																													lsb
Row x Col		**32**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2-0**
14 x 10 x 3	MACT_B		13	12	11	10	9	8	7	6	5	4	3	2	1	0																
	CS_SEL																0															
	MBG																	1	0													
	MBA																			1	0											
	MCAS_B																					9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	MACT_B			12	11	10	9	8	7	6	5	4	3	2	1	0																
	CS_SEL																0															
	MBG																	1	0													
	MBA																			1	0											
	MCAS_B																					9	8	7	6	5	4	3	2	1	0	

The following table shows examples of address decode when interleaving between four chip selects.

Table 12-12. Example of Address Multiplexing for 64-Bit Data Bus Interleaving Between Four Banks

		msb	Address from Core Master																																		lsb
Row x Col		33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2-0
14 x 10 x 3	MACT_B		13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
	CS_SEL																1	0															
	MBG																		1	0													
	MBA																				1	0											
	MCAS_B																						9	8	7	6	5	4	3	2	1	0	
13 x 10 x 3	MACT_B			12	11	10	9	8	7	6	5	4	3	2	1	0																	
	CS_SEL																1	0															
	MBG																		1	0													
	MBA																				1	0											
	MCAS_B																						9	8	7	6	5	4	3	2	1	0	

12.5.3 DDR SDRAM Registered DIMM Mode

To reduce loading, registered DIMMs latch the DDR SDRAM control signals internally before using them to access the array. Setting DDR_SDRAM_CFG[RD_EN] compensates for this delay on the DIMMs' control bus by delaying the data and data mask writes (on SDRAM buses) by an extra SDRAM clock cycle.

NOTE

Application system board must assert the reset signal on DDR memory devices until software is able to program the DDR

memory controller configuration registers, and must deassert the reset signal on DDR memory devices before DDR_SDRAM_CFG[MEM_EN] is set. This ensures that the DDR memory devices are held in reset until a stable clock is provided and, further, that a stable clock is provided before memory devices are released from reset.

12.5.4 DDR SDRAM Write Timing Adjustments

The DDR memory controller facilitates system design flexibility by providing a write timing adjustment calculated by write leveling for data and DQS. The DDR SDRAM specification requires DQS be received no sooner than 75% of an SDRAM clock period-and no later than 125% of a clock period-from the capturing clock edge of the command/address at the SDRAM. The write leveling calibration is used to meet this timing requirement for a variety of system configurations, ranging from a system with one DIMM to a fully populated system with two DIMMs. DDR_WRLVL_CNTL is used to set up and enable write leveling.

12.5.5 DDR SDRAM Refresh

The DDR memory controller supports auto-refresh and self-refresh. Auto refresh is used during normal operation and is controlled by the DDR_SDRAM_INTERVAL[REFINT] value; self-refresh is used only when the DDR memory controller is set to enter a sleep power management state. The REFINT value, which represents the number of memory bus clock cycles between refresh cycles, must allow for possible outstanding transactions to complete before a refresh request is sent to the memory after the REFINT value is reached. If a memory transaction is in progress when the refresh interval is reached, the refresh cycle waits for the transaction to complete. In the worst case, the refresh cycle must wait the number of bus clock cycles required by the longest programmed access. To ensure that the latency caused by a memory transaction does not violate the device refresh period, it is recommended that the programmed value of REFINT be less than that required by the SDRAM.

When a refresh cycle is required, the DDR memory controller does the following:

1. Completes all current memory requests.
2. Closes all open pages with a PRECHARGE-ALL command to each DDR SDRAM bank with an open page (as indicated by the row open table).

3. Issues one or more auto-refresh commands to each DDR SDRAM bank (as identified by its chip select) to refresh one row in each logical bank of the selected physical bank.

The auto-refresh commands are staggered across the four possible banks to reduce the system's instantaneous power requirements. Two sets of auto refresh commands will be issued on consecutive cycles when the memory is fully populated with two DIMMs. The initial PRECHARGE-ALL commands are also staggered in two groups for convenience. It is important to note that when entering self-refresh mode, only one refresh command is issued simultaneously to all physical banks. For this entire refresh sequence, no cycle optimization occurs for the usual case where fewer than four banks are installed. After the refresh sequence completes, any pending memory request is initiated after an inactive period specified by TIMING_CFG_1 [REFREC] and TIMING_CFG_3[EXT_REFREC]. In addition, posted refreshes are supported to allow the refresh interval to be set to a larger value.

12.5.5.1 DDR SDRAM Refresh Timing

Refresh timing for the DDR SDRAM is controlled by the programmable timing parameter TIMING_CFG_1 [REFREC], which specifies the number of memory bus clock cycles from the refresh command until a logical bank activate command is allowed. The DDR memory controller implements bank staggering for refreshes, as shown in this figure (TIMING_CFG_1 [REFREC] = 10 in this example).

Figure 12-4. DDR SDRAM Bank Staggered Auto Refresh Timing

System software is responsible for optimal configuration of TIMING_CFG_1 [REFREC] and TIMING_CFG_3[EXT_REFREC] at reset. Configuration must be completed before DDR SDRAM accesses are attempted.

12.5.5.2 DDR SDRAM Refresh and Power-Saving Modes

In full-on mode, the DDR memory controller supplies the normal auto refresh to SDRAM. In sleep mode, the DDR memory controller can be configured to take advantage of self-refreshing SDRAMs or to provide no refresh support. Self-refresh support is enabled with the SREN memory control parameter.

This table summarizes the refresh types available in each power-saving mode.

Table 12-13. DDR SDRAM Power-Saving Modes Refresh Configuration

Power Saving Mode	Refresh Type	SREN
Sleep	Self	1
	None	-

Note that in the absence of refresh support, system software must preserve DDR SDRAM data (such as by copying the data to disk) before entering the power-saving mode.

The dynamic power-saving mode uses the CKE DDR SDRAM pin to dynamically power down when there is no system memory activity. The CKE pin is negated when both of the following conditions are met:

- No memory refreshes are scheduled
- No memory accesses are scheduled

CKE is reasserted when a new access or refresh is scheduled or the dynamic power mode is disabled. This mode is controlled with DDR_SDRAM_CFG[DYN_PWR_MGMT].

Dynamic power management mode offers tight control of the memory system's power consumption by trading power for performance through the use of CKE. Powering up the DDR SDRAM when a new memory reference is scheduled causes an access latency penalty, depending on whether active or precharge powerdown is used, along with the settings of TIMING_CFG_0[ACT_PD_EXIT] and TIMING_CFG_0[PRE_PD_EXIT].

12.5.6 DDR Data Beat Ordering

Transfers to and from memory are always performed in four- or eight-beat bursts. For transfer sizes other than four or eight beats, the data transfers are still operated as four- or eight-beat bursts. If ECC is enabled and either the access is not doubleword aligned or the

size is not a multiple of a doubleword, a full read-modify-write is performed for a write to SDRAM. If ECC is disabled or both the access is doubleword aligned with a size that is a multiple of a doubleword, the data masks (MDM[0:8]) can be used to prevent the writing of unwanted data to SDRAM. The DDR memory controller also uses data masks to prevent all unintended full double words from writing to SDRAM. For example, if a write transaction is desired with a size of one double word (8 bytes), then the remaining Seven beats of data are not written to DRAM.

All writes for DDR4 mode will be aligned to beat 0 of the DRAM.

12.5.7 Page Mode and Logical Bank Retention

The DDR memory controller supports an open/closed page mode with an allowable open page for each logical bank (and bank group for DDR4) of DRAM used. In closed page mode for DDR SDRAMs, the DDR memory controller uses the SDRAM auto-precharge feature, which allows the controller to indicate that the page must be automatically closed by the DDR SDRAM after the READ or WRITE access. This is performed by using MA[10] of the address during the COMMAND phase of the access to enable auto-precharge. Auto-precharge is non-persistent in that it is either enabled or disabled for each individual READ or WRITE command. It can, however, be enabled or disabled separately for each chip select.

When the DDR memory controller operates in open page mode, it retains the currently active SDRAM page by not issuing a precharge command. The page remains opens until one of the following conditions occurs:

- Refresh interval is met.
- The user-programmable DDR_SDRAM_INTERVAL[BSTOPRE] value is exceeded.
- There is a logical bank row collision with another transaction that must be issued.

Page mode can dramatically reduce access latencies for page hits. Depending on the memory system design and timing parameters, using page mode can save two to three clock cycles for subsequent burst accesses that hit in an active page. Also, better performance can be obtained by using more banks, especially in systems which use many different channels. Page mode is disabled by clearing DDR_SDRAM_INTERVAL[BSTOPRE] or setting CS n_CONFIG[AP_ nEN].

12.5.8 Error Checking and Correcting (ECC)

The DDR memory controller supports error checking and correcting (ECC) for the data path between the core master and system memory. The memory detects all double-bit errors, detects all multi-bit errors within a nibble, and corrects all single-bit errors. Other errors may be detected, but are not guaranteed to be corrected or detected. Double-bit errors are always reported when error reporting is enabled. When a single-bit error occurs, the single-bit error counter register is incremented, and its value compared to the single-bit error trigger register. An error is reported when these values are equal. The single-bit error registers can be programmed such that minor memory faults are corrected and ignored, but a catastrophic memory failure generates an interrupt.

For writes that are smaller than 64 bits, the DDR memory controller performs a double-word read from system memory of the address for the write (checking for errors), and merges the write data with the data read from memory. Then, a new ECC code is generated for the merged double word. The data and ECC code is then written to memory. If a multi-bit error is detected on the read, the transaction completes the read-modify-write to keep the DDR memory controller from hanging. However, the corrupt data is masked on the write, so the original contents in SDRAM remain unchanged.

The DDR controller also supports ECC scrubbing, which is enabled via DDR_SDRAM_CFG_3[ECC_SCRUB_EN]. In this mode, all single-bit errors detected will be fixed by hardware. In addition, DDR_SDRAM_CFG_3[ECC_SCRUB_INT] can be programmed to enable periodic reads by the DDR controller to search for and fix single-bit errors.

The syndrome encodings for the ECC code are shown in these tables.

Table 12-14. DDR SDRAM ECC Syndrome Encoding

Data Bit	Syndrome Bit									Data Bit	Syndrome Bit							
	7	6	5	4	3	2	1	0			7	6	5	4	3	2	1	0
0	•	•						•		32			•	•				•
1	•		•					•		33			•		•			•
2	•			•				•		34	•		•		•			
3	•				•			•		35		•	•		•			
4	•	•				•				36			•	•		•		
5	•		•			•				37			•		•	•		
6	•			•		•				38	•		•		•	•		•
7	•				•	•				39		•	•		•	•		•
8	•	•					•			40			•	•			•	
9	•		•				•			41			•		•		•	
10	•			•			•			42	•		•		•		•	•
11	•				•		•			43		•	•		•		•	•

Table continues on the next page...

Table 12-14. DDR SDRAM ECC Syndrome Encoding (continued)

Data Bit	Syndrome Bit									Data Bit	Syndrome Bit							
	7	6	5	4	3	2	1	0			7	6	5	4	3	2	1	0
12	•	•				•	•	•		44			•	•		•	•	•
13	•		•			•	•	•		45			•		•	•	•	•
14	•			•		•	•	•		46	•		•		•	•	•	
15	•				•	•	•	•		47		•	•		•	•	•	
16		•	•					•		48		•				•	•	
17		•		•				•		49			•			•	•	
18		•			•			•		50				•		•	•	
19	•	•			•					51	•					•	•	
20		•	•			•				52		•				•		•
21		•		•		•				53			•			•		•
22		•			•	•				54				•		•		•
23	•	•			•	•		•		55	•					•		•
24		•	•				•			56		•					•	•
25		•		•			•			57			•				•	•
26		•			•		•			58				•			•	•
27	•	•			•		•	•		59	•						•	•
28		•	•			•	•	•		60				•	•		•	
29		•		•		•	•	•		61	•			•	•		•	•
30		•			•	•	•	•		62		•		•	•		•	•
31	•	•			•	•	•			63			•	•	•		•	•

Table 12-15. DDR SDRAM ECC Syndrome Encoding (Check Bits)

Check Bit	Syndrome Bit							
	7	6	5	4	3	2	1	0
0	•							
1		•						
2			•					
3				•				
4					•			
5						•		
6							•	
7								•

12.5.9 Error Management

The DDR memory controller detects four different kinds of errors: training, single-bit, multi-bit, and memory select errors. The following discussion assumes all the relevant error detection, correction, and reporting functions are enabled as described in Memory error interrupt enable (ERR_INT_EN), Memory error disable (ERR_DISABLE), and Memory error detect (ERR_DETECT)

Single-bit errors are counted and reported based on the ERR_SBE value. When a single-bit error is detected, the DDR memory controller does the following:

- Corrects the data
- Increments the single-bit error counter ERR_SBE[SBEC]
- Generates a critical interrupt if the counter value ERR_SBE[SBEC] equals the programmable threshold ERR_SBE[SBET]
- Completes the transaction normally

If a multi-bit error is detected for a read, the DDR memory controller logs the error and generates the machine check or critical interrupt (if enabled, as described in Memory error disable (ERR_DISABLE)). Another error the DDR memory controller detects is a memory select error, which causes the DDR memory controller to log the error and generate a critical interrupt (if enabled, as described in Memory error detect (ERR_DETECT)). This error is detected if the address from the memory request does not fall into any of the enabled, programmed chip select address ranges. For all memory select errors, the DDR memory controller does not issue any transactions onto the pins after the first read has returned data strobes. If the DDR memory controller is not using sample points, then a dummy transaction is issued to DDR SDRAM with the first enabled chip select. In this case, the source port on the pins is forced to 0x1F to show the transaction is not real. Table 12-16 shows the errors with their descriptions. The final error the memory controller detects is the automatic calibration error. This error is set if the memory controller detects an error during its training sequence.

Table 12-16. Memory Controller Errors

Category	Error	Descriptions	Action	Detect Register
Notification	Single-bit ECC threshold	The number of ECC errors has reached the threshold specified in the ERR_SBE.	The error is reported via regular or critical interrupt if enabled.	The error control register only logs read versus write, not full type
Access Error	Multi-bit ECC error	A multi-bit ECC error is detected during a read, or read-modify-write memory operation.	The error is reported via machine check or critical interrupt if enabled.	
	Memory select error	Read, or write, address does not fall within the address range of any of the memory banks.		

12.5.10 DDR Rapid Clear of Memory

Upon Hard Fail condition, the security monitor drives an interrupt to the DDR controller indicating the memory needs to be cleared, the DDR controller will clear all of memory defined by the CS*n*_BNDS registers for all enabled chip selects. DDRDSR_2[RPD_ST] will be set once the DDR controller begins clearing memory. At this time, writes to the DDR CCSR space are not allowed to modify the register values. Once the rapid clear of memory sequence is complete, writes to the DDR registers may proceed. The DDRDSR_2[RPD_EN] bit is provided so software can determine when the rapid clear of memory sequence is complete and can therefore update registers again. Software can clear the RPD_ST and RPD_EN bits after the rapid memory clear is complete.

12.6 Initialization/Application Information

At system reset or start-up, initialization software (boot code) must set up the programmable parameters in the memory interface configuration registers. See DDR register descriptions for more detailed descriptions of the configuration registers. These parameters are shown in this table.

Table 12-17. Memory Interface Configuration Register Initialization Parameters

Name	Description	Parameter	Section/Page
CS n_BNDS	Chip select memory bounds	SA EA	Chip select a memory bounds (CS0_BNDS - CS3_BNDS)
CS n_CONFIG	Chip select configuration	CS_EN AP_EN ODT_RD_CFG ODT_WR_CFG BA_BITS_CS ROW_BITS_CS BG_BITS_CS COL_BITS_CS	Chip select a configuration (CS0_CONFIG - CS3_CONFIG)
TIMING_CFG_3	Extended timing parameters for fields in TIMING_CFG_1	EXT_PRETOACT EXT_ACTTOPRE EXT_ACTTORW EXT_REFREC EXT_CASLAT EXT_WRREC	DDR SDRAM timing configuration 3 (TIMING_CFG_3)

Table continues on the next page...

Table 12-17. Memory Interface Configuration Register Initialization Parameters (continued)

Name	Description	Parameter	Section/Page
		CNTL_ADJ	
TIMING_CFG_0	Timing configuration	RWT WRT RRT WWT ACT_PD_EXIT PRE_PD_EXIT MRS_CYC	DDR SDRAM timing configuration 0 (TIMING_CFG_0)
TIMING_CFG_1	Timing configuration	PRETOACT ACTTOPRE ACTTORW CASLAT REFREC WRREC ACTTOACT WRTORD	DDR SDRAM timing configuration 1 (TIMING_CFG_1)
TIMING_CFG_2	Timing configuration	ADD_LAT WR_LAT RD_TO_PRE CKE_PLS FOUR_ACT	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
DDR_SDRAM_CFG	Control configuration	SREN ECC_EN RD_EN SDRAM_TYPE DYN_PWR BE_8 DBW T2_EN T3_EN BA_INTLV_CTL HSE BI	DDR SDRAM control configuration (DDR_SDRAM_CFG)
DDR_SDRAM_CFG_2	Control configuration	ODT_CFG NUM_PR OBC_CFG AP_EN	DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)

Table continues on the next page...

Table 12-17. Memory Interface Configuration Register Initialization Parameters (continued)

Name	Description	Parameter	Section/Page
		D_INIT RCW_EN MD_EN	
DDR_SDRAM_MODE	Mode configuration	ESDMODE SDMODE	DDR SDRAM mode configuration (DDR_SDRAM_MODE)
DDR_SDRAM_MODE_2	Mode configuration	ESDMODE2 ESDMODE3	DDR SDRAM mode configuration 2 (DDR_SDRAM_MODE_2)
DDR_SDRAM_INTERVAL	Interval configuration	REFINT BSTOPRE	DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL)
DDR_DATA_INIT	Data initialization configuration register	INIT_VALUE	DDR SDRAM data initialization (DDR_DATA_INIT)
DDR_SDRAM_CLK_CNTL	Clock adjust	CLK_ADJUST	DDR SDRAM clock control (DDR_SDRAM_CLK_CNTL)
DDR_INIT_ADDR	Initialization address	INIT_ADDR	DDR training initialization address (DDR_INIT_ADDR)
DDR_INIT_EXT_ADDRESS	Extended initialization address	INIT_EXT_ADDR	DDR training initialization extended address (DDR_INIT_EXT_ADDRESS)
TIMING_CFG_4	Timing configuration	RWT WRT RRT WWT EXT_RWT EXT_WRT EXT_RRT EXT_WWT EXT_REFINT DLL_LOCK	DDR SDRAM timing configuration 4 (TIMING_CFG_4)
TIMING_CFG_5	Timing configuration	RODT_ON RODT_OFF WODT_ON WODT_OFF	DDR SDRAM timing configuration 5 (TIMING_CFG_5)
TIMING_CFG_7	Timing configuration	CKE_RST CKSRE CKSRX PAR_LAT CS_TO_CMD	DDR SDRAM timing configuration 7 (TIMING_CFG_7)

Table continues on the next page...

Table 12-17. Memory Interface Configuration Register Initialization Parameters (continued)

Name	Description	Parameter	Section/Page
TIMING_CFG_8	Timing configuration	RWT_BG WRT_BG RRT_BG WWT_BG ACTTOACT_BG WRTORD_BG PRE_ALL_REC	DDR SDRAM timing configuration 8 (TIMING_CFG_8)
DDR_ZQ_CNTL	ZQ calibration control	ZQ_EN ZQINIT ZQOPER ZQCS	DDR ZQ calibration control (DDR_ZQ_CNTL)
DDR_WRLVL_CNTL	Write leveling control	WRLVL_EN WRLVL_MRD WRLVL_ODTEN WRLVL_DQSEN WRLVL_SMPL WRLVL_WLR WRLVL_START	DDR write leveling control (DDR_WRLVL_CNTL)
DDR_WRLVL_CNTL_2	Write leveling control	WRLVL_START_1 WRLVL_START_2 WRLVL_START_3 WRLVL_START_4	DDR write leveling control 2 (DDR_WRLVL_CNTL_2)
DDR_WRLVL_CNTL_3	Write leveling control	WRLVL_START_5 WRLVL_START_6 WRLVL_START_7 WRLVL_START_8	DDR write leveling control 3 (DDR_WRLVL_CNTL_3)
DDR_SR_CNTR	Self refresh control	SR_IT	DDR Self Refresh Counter (DDR_SR_CNTR)
DDR_SDRAM_RCW_1	Register control words configuration	RCW0 RCW1 RCW2 RCW3 RCW4 RCW5 RCW6 RCW7	DDR Register Control Words 1 (DDR_SDRAM_RCW_1)
DDR_SDRAM_RCW_2	Register control words configuration	RCW8 RCW9	DDR Register Control Words 2 (DDR_SDRAM_RCW_2)

Table continues on the next page...

Table 12-17. Memory Interface Configuration Register Initialization Parameters (continued)

Name	Description	Parameter	Section/Page
		RCW10 RCW11 RCW12 RCW13 RCW14 RCW15	
DDR_SDRAM_CFG_3	Control configuration	DDRC_RST ECC_FIX_EN ECC_SCRUB_INT WR_PRE RD_PRE DM_CFG	DDR SDRAM control configuration 3 (DDR_SDRAM_CFG_3)
DDRCDR_1	Driver control	DHC_EN ODT DSO_C_EN DSO_D_EN DSO_CZ DSO_DZ	DDR Control Driver Register 1 (DDRCDR_1)
DDRCDR_2	Driver control	DSO_CLK_EN DSO_CLKZ	DDR Control Driver Register 2 (DDRCDR_2)

12.6.1 Programming Summary

Depending on the memory type used, certain fields must be programmed differently.

This table illustrates the differences in certain fields for DDR4 memory types. Note: This table does not list all fields that must be programmed.

Table 12-18. Programming Summary, DDR4 memory types

Parameter	Description	Summary	Section
AP*n* _EN	Chip Select *n*Auto Precharge Enable	Can be used to place chip select *n*in auto precharge mode	Chip select a configuration (CS0_CONFIG - CS3_CONFIG)

Table continues on the next page...

Table 12-18. Programming Summary, DDR4 memory types (continued)

Parameter	Description	Summary	Section
ODT_RD_CFG	Chip Select ODT Read Configuration	Can be enabled to assert ODT if desired. This could be set differently depending on system topology. However, systems with only 1 chip select will typically not use ODT when issuing reads to the memory.	Chip select a configuration (CS0_CONFIG - CS3_CONFIG)
ODT_WR_CFG	Chip Select ODT Write Configuration	Can be enabled to assert ODT if desired. This could be set differently depending on system topology. However, ODT will typically be set to assert for the chip select that is getting written to (value would be set to 001).	Chip select a configuration (CS0_CONFIG - CS3_CONFIG)
PRETOACT	Precharge to Activate Timing	Should be set according to the specifications for the memory used (t_{RP})	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
ACTTOPRE	Activate to Precharge Timing	Should be set, along with the Extended Activate to Precharge Timing, according to the specifications for the memory used (t_{RAS})	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
ACTTORW	Activate to Read/Write Timing	Should be set according to the specifications for the memory used (t_{RCD})	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
CASLAT	CAS Latency	Should be set, along with the Extended CAS Latency, to the desired CAS latency	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
REFREC	Refresh Recovery	Should be set, along with the Extended Refresh Recovery, to the specifications for the memory used (T_{RFC})	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
WRREC	Write Recovery	Should be set according to the specifications for the memory used (t_{WR}). If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this should be programmed to t_{WR} + 2 DRAM cycles.	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
ACTTOACT	Activate *A* to Activate *B*	Should be set according to the specifications for the memory used (t_{RRD})	DDR SDRAM timing configuration 1 (TIMING_CFG _1)
WRTORD	Write to Read Timing	Should be set according to the specifications for the memory used (t_{WTR}) If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this should be programmed to t_{WTR} + 2 DRAM cycles.	DDR SDRAM timing configuration 1 (TIMING_CFG _1)

Table continues on the next page...

Table 12-18. Programming Summary, DDR4 memory types (continued)

Parameter	Description	Summary	Section
ADD_LAT	Additive Latency	Should be set to the desired additive latency. This must be set to a value less than TIMING_CFG_1[ACTTORW]	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
WR_LAT	Write Latency	Should be set to the desired write latency. The minimum WR_LAT that can be used in 1T timing mode is 5 cycles if DDR_RATE=0	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
RD_TO_PRE	Read to Precharge Timing	Should be set according to the specifications for the memory used (t_{RTP}). Time between read and precharge for non-zero value of additive latency (AL) is a minimum of AL + t_{RTP} cycles. If DDR_SDRAM_CFG_2[OBC_CFG] is set, then this should be programmed to t_{RTP} + 2 DRAM cycles.	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
CKE_PLS	Minimum CKE Pulse Width	Should be set according to the specifications for the memory used (t_{CKE})	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
FOUR_ACT	Four Activate Window	Should be set according to the specifications for the memory used (t_{FAW}).	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
RD_EN	Registered DIMM Enable	If registered DIMMs are used, then this field should be set to 1	DDR SDRAM control configuration (DDR_SDRAM_CFG)
BE_8	8-beat burst enable	Set to 1 for fixed 8-beat burst mode. Set to 0 for on-the-fly burst chop mode and ensure DDR_SDRAM_CFG_2[OBC_CFG] = 1. If this bit is set to 0, then other requirements in TIMING_CFG_4 will be needed to ensure t_{CCD} is met.	DDR SDRAM control configuration (DDR_SDRAM_CFG)
T2_EN	2T Timing Enable	In heavily loaded systems, this can be set to 1 to gain extra timing margin on the interface at the cost of address/ command bandwidth.	DDR SDRAM control configuration (DDR_SDRAM_CFG)
ODT_CFG	ODT Configuration	Can be set for termination at the IOs according to system topology. Typically, if ODT is enabled, then the internal IOs should be set up for termination only during reads to DRAM.	DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)
OBC_CFG	On-The-Fly Burst Chop Configuration	Can be set to 1 if on-the-fly burst chop will be used. This feature can only be used if a 64-bit data bus is used.	DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)

Table continues on the next page...

Table 12-18. Programming Summary, DDR4 memory types (continued)

Parameter	Description	Summary	Section
RWT	Read-to-write turnaround for same chip select (in TIMING_CFG_4)	This can be used to force a longer read-to-write turnaround time when accessing the same chip select. This is useful for burst chop mode, as there are some timing requirements to the same chip select that still must be met.	DDR SDRAM timing configuration 4 (TIMING_CFG_4)
WRT	Write-to-read turnaround for same chip select (in TIMING_CFG_4)	This could be used to force a certain turnaround time between a write and read to the same chip select. This is useful for burst chop mode. However, it is expected that TIMING_CFG_1[WRTORD] will be programmed appropriately such that TIMING_CFG_4[WRT] can be set to 0000.	DDR SDRAM timing configuration 4 (TIMING_CFG_4)
RRT	Read-to-read turnaround for same chip select (in TIMING_CFG_4)	Should typically be set to 0010 in burst chop mode (on-the-fly or fixed).	DDR SDRAM timing configuration 4 (TIMING_CFG_4)
WWT	Write-to-write turnaround for same chip select (in TIMING_CFG_4)	Should typically be set to 0010 in burst chop mode (on-the-fly or fixed).	DDR SDRAM timing configuration 4 (TIMING_CFG_4)
ZQ_EN	ZQ Calibration Enable	Should be set to 1. The other fields in DDR_ZQ_CNTL should also be programmed appropriately based on the DRAM specifications.	DDR SDRAM timing configuration 7 (TIMING_CFG_7)
WRLVL_EN	Write Leveling Enable	Should be set to 1 as write leveling is recommended. All other fields in DDR_WRLVL_CNTL, DDR_WRLVL_CNTRL_2, and DDR_WRLVL_CNTL_3 should be programmed appropriately based on the DRAM specifications and board layout specifics.	DDR write leveling control (DDR_WRLVL_CNTL)
BSTOPRE	Burst To Precharge Interval	Can be set to any value, depending on the application. Auto precharge can be enabled by setting this field to all 0s.	DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL)

12.6.2 DDR SDRAM Initialization Sequence

After configuration of all parameters is complete, system software must set DDR_SDRAM_CFG[MEM_EN] to enable the memory interface. Note that 500 µs must elapse after DRAM clocks are stable (after DDRCDR_1[DHC_EN] is set, DDR_SDRAM_CLK_CNTL[CLK_ADJUST] is set, and any chip select is enabled) before MEM_EN can be set. A delay loop in the initialization code may be necessary if software is enabling the memory controller. If DDR_SDRAM_CFG[BI] is not set, the

DDR memory controller will conduct an automatic initialization sequence to the memory, which will follow the memory specifications. If the bypass initialization mode is used, then software can initialize the memory through the DDR_SDRAM_MD_CNTL register.

12.6.3 Using Forced Self-Refresh Mode to Implement a Battery-Backed RAM System

This section describes the options offered by this device to support battery-backed main memory.

12.6.3.1 Software Based Self-Refresh Scheme

The DDR controller also has a software-programmable bit, DDR_SDRAM_CFG_2[FRC_SR], that immediately puts main memory into self-refresh mode. See DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2) for a description of this register.

It is expected that a critical interrupt routine triggered by an external voltage sensing device will have time to set this bit.

12.6.3.2 Bypassing Re-initialization During Battery-Backed Operation

The DDR controller offers an initialization bypass feature (DDR_SDRAM_CFG[BI]), which system designers may use to prevent re-initialization of main memory during system power-on following an abnormal shutdown. See DDR SDRAM control configuration (DDR_SDRAM_CFG) for information on this bit and DDR training initialization address (DDR_INIT_ADDR) for a discussion of avoiding possible ECC errors in this mode.

Note that when this mode is used, the controller will wait a user-programmable number of cycles before issuing any command after the assertion of MCKEn; this number is set in TIMING_CFG_4[DLL_LOCK].

Chapter 13
DUART

13.1 The DUART module as implemented on the chip

This section provides details about how the DUART module is integrated into this chip.

The chip implements two DUART modules, DUART1 and DUART2. The DUART1 module contains UART1 and UART2 and the DUART2 module contains UART3 and UART4.

Table 13-1. DUART module as implemented on chip

Module name	Module base address	Comments
DUART1	021C_0000h	UART1 0x21C_0500 UART2 0x21C_0600
DUART2	021D_0000h	UART3 0x21D_0500 UART4 0x21D_0600

13.2 Overview

This chapter describes the dual universal asynchronous receiver/transmitters (DUART). It describes the functional operation, the initialization sequence, and the programming details for the dual UART registers and features.

The dual UART consists of two universal asynchronous receiver/transmitters (UARTs).The UARTs act independently; all references to UART refer to one of these receiver/transmitters. Each UART is clocked by the platform clock/2. The dual UART programming model is compatible with the PC16552D.

The UART interface is point to point, meaning that only two UART devices are attached to the connecting signals. As shown in the figure below, each UART module consists of the following:

- Receive and transmit buffers
- Clear to send (CTS_B) input port and request to send (RTS_B) output port for data flow control
- 16-bit counter for baud rate generation
- Interrupt control logic

Figure 13-1. UART block diagram

13.2.1 Features

The DUART includes these distinctive features:

- Full-duplex operation
- Programming model compatible with original PC16450 UART and PC16550D (improved version of PC16450 that also operates in FIFO mode)
- PC16450 register reset values
- Configurable FIFO mode for both transmitter and receiver, providing 64-byte FIFOs
- Serial data encapsulation and decapsulation with standard asynchronous communication bits (START, STOP, and parity)
- Maskable transmit, receive, line status, and modem status interrupts
- Software-programmable baud generators that divide the platform clock/2 by 1 to (2^{16} - 1) and generate a 16x clock for the transmitter and receiver engines
- Clear to send (CTS_B and ready to send (RTS_B) modem control functions

- Auto flow for clear to send (CTS_B) and ready to send (RTS_B) modem control functions
- Software-selectable serial interface data format (data length, parity, 1/1.5/2 STOP bit, baud rate)
- Line and modem status registers
- Line-break detection and generation
- Internal diagnostic support, local loopback, and break functions
- Prioritized interrupt reporting
- Overrun, parity, and framing error detection

13.2.2 Modes of operation

The communication channel provides a full-duplex asynchronous receiver and transmitter using an operating frequency derived from the one-half of the platform clock .

The transmitter accepts parallel data from a write to the transmitter holding register (UTHR). In FIFO mode, the data is placed directly into an internal transmitter shift register of the transmitter FIFO. The transmitter converts the data to a serial bit stream inserting the appropriate start, stop, and optional parity bits. Finally, it outputs a composite serial data stream on the channel transmitter serial data output signal (SOUT). The transmitter status may be polled or interrupt driven.

The receiver accepts serial data bits on the channel receiver serial data input signal (SIN), converts it to parallel format, checks for a start bit, parity (if any), stop bits, and transfers the assembled character (with start, stop, parity bits removed) from the receiver buffer (or FIFO) in response to a read of the UART's receiver buffer register (URBR). The receiver status may be polled or interrupt driven.

13.3 DUART external signal descriptions

The DUART signals are described in the table below.

NOTE

Although the actual device signal names are prepended with the UART_ prefix as shown in the table, the abbreviated signal names are often used throughout this chapter.

Table 13-2. DUART signals-detailed signal descriptions

Signal	I/O	Description	
UART*n*_SIN	I	Serial data in. Data is received on the receivers of UART*n* through the respective serial data input signal, with the least-significant bit received first. Note that UART3_SIN is not available when UART1 is configured for CTS_B mode and UART4_SIN is not available when UART2 is configured for CTS_B mode.	
		State meaning	Asserted/Negated-Represents the data being received on the UART interface.
		Timing	Assertion/Negation-An internal logic sample signal, *rxcnt*, uses the frequency of the baud-rate generator to sample the data on SIN.
UART*n*_SOUT	O	Serial data out. The serial data output signals for the UART*n* are set ('mark' condition) when the transmitter is disabled, operating in the local loopback mode, or idle. Data is shifted out on these signals, with the least significant bit transmitted first. Note that UART3_SOUT is not available when UART1 is configured for RTS_B mode and UART4_SOUT is not available when UART2 is configured for RTS_B mode.	
		State meaning	Asserted/Negated-Represents the data being transmitted on the respective UART interface.
		Timing	Assertion/Negation- An internal logic sample signal, *rxcnt*, uses the frequency of the baud-rate generator to update and drive the data on SOUT.
UART*n*_CTS_B	I	Clear to send. These active-low inputs are the clear-to-send inputs. They are connected to the respective RTS_B outputs of the other UART devices on the bus. They can be programmed to generate an interrupt on change-of-state of the signal. Note that UART1_CTS_B is not available when UART3 is configured for SIN mode and UART2_CTS_B is not available when UART4 is configured for SIN mode.	
		State meaning	Asserted/Negated-Represent the clear to send condition for their respective UART.
		Timing	Assertion/Negation-Sampled at the rising edge of every input clock (platform clock/2) .
UART*n*_RTS_B	O	Request to send. UART*n*_RTS_B are active-low output signals that can be programmed to be negated and asserted by either the receiver or transmitter. When connected to the clear-to-send (CTS_B) input of a transmitter, this signal can be used to control serial data flow. Note that UART1_RTS_B is not available when UART3 is configured for SOUT mode and UART2_RTS_B is not available when UART4 is configured for SOUT mode.	
		State meaning	Asserted/Negated-Represents the data being transmitted on the respective UART interface.
		Timing	Assertion/Negation-Updated and driven at the rising edge of every input clock (platform clock/2) .

13.4 DUART register descriptions

The table below lists the DUART registers and their offsets. It lists the address, name, and a cross-reference to the complete description of each register.

The UARTs on the device are identical, except that the registers for each UART are located at different offsets. Throughout this chapter, the registers are described by a singular acronym: for example, LCR represents the line control register for each UART (see "The DUART module as implemented on the chip" section for the number of UARTs supported on chip).

The registers in each UART interface are used for configuration, control, and status. The divisor latch access bit, ULCR[DLAB], is used to access the divisor latch least- and most-significant bit registers and the alternate function register. Refer to UART line control register (ULCR1 - ULCR2) , for more information on ULCR[DLAB].

All the DUART registers are one-byte wide. Reads and writes to these registers must be byte-wide operations. The table below provides a register summary with references to the section and page that contains detailed information about each register. Undefined byte address spaces within offset 0x000-0xFFF are reserved.

13.4.1 DUART memory map

DUART1 base address: 21C_0000h

DUART2 base address: 21D_0000h

Offset	Register	Width (In bits)	Access	Reset value
500h	UART divisor least significant byte register (UDLB1)	8	RW	00h
500h	UART receiver buffer register (URBR1)	8	RO	00h
500h	UART transmitter holding register (UTHR1)	8	WO	00h
501h	UART divisor most significant byte register (UDMB1)	8	RW	00h
501h	UART interrupt enable register (UIER1)	8	RW	00h
502h	UART alternate function register (UAFR1)	8	RW	00h
502h	UART FIFO control register (UFCR1)	8	WO	00h
502h	UART interrupt ID register (UIIR1)	8	RO	01h
503h	UART line control register (ULCR1)	8	RW	00h
504h	UART modem control register (UMCR1)	8	RW	00h
505h	UART line status register (ULSR1)	8	RO	60h
506h	UART modem status register (UMSR1)	8	RO	00h
507h	UART scratch register (USCR1)	8	RW	00h
510h	UART DMA status register (UDSR1)	8	RO	01h
600h	UART divisor least significant byte register (UDLB2)	8	RW	00h
600h	UART receiver buffer register (URBR2)	8	RO	00h
600h	UART transmitter holding register (UTHR2)	8	WO	00h
601h	UART divisor most significant byte register (UDMB2)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
601h	UART interrupt enable register (UIER2)	8	RW	00h
602h	UART alternate function register (UAFR2)	8	RW	00h
602h	UART FIFO control register (UFCR2)	8	WO	00h
602h	UART interrupt ID register (UIIR2)	8	RO	01h
603h	UART line control register (ULCR2)	8	RW	00h
604h	UART modem control register (UMCR2)	8	RW	00h
605h	UART line status register (ULSR2)	8	RO	60h
606h	UART modem status register (UMSR2)	8	RO	00h
607h	UART scratch register (USCR2)	8	RW	00h
610h	UART DMA status register (UDSR2)	8	RO	01h

13.4.2 UART divisor least significant byte register (UDLB1 - UDLB2)

13.4.2.1 Offset

Register	Offset
UDLB1	500h
UDLB2	600h

13.4.2.2 Function

This register is accessible when ULCR[DLAB] = 1.

The divisor least significant byte register (UDLB) is concatenated with the divisor most significant byte register (UDMB) to create the divisor used to divide the input clock into the DUART. The output frequency of the baud generator is 16 times the baud rate; therefore the desired baud rate = platform clock frequency/2 ÷ (16 x [UDMB||UDLB]). Equivalently, [UDMB||UDLB:0b0000] = platform clock frequency/2 ÷ desired baud rate. Baud rates that can be generated by specific input clock frequencies are shown in the table below.

The following table shows examples of baud rate generation based on common input clock frequencies. Many other target baud rates are also possible.

NOTE

Because only integer values can be used as divisors, the actual baud rate differs slightly from the desired (target) baud rate; for this reason, both target and actual baud rates are given, along with the percentage of error.

Table 13-3. Baud Rate Examples

Target Baud Rate (Decimal)	Divisor		Platform Cock/2 Frequency (MHz)	Actual Baud Rate (Decimal)	Percent Error (Decimal)
	Decimal	Hex			
9,600	1736	6C8	266	9600.61444	0.0064
19,200	868	364	266	19,201.22888	0.0064
38,400	434	1B2	266	38,402.45776	0.0064
57,600	289	121	266	57,670.12673	0.1217
115,200	145	91	266	114,942.52844	0.2235
230,400	72	48	266	231,481.48090	0.4694
9,600	1953	07A1	300	9600.61444	0.0064
19,200	977	03D1	300	19,191.40225	0.0448
38,400	488	01E8	300	38,422.13115	0.0576
57,600	326	0146	300	57,515.33742	0.1470
115,200	163	00A3	300	115,030.67485	0.1470
230,400	81	0051	300	231,481.48148	0.4694
9,600	2170	87A	333	9600.61444	0.0064
19,200	1085	43D	333	19,201.22888	0.0064
38,400	543	21F	333	38,367.09638	0.0858
57,600	362	16A	333	57,550.64457	0.0857
115,200	181	B5	333	115,101.28913	0.0857
230,400	90	5A	333	231,481.48148	0.4694
9,600	2604	0A2C	400	9600.61444	0.0064
19,200	1302	0516	400	19,201.22888	0.0064
38,400	651	028B	400	38,402.45776	0.0064
57,600	434	01B2	400	57,603.68664	0.0064
115,200	217	00D9	400	115,207.37327	0.0064
230,400	109	006D	400	229,357.79817	0.4523
9,600	3472	D90	533	9600.61444	0.0064
19,200	1736	6C8	533	19,201.22888	0.0064
38,400	868	364	533	38,402.45776	0.0064
57,600	579	243	533	57,570.52389	0.0512
115,200	289	121	533	115,340.25375	0.1217
230,400	145	91	533	229,885.05747	0.2235

13.4.2.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	UDLB							
Reset	0	0	0	0	0	0	0	0

13.4.2.4 Fields

Field	Function
7-0 UDLB	Divisor least significant byte. This is concatenated with UDMB.

13.4.3 UART receiver buffer register (URBR1 - URBR2)

13.4.3.1 Offset

Register	Offset
URBR1	500h
URBR2	600h

13.4.3.2 Function

This register is accessible when ULCR[DLAB] = 0.

These registers contain the data received from the transmitter on the UART buses. In FIFO mode, when read, they return the first byte received. For FIFO status information, refer to the UDSR[RXRDY] description.

Except for the case when there is an overrun, URBR returns the data in the order it was received from the transmitter. Refer to the ULSR[OE] description, UART line status register (ULSR1 - ULSR2) . Note that these registers have same offset as the UTHRs.

13.4.3.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	DATA							
W								
Reset	0	0	0	0	0	0	0	0

13.4.3.4 Fields

Field	Function
7-0 DATA	Data received from transmitter. Data received from the transmitter on the UART bus (read only)

13.4.4 UART transmitter holding register (UTHR1 - UTHR2)

13.4.4.1 Offset

Register	Offset
UTHR1	500h
UTHR2	600h

13.4.4.2 Function

This register is accessible when ULCR[DLAB] = 0.

A write to these 8-bit registers causes the UART devices to transfer 5-8 data bits on the UART bus in the format set up in the ULCR (line control register). In FIFO mode, data written to UTHR is placed into the FIFO. The data written to UTHR is the data sent onto the UART bus, and the first byte written to UTHR is the first byte onto the bus. UDSR[TXRDY_B] indicates when the FIFO is full. See UART DMA status register (UDSR1 - UDSR2) for more details.

13.4.4.3 Diagram

Bits	7	6	5	4	3	2	1	0
R								
W	DATA							
Reset	0	0	0	0	0	0	0	0

13.4.4.4 Fields

Field	Function
7-0 DATA	Data that is written to UTHR (write only)

13.4.5 UART divisor most significant byte register (UDMB1 - UDMB2)

13.4.5.1 Offset

Register	Offset
UDMB1	501h
UDMB2	601h

13.4.5.2 Function

This register is accessible when ULCR[DLAB] = 1.

The divisor least significant byte register (UDLB) is concatenated with the divisor most significant byte register (UDMB) to create the divisor used to divide the input clock into the DUART. The output frequency of the baud generator is 16 times the baud rate; therefore the desired baud rate = platform clock frequency/2 ÷ (16 x [UDMB||UDLB]). Equivalently, [UDMB||UDLB:0b0000] = platform clock frequency/2 ÷ desired baud rate. Baud rates that can be generated by specific input clock frequencies are shown in UART divisor least significant byte register (UDLB1 - UDLB2).

13.4.5.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	UDMB							
Reset	0	0	0	0	0	0	0	0

13.4.5.4 Fields

Field	Function
7-0 UDMB	Divisor most significant byte

13.4.6 UART interrupt enable register (UIER1 - UIER2)

13.4.6.1 Offset

Register	Offset
UIER1	501h
UIER2	601h

13.4.6.2 Function

This register is accessible when ULCR[DLAB] = 0.

The UIER gives the user the ability to mask specific UART interrupts to the programmable interrupt controller (PIC).

13.4.6.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Reserved				EMSI	ERLSI	ETHREI	ERDAI
Reset	0	0	0	0	0	0	0	0

13.4.6.4 Fields

Field	Function
7-4 —	Reserved.
3 EMSI	Enable modem status interrupt. 0b - Mask interrupts caused by UMSR[DCTS] being set 1b - Enable and assert interrupts when the clear-to-send bit in the UART modem status register (UMSR) changes state
2 ERLSI	Enable receiver line status interrupt. 0b - Mask interrupts when ULSR's overrun, parity error, framing error or break interrupt bits are set 1b - Enable and assert interrupts when ULSR's overrun, parity error, framing error or break interrupt bits are set
1 ETHREI	Enable transmitter holding register empty interrupt. 0b - Mask interrupt when ULSR[THRE] is set 1b - Enable and assert interrupts when ULSR[THRE] is set
0 ERDAI	Enable received data available interrupt. 0b - Mask interrupt when new receive data is available or receive data time out has occurred 1b - Enable and assert interrupts when a new data character is received from the external device and/or a time-out interrupt occurs in the FIFO mode

13.4.7 UART alternate function register (UAFR1 - UAFR2)

13.4.7.1 Offset

Register	Offset
UAFR1	502h
UAFR2	602h

13.4.7.2 Function

This register is accessible when ULCR[DLAB] = 1.

The UAFRs give software the ability to gate off the baud clock and write to each UARTn registers simultaneously with the same write operation.

13.4.7.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Reserved						BO	CW
Reset	0	0	0	0	0	0	0	0

13.4.7.4 Fields

Field	Function
7-2 —	Reserved.
1 BO	Baud clock select. 0b - The baud clock is not gated off. 1b - The baud clock is gated off.
0 CW	Concurrent write enable. 0b - Disables writing to each UART*n* 1b - Enables concurrent writes to corresponding UART registers. A write to a register in UART*n* is also a write to the corresponding register in UART*n+1* and vice versa for each DUART where *n* refers to the number of UART controller. The user needs to ensure that the ULCR[DLAB] of each UART is in the same state before executing a concurrent write to register addresses 0x*m*00, 0x*m*01 and 0x*m*02, where *m* is the base address of the corresponding UART.

13.4.8 UART FIFO control register (UFCR1 - UFCR2)

13.4.8.1 Offset

Register	Offset
UFCR1	502h
UFCR2	602h

13.4.8.2 Function

This register is accessible when ULCR[DLAB] = 0.

The UFCR, a write-only register, is used to enable and clear the receiver and transmitter FIFOs, set a receiver FIFO trigger level to control the received data available interrupt, and select the type of DMA signaling.

When the UFCR bits are written, the FIFO enable bit must also be set or else the UFCR bits are not programmed. When changing from FIFO mode to 16450 mode (non-FIFO mode) and vice versa, data is automatically cleared from the FIFOs.

After all the bytes in the receiver FIFO are cleared, the receiver internal shift register is not cleared. Similarly, the bytes are cleared in the transmitter FIFO, but the transmitter internal shift register is not cleared. Both TFR and RFR are self-clearing bits.

13.4.8.3 Diagram

Bits	7	6	5	4	3	2	1	0
R				Reserved	Reserved			
W	RTL		EN64			TFR	RFR	FEN
Reset	0	0	0	0	0	0	0	0

13.4.8.4 Fields

Field	Function
7-6 RTL	Receiver trigger level. A received data available interrupt occurs when UIER[ERDAI] is set and the number of bytes in the receiver FIFO equals the designated interrupt trigger level as follows: 00b - 1 byte, if EN64 1 byte 01b - 4 bytes, if EN64 16 bytes 10b - 8 bytes, if EN64 32 bytes

Table continues on the next page...

Field	Function
	11b - 14 bytes, if EN64 56 bytes
5 EN64	Enable 64-byte FIFO 0b - Disables the 64-byte FIFOs 1b - Enables the 64-byte FIFOs
4 —	Reserved
3 —	Reserved
2 TFR	Transmitter FIFO reset 0b - No action 1b - Clears all bytes in the transmitter FIFO and resets the FIFO counter/pointer to 0
1 RFR	Receiver FIFO reset 0b - No action 1b - Clears all bytes in the receiver FIFO and resets the FIFO counter/pointer to 0
0 FEN	FIFO enable 0b - FIFOs are disabled and cleared 1b - Enables the transmitter and receiver FIFOs

13.4.9 UART interrupt ID register (UIIR1 - UIIR2)

13.4.9.1 Offset

Register	Offset
UIIR1	502h
UIIR2	602h

13.4.9.2 Function

This register is accessible when ULCR[DLAB] = 0.

The UIIRs indicate when an interrupt is pending from the corresponding UART and what type of interrupt is active. They also indicate if the FIFOs are enabled.

The DUART prioritizes interrupts into four levels and records these in the corresponding UIIR. The four levels of interrupt conditions in order of priority are:

1. Receiver line status
2. Received data ready/character time-out

3. Transmitter holding register empty
4. Modem status

When the UIIR is read, the associated DUART serial channel freezes all interrupts and indicates the highest priority pending interrupt. While this read transaction is occurring, the associated DUART serial channel records new interrupts, but does not change the contents of UIIR until the read access is complete.

Table 13-4. UIIR IID Bits Summary

IID Bits IID[3-0]	Priority Level	Interrupt Type	Interrupt Description	How To Reset Interrupt
0b0001	-	-	-	-
0b0110	Highest	Receiver line status	Overrun error, parity error, framing error, or break interrupt	Read the line status register.
0b0100	Second	Received data available	Receiver data available or trigger level reached in FIFO mode	Read the receiver buffer register or interrupt is automatically reset if the number of bytes in the receiver FIFO drops below the trigger level.
0b1100	Second	Character time-out	No characters have been removed from or input to the receiver FIFO during the last 4 character times and there is at least one character in the receiver FIFO during this time.	Read the receiver buffer register.
0b0010	Third	UTHR empty	Transmitter holding register is empty	Read the UIIR or write to the UTHR.
0b0000	Fourth	Modem status	CTS_B input value changed since last read of UMSR	Read the UMSR.

13.4.9.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	F E		FE64	Reserved	IID3	IID2	IID1	IID0
W								
Reset	0	0	0	0	0	0	0	1

13.4.9.4 Fields

Field	Function
7-6 FE	FIFOs enabled. Reflects the setting of UFCR[FEN]
5 FE64	64-byte FIFOs enabled. Reflects the setting of UFCR[EN64]. 0b - 64-byte FIFOs disabled 1b - 64-byte FIFOs enabled
4 —	Reserved
3 IID3	Interrupt ID bits identify the highest priority interrupt that is pending as indicated in the table above. IID3 is set along with IID2 only when a timeout interrupt is pending for FIFO mode.
2 IID2	Interrupt ID bits identify the highest priority interrupt that is pending as indicated in the table above.
1 IID1	Interrupt ID bits identify the highest priority interrupt that is pending as indicated in the table above.
0 IID0	IID0 indicates when an interrupt is pending. 0b - The UART has an active interrupt ready to be serviced. 1b - No interrupt is pending.

13.4.10 UART line control register (ULCR1 - ULCR2)

13.4.10.1 Offset

Register	Offset
ULCR1	503h
ULCR2	603h

13.4.10.2 Function

This register is accessible when ULCR[DLAB] = x.

The ULCRs specify the data format for the UART bus and set the divisor latch access bit ULCR[DLAB], which controls the ability to access the divisor latch least and most significant bit registers and the alternate function register.

After initializing the ULCR, the software should not re-write the ULCR when valid transfers on the UART bus are active. The software should not re-write the ULCR until the last STOP bit has been received and there are no new characters being transferred on the bus.

The stick parity bit, ULCR[SP], assigns a set parity value for the parity bit time slot sent on the UART bus. The set value is defined as mark parity (logic 1) or space parity (logic 0). ULCR[PEN] and ULCR[EPS] help determine the set parity value. See the table below for more information. ULCR[NSTB], defines the number of STOP bits to be sent at the end of the data transfer. The receiver only checks the first STOP bit, regardless of the number of STOP bits selected. The word length select bits (1 and 0) define the number of data bits that are transmitted or received as a serial character. The word length does not include START, parity, and STOP bits.

Table 13-5. Parity Selection Using ULCR[PEN], ULCR[SP], and ULCR[EPS]

PEN	SP	EPS	Parity Selected
0	0	0	No parity
0	0	1	No parity
0	1	0	No parity
0	1	1	No parity
1	0	0	Odd parity
1	0	1	Even parity
1	1	0	Mark parity
1	1	1	Space parity

13.4.10.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	DLAB	SB	SP	EPS	PEN	NSTB	WLS	
Reset	0	0	0	0	0	0	0	0

13.4.10.4 Fields

Field	Function
7 DLAB	Divisor latch access bit. 0b - Access to all registers except UDLB, UAFR, and UDMB

Table continues on the next page...

Field	Function
	1b - Ability to access divisor latch least and most significant byte registers and alternate function register (UAFR)
6 SB	Set break. 0b - Send normal UTHR data onto the serial output (SOUT) signal 1b - Force logic 0 to be on the SOUT signal. Data in the UTHR is not affected
5 SP	Stick parity. 0b - Stick parity is disabled. 1b - If PEN = 1 and EPS = 1, space parity is selected. And if PEN = 1 and EPS = 0, mark parity is selected.
4 EPS	Even parity select. See the table above for more information. 0b - If PEN = 1 and SP = 0, odd parity is selected. 1b - If PEN = 1 and SP = 0, even parity is selected.
3 PEN	Parity enable. 0b - No parity generation and checking 1b - Generate parity bit as a transmitter, and check parity as a receiver
2 NSTB	Number of STOP bits. 0b - One STOP bit is generated in the transmitted data. 1b - When a 5-bit data length is selected, 1½ STOP bits are generated. When either a 6-, 7-, or 8-bit word length is selected, two STOP bits are generated.
1-0 WLS	Word length select. Number of bits that comprise the character length. The word length select values are as follows: 00b - 5 bits 01b - 6 bits 10b - 7 bits 11b - 8 bits

13.4.11 UART modem control register (UMCR1 - UMCR2)

13.4.11.1 Offset

Register	Offset
UMCR1	504h
UMCR2	604h

13.4.11.2 Function

This register is accessible when ULCR[DLAB] = x.

The UMCRs control the interface with the external peripheral device on the UART bus.

13.4.11.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Reserved		AFE	LOOP	Reserved		RTS	Reserved
Reset	0	0	0	0	0	0	0	0

13.4.11.4 Fields

Field	Function
7-6 —	Reserved.
5 AFE	Auto Flow Control Enable Setting this bit to 1 enables the UART's autflow. RTS AFE Auto-flow configuration: 0b - When RTS is either 0 or 1, both Auto-RTS and Auto-CTS are disabled. 1b - When RTS is 0, only Auto-CTS is enabled. When RTS is 1, both Auto-CTS and Auto-RTS are enabled.
4 LOOP	Local loopback mode. 0b - Normal operation 1b - Functionally, the data written to UTHR can be read from URBR of the same UART and UMCR[RTS] is tied to UMSR[CTS].
3-2 —	Reserved.
1 RTS	Ready to send. 0b - Negates corresponding RTS_B output 1b - Assert corresponding RTS_B output. Informs external modem or peripheral that the UART is ready for sending/receiving data
0 —	Reserved.

13.4.12 UART line status register (ULSR1 - ULSR2)

13.4.12.1 Offset

Register	Offset
ULSR1	505h
ULSR2	605h

13.4.12.2 Function

This register is accessible when ULCR[DLAB] = x.

The ULSRs are read-only registers that monitor the status of the data transfer on the UART buses. To isolate the status bits from the proper character received through the UART bus, software should read the ULSR and then the URBR.

13.4.12.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	RFE	TEMT	THRE	BI	FE	PE	OE	DR
W								
Reset	0	1	1	0	0	0	0	0

13.4.12.4 Fields

Field	Function
7 RFE	Receiver FIFO error. 0b - This bit is cleared when there are no errors in the receiver FIFO or on a read of the ULSR with no remaining receiver FIFO errors. 1b - Set to one when one of the characters in the receiver FIFO encounters an error (framing, parity, or break interrupt)
6 TEMT	Transmitter empty. 0b - Either or both the UTHR or the internal transmitter shift register has a data character. In FIFO mode, a data character is in the transmitter FIFO or the internal transmitter shift register. 1b - Both the UTHR and the internal transmitter shift register are empty. In FIFO mode, both the transmitter FIFO and the internal transmitter shift register are empty.
5 THRE	Transmitter holding register empty. 0b - The UTHR is not empty.

Table continues on the next page...

Field	Function
	1b - A data character has transferred from the UTHR into the internal transmitter shift register. In FIFO mode, the transmitter FIFO contains no data character.
4 BI	Break interrupt. **NOTE:** For a single break signal, BI and DR are set multiple times, approximately once every character period. The BI and DR bits continue to be set each character period after they are cleared. This continues for the entire duration of the break signal. To accommodate this behavior, read URBR, which returns zeros and clears DR. Then delay one character period and read URBR again. Note that at the end of the break signal, a random character may be falsely detected and received in the URBR, with ULSR[DR] being set. 0b - This bit is cleared when the ULSR is read or when a valid data transfer is detected (that is, STOP bit is received). 1b - Received data of logic 0 for more than START bit + Data bits + Parity bit + one STOP bits length of time. A new character is not loaded until SIN returns to the mark state (logic 1) and a valid START is detected. In FIFO mode, a zero character is encountered in the FIFO (the zero character is at the top of the FIFO). In FIFO mode, only one zero character is stored.
3 FE	Framing error. 0b - This bit is cleared when ULSR is read or when a new character is loaded into the URBR from the receiver shift register. 1b - Invalid STOP bit for receive data (only the first STOP bit is checked). In FIFO mode, this bit is set when the character that detected a framing error is encountered in the FIFO (that is the character at the top of the FIFO). An attempt to resynchronize occurs after a framing error. The UART assumes that the framing error (due to a logic 0 being read when a logic 1 (STOP) was expected) was due to a STOP bit overlapping with the next START bit, so it assumes this logic 0 sample is a true START bit and then receives the following new data.
2 PE	Parity error. 0b - This bit is cleared when ULSR is read or when a new character is loaded into the URBR. 1b - Unexpected parity value encountered when receiving data. In FIFO mode, the character with the error is at the top of the FIFO .
1 OE	Overrun error. 0b - This bit is cleared when ULSR is read. 1b - Before the URBR is read, the URBR was overwritten with a new character. The old character is loss. In FIFO mode, the receiver FIFO is full (regardless of the receiver FIFO trigger level setting) and a new character has been received into the internal receiver shift register. The old character was overwritten by the new character. Data in the receiver FIFO was not overwritten.
0 DR	Data ready. 0b - This bit is cleared when URBR is read or when all of the data in the receiver FIFO is read. 1b - A character has been received in the URBR or the receiver FIFO.

13.4.13 UART modem status register (UMSR1 - UMSR2)

13.4.13.1 Offset

Register	Offset
UMSR1	506h
UMSR2	606h

13.4.13.2 Function

This register is accessible when ULCR[DLAB] = x.

The UMSRs track the status of the modem (or external peripheral device) clear to send (CTS_B) signal for the corresponding UART.

13.4.13.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Reserved			CTS	Reserved			DCTS
W								
Reset	0	0	0	0	0	0	0	0

13.4.13.4 Fields

Field	Function
7-5 —	Reserved.
4 CTS	Clear to send. Represents the inverted value of the CTS_B input pin from the external peripheral device 0b - Corresponding UART*n*_CTS_B is negated 1b - Corresponding UART*n*_CTS_B is asserted. The modem or peripheral device is ready for data transfers.
3-1 —	Reserved.
0 DCTS	Clear to send. 0b - No change on the corresponding UART*n*_CTS_B signal since the last read of UMSR[CTS] 1b - The UART*n*_CTS_B value has changed, since the last read of UMSR[CTS]. Causes an interrupt if UIER[EMSI] is set to detect this condition

13.4.14 UART scratch register (USCR1 - USCR2)

13.4.14.1 Offset

Register	Offset
USCR1	507h
USCR2	607h

13.4.14.2 Function

This register is accessible when ULCR[DLAB] = x.

The USCR registers are for debugging software or the DUART hardware. The USCRs do not affect the operation of the DUART.

13.4.14.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	DATA							
Reset	0	0	0	0	0	0	0	0

13.4.14.4 Fields

Field	Function
7-0 DATA	Data

13.4.15 UART DMA status register (UDSR1 - UDSR2)

13.4.15.1 Offset

Register	Offset
UDSR1	510h
UDSR2	610h

13.4.15.2 Function

This register is accessible when ULCR[DLAB] = x.

The DMA status registers (UDSRs) are read-only registers that return transmitter and receiver FIFO status. UDSRs also provide the ability to assist DMA data operations to and from the FIFOs.

Table 13-6. UDSR[TXRDY] Set Conditions

DMS	FEN	DMA Mode	Meaning
0	0	0	TXRDY is set after the first character is loaded into the transmitter FIFO or UTHR.
0	1	0	
1	0	0	
1	1	1	TXRDY is set when the transmitter FIFO is full.

Table 13-7. UDSR[TXRDY] Cleared Conditions

DMS	FEN	DMA Mode	Meaning
0	0	0	TXRDY is cleared when there are no characters in the transmitter FIFO or UTHR.
0	1	0	
1	0	0	
1	1	1	TXRDY is cleared when there are no characters in the transmitter FIFO or UTHR. TXRDY remains clear when the transmitter FIFO is not yet full.

Table 13-8. UDSR[RXRDY] Set Conditions

DMS	FEN	DMA Mode	Meaning
0	0	0	RXRDY is set when there are no characters in the receiver FIFO or URBR.
0	1	0	
1	0	0	
1	1	1	RXRDY is set when the trigger level has not been reached and there has been no time out.

Table 13-9. UDSR[RXRDY] Cleared Conditions

DMS	FEN	DMA Mode	Meaning
0	0	0	RXRDY is cleared when there is at least one character in the receiver FIFO or URBR.
0	1	0	
1	0	0	
1	1	1	RXRDY is cleared when the trigger level or a time-out has been reached. RXRDY remains cleared until the receiver FIFO is empty.

13.4.15.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Reserved						TXRDY	RXRDY
W								
Reset	0	0	0	0	0	0	0	1

13.4.15.4 Fields

Field	Function
7-2 —	Reserved
1 TXRDY	Transmitter ready. This read-only bit reflects the status of the transmitter FIFO or the UTHR. The status depends on the DMA mode selected, which is determined by the DMS and FEN bits in the UFCR. 0b - This bit is cleared, as shown in Table 13-7 . 1b - This bit is set, as shown in Table 13-6 .
0 RXRDY	Receiver ready. This read-only bit reflects the status of the receiver FIFO or URBR. The status depends on the DMA mode selected, which is determined by the DMS and FEN bits in the UFCR. 0b - This bit is cleared, as shown in Table 13-9 . 1b - This bit is set, as shown in Table 13-8 .

13.5 Functional description

The following sections provide the function of the DUART controller.

13.5.1 Serial interface

The UART bus is a serial, full-duplex, point-to-point bus as shown in the following figure. Therefore, only two devices are attached to the same signals and there is no need for address or arbitration bus cycles.

Figure 13-2. UART bus interface transaction protocol example

A standard UART bus transfer is composed of either three or four parts:

- START bit
- Data transfer bits (least-significant bit is first data bit on the bus)
- Parity bit (optional)
- STOP bits

An internal logic sample signal, *rxcnt*, uses the frequency of the baud-rate generator to drive the bits on SOUT.

The following sections describe the four components of the serial interface, the baud-rate generator, local loopback mode, different errors, and FIFO mode.

13.5.1.1 START bit

A write to the transmitter holding register (UTHR) generates a START bit on the SOUT signal.

Figure 13-2 the figure shows that the START bit is defined as a logic 0. The START bit denotes the beginning of a new data transfer which is limited to the bit length programmed in the UART line control register (ULCR). When the bus is idle, SOUT is high.

13.5.1.2 Data transfer

Each data transfer contains 5-8 bits of data.

The ULCR data bit length for the transmitter and receiver UART devices must agree before a transfer begins; otherwise, a parity or framing error may occur. A transfer begins when UTHR is written. At that time a START bit is generated followed by 5-8 of the data bits previously written to the UTHR. The data bits are driven from the least significant to the most significant bits. After the parity and STOP bits, a new data transfer can begin if new data is written to the UTHR.

13.5.1.3 Parity bit

The user has the option of using even, odd, no parity, or stick parity.

See UART line control register (ULCR1 - ULCR2). Both the receiver and transmitter parity definition must agree before attempting to transfer data. When receiving data, a parity error can occur if an unexpected parity value is detected. See UART line status register (ULSR1 - ULSR2).

13.5.1.4 STOP bit

The transmitter device ends the write transfer by generating a STOP bit.

The STOP bit is always high. The length of the STOP bit(s) can be programmed in the ULCR. Both the receiver and transmitter STOP bit length must agree before attempting to transfer data. A framing error can occur if an invalid STOP bit is detected.

13.5.2 Baud-rate generator logic

Each UART contains an independent programmable baud-rate generator, that is capable of taking the platform clock frequency/2 as input and dividing the input by any divisor from 1 to 2^{16} - 1.

The baud rate is defined as the number of bits per second that can be sent over the UART bus. The formula for calculating baud rate is as follows:

Baud rate = (1/16) x ((platform clock/2) frequency ÷ divisor value)

Therefore, the output frequency of the baud-rate generator is 16 times the baud rate.

The divisor value is determined by the following two 8-bit registers to form a 16-bit binary number:

- UART divisor most significant byte register (UDMB)
- UART divisor least significant byte register (UDLB)

Upon loading either of the divisor latches, a 16-bit baud-rate counter is loaded.

The divisor latches must be loaded during initialization to ensure proper operation of the baud-rate generator. Both UART devices on the same bus must be programmed for the same baud-rate before starting a transfer.

The baud clock can be passed to the performance monitor by enabling the UAFR[BO] bit. This can be used to determine baud rate errors.

13.5.3 Local loopback mode

Local loopback mode is provided for diagnostic testing.

The data written to UTHR can be read from the receiver buffer register (URBR) of the same UART. In this mode, the modem control register UMCR[RTS] is internally tied to the modem status register UMSR[CTS]. The transmitter SOUT is set to a logic 1 and the receiver SIN is disconnected. The output of the transmitter shift register is looped back into the receiver shift register input. The CTS_B (input signal) is disconnected, RTS_B is internally connected to CTS_B, and the RTS_B (output signal) becomes inactive. In this diagnostic mode, data that is transmitted is immediately received. In local loopback mode, the transmit and receive data paths of the DUART can be verified.

NOTE

In local loopback mode, the transmit/receive interrupts are fully operational and can be controlled by the interrupt enable register (UIER).

13.5.4 Errors

The following sections describe framing, parity, and overrun errors which may occur while data is transferred on the UART bus.

Each of the error bits are usually cleared, as described below, when the line status register (ULSR) is read.

13.5.4.1 Framing error

When an invalid STOP bit is detected, a framing error occurs and ULSR[FE] is set.

Note that only the first STOP bit is checked. In FIFO mode, ULSR[FE] is set when the character at the top of the FIFO detects a framing error. An attempt to re-synchronize occurs after a framing error. The UART assumes that the framing error (due to a logic 0 being read when a logic 1 (STOP) was expected) was due to a STOP bit overlapping with the next START bit. ULSR[FE] is cleared when ULSR is read or when a new character is loaded into the URBR from the receiver shift register.

13.5.4.2 Parity error

A parity error occurs, and ULSR[PE] is set, when unexpected parity values are encountered while receiving data.

In FIFO mode, ULSR[PE] is set when the character with the error is at the top of the FIFO. ULSR[PE] is cleared when ULSR is read or when a new character is loaded into the URBR.

13.5.4.3 Overrun error

When a new (overwriting character) STOP bit is detected and the old character is lost, an overrun error occurs and ULSR[OE] is set.

In FIFO mode, ULSR[OE] is set after the receiver FIFO is full (despite the receiver FIFO trigger level setting) and a new character has been received into the internal receiver shift register. Data in the FIFO is not overwritten; only the shift register data is overwritten. Therefore, the interrupt occurs immediately. ULSR[OE] is cleared when ULSR is read.

13.5.5 FIFO mode

The UARTs use an alternate mode (FIFO mode) to relieve the processor core from excessive software overhead.

The FIFO control register (UFCR) is used to enable and clear the receiver and transmitter FIFOs and set the FIFO receiver trigger level UFCR[RTL] to control the received data available interrupt UIER[ERDAI].

The UFCR also selects the type of DMA signaling. The UDSR[RXRDY] indicates the status of the receiver FIFO. The DMA status registers (UDSR[TXRDY]) indicate when the transmitter FIFO is full. When in FIFO mode, data written to UTHR is placed into the transmitter FIFO. The first byte written to UTHR is the first byte onto the UART bus.

13.5.5.1 FIFO interrupts

In FIFO mode, the UIER[ERDAI] is set when a time-out interrupt occurs. When a receive data time-out occurs there is a maskable interrupt condition (through UIER[ERDAI]). See UART interrupt enable register (UIER1 - UIER2) for more details on interrupt enables.

The interrupt ID register (UIIR) indicates if the FIFOs are enabled. The interrupt ID3 bit UIIR[IID3] is only set for FIFO mode interrupts. The character time-out interrupt occurs when no characters have been removed from or input to the receiver FIFO during the last four character times and there is at least one character in the receiver FIFO during this time. The character time-out interrupt (controlled by UIIR[IID*n*]) is cleared when the URBR is read. See UART interrupt ID register (UIIR1 - UIIR2) for more information.

The UIIR[FE] bit indicates if FIFO mode is enabled.

13.5.5.2 Interrupt control logic

An interrupt is active when DUART interrupt ID register bit 7 (UIIR[IID0]), is cleared.

The interrupt enable register (UIER) is used to mask specific interrupt types. See UART interrupt enable register (UIER1 - UIER2) for more details.

When the interrupts are disabled in UIER, polling software cannot use UIIR[IID0] to determine whether the UART is ready for service. The software must monitor the appropriate bits in the line status (ULSR) and/or the modem status registers (UMSR). UIIR[IID0] can be used for polling if the interrupts are enabled in UIER.

13.6 Initialization/Application information

The following requirements must be met for DUART accesses:

- All DUART registers must be mapped to a cache-inhibited and guarded area. (That is, the WIMG setting in the MMU needs to be 0b01X1.)
- All DUART registers are 1 byte wide. Reads and writes to these registers must be byte-wide operations.

A system reset puts the DUART registers to a default state. Before the interface can transfer serial data, the following initialization steps are recommended:

1. Update the programmable interrupt controller (PIC) DUART channel interrupt vector source registers.
2. Set data attributes and control bits in the ULCR, UFCR, UAFR, UMCR, UDLB, and UDMB.
3. Set the data attributes and control bits of the external modem or peripheral device.
4. Set the interrupt enable register (UIER).
5. To start a write transfer, write to the UTHR.
6. Poll UIIR, if the interrupts generated by the DUART are masked.

Chapter 14
Enhanced Secured Digital Host Controller (eSDHC)

14.1 Overview

The enhanced secured digital host controller (eSDHC) provides interface between the host system and the SD/SDIO cards and eMMC devices, as depicted in the figure below.

The eSDHC acts as a bridge, passing host bus transactions to the SD/SDIO cards and eMMC devices by sending commands and performing data accesses to/from the cards.

It handles the SD/SDIO/eMMC protocols at the transmission level.

Different types of cards supported by the eSDHC are described below:

- Secure digital (SD) card

  The secure digital (SD) card is an evolution over the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in the emerging audio and video consumer electronic devices. The physical form factor, pin assignments, and data transfer protocol are forward-compatible with the old MMC. The chip supports SDXC card.

- SDIO

  Under the SD protocol, the SD cards can be categorized as a memory card, I/O card, or combo card. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O card provides high-speed data I/O with low power consumption for mobile electronic devices. The combo card has both memory and I/O functions. For the sake of simplicity, the following figure does not show cards with reduced sizes.

In addition, the chip also supports the embedded devices, such as eMMC and eSDIO. Note that all references to SDIO are also applicable to eSDIO.

The eSDHC acts as a bridge, passing host bus transactions to SD/SDIO cards by sending commands and performing data accesses to or from the cards. It handles the SD/SDIO protocol at the transmission level. The figure below shows connection of the eSDHC.

Figure 14-1. System connection of the eSDHC

NOTE

The card slot is not valid for embedded devices.

The figure below is a high-level block diagram of eSDHC.

Figure 14-2. Enhanced secure digital host controller block diagram

14.1.1 eSDHC features summary

The features of the eSDHC module include the following:

- Conforms to the SD Host controller standard specification version 3.0, including test event register support
- Compatible with the eMMC system specification version 4.5
- Compatible with the SD memory card specification version 3.01, and supports the high capacity SD memory card

- Compatible with the SDIO card specification version 3.0
- Designed to work with SD memory, SDIO, SD combo, and their variants, such as mini and micro.
- eMMC high capacity support
- Supports 1- or 4-bit SD and SDIO modes, 1- or 4- or 8-bit eMMC modes
- Supports single-block and multi-block read, and write data transfer
- Supports block sizes of 1-2048 bytes
- Supports the mechanical write protect detection. In the case where write protect is enabled, the host will not initiate any write data command to the card
- Supports card detection from SDHC_CD_B
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO read wait and suspend resume operations
- Supports Auto CMD12 and Auto CMD23 for multi-block transfer
- Host can initiate command that does not use data lines, while data transfer is in progress
- Allows card to interrupt the host in 1 bit and 4 bit SDIO modes, also supports interrupt period
- Embodies a configurable 128 x 32 bit FIFO for read/write data
- Supports SDMA, ADMA1, and ADMA2 capabilities
- Host will send 80 idle SD clock cycles to card, which are needed during card power-up, if bit INITA in the system control register (SYSCTL) is set
- Supports SDIO asynchronous interrupt
- Supports clock divider with finer granularity, that is, with values 1,2,3....1024 or 1,2,4,8...2048
- Supports standard, high and extended capacity card types
- Supports SD UHS-1 speed modes: SDR12, SDR25, SDR50, SDR104, DDR50
- Supports eMMC high speed, HS200 and DDR mode

14.1.2 Modes and operations

14.1.2.1 Data transfer modes

The eSDHC can select the following modes for data transfer:

- SD 1 bit
- SD 4 bit
- eMMC 1 bit
- eMMC 4 bit
- eMMC 8 bit

- Identification mode
- eMMC full speed mode
- eMMC high speed mode
- SD/SDIO full speed mode
- SD/SDIO high speed mode
- SD/SDIO UHS-1 SDR12 mode
- SD/SDIO UHS-1 SDR25 mode
- SD/SDIO UHS-1 SDR50 mode
- SD/SDIO UHS-1 SDR104 mode
- SD/SDIO UHS-1 DDR50 mode
- eMMC HS200 mode
- eMMC DDR mode

NOTE

For the maximum supported speed of the modes, refer chip-specific datasheet.

14.2 External signals

14.2.1 External signals overview

The eSDHC has the following associated I/O signals:

- SDHC_CLK is an internally generated clock signal that drives the SD/SDIO card or eMMC device.
- SDHC_CLK_SYNC_OUT has same frequency as SDHC_CLK. It should be sent out on board for loop back from close to external card.
- SDHC_CLK_SYNC_IN has same frequency as SDHC_CLK. It is loopback input from close to card on board.
- SDHC_CMD I/O sends commands to and receives responses from the card.
- SDHC_DAT[7:0] performs data transfers between the eSDHC and the card. If the eSDHC is desired to support a 4 bit data transfer, DAT[7:4] can also be optional and tied high.
- SDHC_CD_B and SDHC_WP are card detection and write protection signals from the socket.
- SDHC_CMD_DIR is an output signal used to control the direction of CMD line in an external voltage translator.
- SDHC_DAT0_DIR is an output signal used to control the direction of DAT0 line in an external voltage translator.

- SDHC_DAT123_DIR is an output signal used to control the direction of DAT1-DAT3 lines in an external voltage translator.
- SDHC_VS is an output signal used to control the voltage of external SD bus supply.

14.2.2 eSDHC signal descriptions

The following table describes eSDHC signals.

Table 14-1. Signal properties

Name	Port	Function	Reset state	Pull up
SDHC_CLK	O	Clock for SD/SDIO card or eMMC device	0	N/A
SDHC_CLK_SYNC_OUT	O	Clock to loopback from close to card	0	N/A
SDHC_CLK_SYNC_IN	I	Clock looped back from card	0	N/A
SDHC_CMD	I/O	CMD line connect to card	1	Pull up
SDHC_CMD_DIR	O	CMD line direction control	0	N/A
SDHC_DAT7	I/O	DAT7 line in 8-bit mode Not used in other modes	1	Pull up
SDHC_DAT6	I/O	DAT6 line in 8-bit mode Not used in other modes	1	Pull up
SDHC_DAT5	I/O	DAT5 line in 8-bit mode Not used in other modes	1	Pull up
SDHC_DAT4	I/O	DAT4 line in 8-bit mode Not used in other modes	1	Pull up
SDHC_DAT3	I/O	DAT3 line in 4/8-bit mode	0	Pull up. Do not use DAT3 pin as a CD pin.
SDHC_DAT2	I/O	DAT2 line or read wait in 4/8-bit mode Read wait in 1-bit mode	1	Pull up
SDHC_DAT1	I/O	DAT1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	1	Pull up
SDHC_DAT0	I/O	DAT0 line in all modes Also used to detect busy state	1	Pull up
SDHC_DAT0_DIR	O	DAT0 line direction control	0	N/A
SDHC_CD_B	I	Card detection pin If not used tie high	N/A	N/A
SDHC_WP	I	Card write protect detect If not used tie low	N/A	N/A
SDHC_VS	O	Control the voltage on SD pads to be high voltage (around 3.0V) or low voltage (around 1.8V). '0' stands for high voltage range	0	N/A

Table continues on the next page...

Table 14-1. Signal properties (continued)

Name	Port	Function	Reset state	Pull up
		Optional output		
SDHC_DAT123_DIR	O	DAT1-DAT3 lines direction control	0	N/A

14.3 eSDHC register descriptions

The table below shows the memory-mapped registers of the eSDHC module and lists the offset, name, and a cross-reference to the complete description of each register. These register only support 32-bit accesses.

14.3.1 eSDHC memory map

eSDHC base address: 214_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	SDMA system address register/Block attributes 2 (DSADDR_BLKATTR2)	32	RW	0000_0000h
4h	Block attributes register (BLKATTR)	32	RW	0000_0000h
8h	Command argument register (CMDARG)	32	RW	0000_0000h
Ch	Transfer type register (XFERTYP)	32	RW	0000_0000h
10h	Command response 0 register (CMDRSP0)	32	RO	0000_0000h
14h	Command response 1 register (CMDRSP1)	32	RO	0000_0000h
18h	Command response 2 register (CMDRSP2)	32	RO	0000_0000h
1Ch	Command Response 3 register (CMDRSP3)	32	RO	0000_0000h
20h	Buffer data port register (DATPORT)	32	RW	0000_0000h
24h	Present state register (PRSSTAT)	32	RO	Table 14-4
28h	Protocol control register (PROCTL)	32	RW	0000_0020h
2Ch	System Control Register when ESDHCCTL[CRS=0] (SYSCTL_ESDHCCTL_CRS_0)	32	RW	0000_8038h
2Ch	System Control Register when ESDHCCTL[CRS=1] (SYSCTL_ESDHCCTL_CRS_1)	32	RW	0000_0000h
30h	Interrupt status register (IRQSTAT)	32	W1C	0000_0000h
34h	Interrupt status enable register (IRQSTATEN)	32	RW	377F_11FFh
38h	Interrupt signal enable register (IRQSIGEN)	32	RW	0400_1000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3Ch	Auto CMD Error Status Register / System Control 2 Register (AUTOCERR_SYSCTL2)	32	RW	0000_0000h
40h	Host controller capabilities register (HOSTCAPBLT)	32	RO	35F2_0000h
44h	Watermark level register (WML)	32	RW	0010_0010h
50h	Force event register (FEVT)	32	WO	0000_0000h
54h	ADMA error status register (ADMAES)	32	RO	0000_0000h
58h	ADMA system address low register (ADSADDRL)	32	RW	0000_0000h
5Ch	ADMA system address high register (ADSADDRH)	32	RW	0000_0000h
FCh	Host controller version register (HOSTVER)	32	RO	0000_2102h
104h	DMA error address low register (DMAERRADDRL)	32	RO	0000_0000h
108h	DMA error address high register (DMAERRADDRH)	32	RO	0000_0000h
10Ch	DMA error attribute register (DMAERRATTR)	32	RO	0000_0000h
114h	Host controller capabilities register 2 (HOSTCAPBLT2)	32	RO	0000_AF07h
120h	Tuning block control register (TBCTL)	32	RW	0000_0002h
124h	Tuning block status register (TBSTAT)	32	RW	0000_0000h
128h	Tuning block pointer register (TBPTR)	32	RW	0000_0000h
140h	SD direction control register (SDDIRCTL)	32	RW	0000_0001h
144h	SD Clock Control Register (SDCLKCTL)	32	RW	0000_0000h
40Ch	eSDHC control register (ESDHCCTL)	32	RW	0000_0000h

14.3.2 SDMA system address register/Block attributes 2 (DSADDR_BLKATTR2)

14.3.2.1 Offset

Register	Offset
DSADDR_BLKATTR2	0h

14.3.2.2 Function

The DSADDR contains the physical system memory address used for single DMA transfers or second argument for Auto CMD23.

14.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.2.4 Fields

Field	Function
31-0 DS_ADDR	DMA system address / Block attributes 2 This register contains the physical system memory address used for DMA transfers or second argument for Auto CMD23. SDMA system address: This register contains the 32-bit system memory address for a single DMA (SDMA) transfer. When the eSDHC stops a DMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). The host driver should initialize this register before starting a every DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. The eSDHC DMA does not support a virtual memory system. It only supports continuous physical memory access. Block attributes 2: This register is used with Auto CMD23 to set 32-bit block count value to the argument of CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by 16-bit Block Count field in Block Attributes register. 65536 blocks is the maximum value available in this case.

14.3.3 Block attributes register (BLKATTR)

14.3.3.1 Offset

Register	Offset
BLKATTR	4h

14.3.3.2 Function

The BLKATTR is used to configure the number of data blocks and the number of bytes in each block.

14.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BLKCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				BLKSIZE											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.3.4 Fields

Field	Function
31-16 BLKCNT	Blocks count for current transfer. • This register is enabled when XFERTYP[BCEN] is set to 1 and is valid only for multiple block transfers. The host driver should set this register to a value between 1 and the maximum block count. The eSDHC decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. • This register should be accessed only when no transaction is executing (that is, after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. • When saving transfer content as a result of a suspend command, the number of blocks yet to be transferred can be determined by reading this register. The reading of this register should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when suspend command is sent out, eSDHC will regard the current transfer is aborted and change BLKCNT back to its original value instead of keeping the dynamical indicator of remained block count. • When restoring transfer content prior to issuing a resume command, the host driver should restore the previously saved block count.

Table continues on the next page...

Field	Function
	0000000000000000b - Stop count 0000000000000001b - 1 block 0000000000000010b - 2 blocks 1111111111111111b - 65535 blocks
15-12 —	Reserved
11-0 BLKSIZE	Transfer block size. This register specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations will be ignored. 000000000000b - No data transfer 000000000001b - 1 byte 000000000010b - 2 bytes 000000000011b - 3 bytes 000000000100b - 4 bytes 000111111111b - 511 bytes 001000000000b - 512 bytes 100000000000b - 2048 bytes

14.3.4 Command argument register (CMDARG)

14.3.4.1 Offset

Register	Offset
CMDARG	8h

14.3.4.2 Function

The CMDARG contains the SD/eMMC command argument.

14.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.4.4 Fields

Field	Function
31-0 CMDARG	Command argument. The SD/eMMC command argument is specified as bits 39-8 of the command format in the SD or eMMC Specification. This register is write protected when PRSSTAT[CIHB] is set.

14.3.5 Transfer type register (XFERTYP)

14.3.5.1 Offset

Register	Offset
XFERTYP	Ch

14.3.5.2 Function

The host driver should set the XFERTYP to issue any new command. To prevent data loss, the eSDHC prevents writing to the bits, that are involved in the data transfer of this register, when data transfer is active: MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver should check PRSSTAT[CDIHB] and PRSSTAT[CIHB] before writing to this register. When PRSSTAT[CDIHB] is set, any attempt to send a command with data by writing to this register is ignored; when PRSSTAT[CIHB] is set, any write to this register is ignored.

On sending commands with data transfer, it is mandatory that the block size is non-zero. Besides, block count must also be non-zero, or indicated as single block transfer (XFERTYP[MSBSEL] is '0' when written), or block count is disabled (XFERTYP[BCEN] is '0' when written), otherwise eSDHC will ignore the sending of this command and do nothing. For write commands, with all above restrictions, it is also mandatory that the write protect switch is not active (PRSSTAT[WPS] is '1), otherwise eSDHC will also ignore the command.

If the commands with write data transfer does not receive the response in 64 clock cycles, that is, response time-out, eSDHC will regard the external device does not accept the command and will not initiate the data transfer on SD bus. In this scenario, the driver should perform error recovery and issue the command again to re-try the transfer.

It is also possible that for some reason the card responds to the read data command but eSDHC does not receive the response, and if it is a DMA (SDMA or ADMA) read operation, the external system memory is over-written by the DMA with data sent back from the card.

Table 14-2. Transfer Type Register Setting for Various Transfer Types

Multi-/Single Block Select	Block Count Enable	Block Count	Function
0	Don't Care	Don't Care	Single Transfer
1	0	Don't Care	Infinite Transfer
1	1	Positive Number	Multiple Transfer
1	1	Zero	No Data Transfer

The table below shows the relationship between the command index check enable and the command CRC check enable, in regards to the response type bits as well as the name of the response type.

Table 14-3. Relationship Between Parameters and the Name of the Response Type

Response type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6, R7
11	1	1	R1b, R5b

- In the SDIO Specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO Specification. But R5b is defined in this specification to specify that the eSDHC checks the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command should be used with R5b.
- The CRC field for R3 and R4 is expected to be all 1 bits. The CRC check should be disabled for these response types.

14.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		CMDINX						CMDTYP		DPSEL	CICEN	CCCEN	Reserved	RSPTYP	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										MSBSEL	DTDSEL	ACEN		BCEN	DMAEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.5.4 Fields

Field	Function
31-30 —	Reserved
29-24 CMDINX	Command index Command index. These bits should be set to the command number that is specified in bits 45-40 of the command format in the SD Memory Card Physical Layer Specification and SDIO Card Specification .
23-22 CMDTYP	Command Type Command Type. There are three types of special commands: suspend, resume, and abort. These bits should be set to 00b for all other commands. • Suspend command: If the suspend command succeeds, the eSDHC should assume that the card bus has been released and that it is possible to issue the next command which uses the DAT line. Since eSDHC does not monitor the content of command response, it does not know if the suspend command succeeded or not. It is the host driver's responsibility to check the status of the suspend command and send another command marked as Suspend to inform the eSDHC that a suspend command was successfully issued. Refer to Suspend resume for more details. After the start bit of command is sent, the eSDHC de-asserts read wait for read transactions and stops checking busy for write transactions. In 4-bit mode, the interrupt cycle starts. If the suspend command fails, the

Table continues on the next page...

Field	Function
	eSDHC will maintain its current state, and the host driver should restart the transfer by setting PROCTL[CREQ]. • Resume command: The host driver re-starts the data transfer by restoring the registers saved before sending the suspend command and then sends the resume command. • Abort command: If this command is set when executing a read transfer, the eSDHC will stop write to the buffer after end bit of abort command is sent. If this command is set when executing a write transfer, the eSDHC will stop driving the DAT line. After issuing the abort command, the host driver should issue a software reset (abort transaction). 00b - Normal other commands 01b - Suspend CMD52 for writing bus suspend in CCCR 10b - Resume CMD52 for writing function select in CCCR 11b - Abort CMD12, CMD52 for writing I/O abort in CCCR
21 DPSEL	Data present select. This bit is set to 1 to indicate that data is present and should be transferred using the DAT line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52) • Commands with no data transfer, but using the busy signal on DAT[0] line (R1b or R5b, for example, CMD38) **NOTE:** In resume command, this bit should be set, and other bits in this register should be set the same as when the transfer was initially launched. When the write protect switch is on (that is, PRSSTAT[WPS] is active as '0'), any command with a write operation will be ignored. That is to say, when this bit is set, while XFRTYP[DTDSEL] is 0, writes to the transfer type register (XFRTYP) are ignored. 0b - No data present 1b - Data present
20 CICEN	Command index check enable. If this bit is set to 1, the eSDHC checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a command index error. If this bit is set to 0, the Index field is not checked. 0b - Disable 1b - Enable
19 CCCEN	Command CRC check enable Command CRC check enable. If this bit is set to 1, the eSDHC should check the CRC field in the response. If an error is detected, it is reported as a command CRC error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. (Refer to RSPTYP[1:0] and Table 14-3.) 0b - Disable 1b - Enable
18 —	Reserved
17-16 RSPTYP	Response type select. 00b - No response 01b - Response length 136 10b - Response length 48 11b - Response length 48, check busy after response
15-6 —	Reserved
5 MSBSEL	Multi-/single-block select Multi-/single-block select. This bit enables multiple block DAT line data transfers. For any other commands, this bit should be set to 0. If this bit is 0, it is not necessary to set the block count register. (Refer to Table 14-2.)

Table continues on the next page...

Field	Function
	0b - Single block 1b - Multiple blocks
4 DTDSEL	Data transfer direction select. This bit defines the direction of DAT line data transfers. The bit is set to 1 by the host driver to transfer data from the SD card to the eSDHC and is set to 0 for all other commands. 0b - Write (host to card) 1b - Read (card to host)
3-2 ACEN	Auto CMD12 enable Auto CMD12 enable. Multiple block transfers for memory require a CMD12 to stop the transaction. When this bit is set to 1, the eSDHC will issue a CMD12 automatically when the last block transfer has completed. The host driver should not set this bit to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File security specification do not require CMD12. In single block transfer, the eSDHC will ignore this bit whether if it is set or not. Auto CMD23 Enable. When this bit is set to 10b, the eSDHC will issue a CMD23 automatically before issuing a command specified in the transfer type register. The following conditions are required to use Auto CMD23: • A memory card that supports CMD23 (For SD card, SCR[33]=1) • If DMA is used, it shall be ADMA • Only when CMD18 or CMD25 is issued **NOTE:** eSDHC does not check command index Auto CMD23 can be issued with or without ADMA, by writing Transfer Type register, eSDHC issues a CMD23 first and then issues a command specified by Transfer Type register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the Auto CMD Error Status register. 32-bit block count value for CMD23 is set to DMA System Address / Argument 2 register. 00b - Auto CMD Disable 01b - Auto CMD12 Enable 10b - Auto CMD23 Enable 11b - Reserved
1 BCEN	Block count enable. This bit is used to enable the block count register, which is only relevant for multiple block transfers. When this bit is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. If ADMA data transfer is more than 65535 blocks, this bit shall be set to 0. In this case, data transfer length is designated by Descriptor Table. 0b - Disable 1b - Enable
0 DMAEN	DMA enable DMA enable. This bit enables DMA functionality. If this bit is set to 1, a DMA operation should begin when the host driver sets the DPSEL bit of this register. Whether the single DMA or ADMA is active depends on PROCTL[DMAS]. 0b - Disable 1b - Enable

14.3.6 Command response 0 register (CMDRSP0)

14.3.6.1 Offset

Register	Offset
CMDRSP0	10h

14.3.6.2 Function

The CMDRSP0 is used to store part 0 of the response bits from the card.

14.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.6.4 Fields

Field	Function
31-0 CMDRSP0	Command response 0 Command response 0. Refer to Command Response 3 register (CMDRSP3) for the mapping of command responses from the SD bus to this register for each response type.

14.3.7 Command response 1 register (CMDRSP1)

14.3.7.1 Offset

Register	Offset
CMDRSP1	14h

14.3.7.2 Function

The CMDRSP1 is used to store part 1 of the response bits from the card.

14.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.7.4 Fields

Field	Function
31-0 CMDRSP1	Command response 1 Command response 1. Refer to Command Response 3 register (CMDRSP3) for the mapping of command responses from the SD bus to this register for each response type.

14.3.8 Command response 2 register (CMDRSP2)

14.3.8.1 Offset

Register	Offset
CMDRSP2	18h

14.3.8.2 Function

The CMDRSP2 is used to store part 2 of the response bits from the card.

14.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.8.4 Fields

Field	Function
31-0 CMDRSP2	Command response 2 Command response 2. Refer to Command Response 3 register (CMDRSP3) for the mapping of command responses from the SD bus to this register for each response type.

14.3.9 Command Response 3 register (CMDRSP3)

14.3.9.1 Offset

Register	Offset
CMDRSP3	1Ch

14.3.9.2 Function

The CMDRSP3 is used to store part 3 of the response bits from the card.

Table below describes the mapping of command responses from the SD bus to command response registers for each response type. In the table, R *n* refers to a bit range within the response data as transmitted on the SD bus.

Table 14-4. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card Status	R[39:8]	CMDRSP0
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[8:31], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (Publish RCA)	New Published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows the following:

Most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, the eSDHC only stores part of the response data in the command response registers (CMDRSP *n*). This enables the host driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the index field and the CRC, are checked by the eSDHC (as specified by the command index check enable and the command CRC check enable

bits in the transfer type register, XFERTYP) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the eSDHC checks R[47:1], and if the response length is 136 the eSDHC checks R[119:1].

Since eSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, it stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows the eSDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When the eSDHC modifies part of the command response registers (CMDRSP *n*), as shown in the table above, it preserves the unmodified bits.

14.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.9.4 Fields

Field	Function
31-0 CMDRSP3	Command response 3 Command response 3. Refer to Command Response 3 register (CMDRSP3) for the mapping of command responses from the SD bus to this register for each response type.

14.3.10 Buffer data port register (DATPORT)

14.3.10.1 Offset

Register	Offset
DATPORT	20h

14.3.10.2 Function

The DATPORT is a 32-bit data port register used to access the internal buffer. Byte access is not allowed.

14.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.10.4 Fields

Field	Function
31-0 DATCONT	Data content. The buffer data port register is for 32-bit data access by the CPU. When the DMA is enabled, any write to this register is ignored, and any read from this register always yields zeros.

14.3.11 Present state register (PRSSTAT)

14.3.11.1 Offset

Register	Offset
PRSSTAT	24h

14.3.11.2 Function

The host driver can get the status of the eSDHC from the PRSSTAT, which is a 32-bit, read-only register.

14.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL								CLSL	Reserved			WPS	CDS	Reserved	CINS
W																
Reset	1	1	1	1	1	1	1	1	1	0	0	0	u	u	0	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				BREN	BWEN	RTA	WTA	SDOFF	Reserved			SDSTB	DLA	CDIHB	CIHB
W																
Reset	0	0	0	0	0	0	0	0	u	0	0	0	1	0	0	0

14.3.11.4 Fields

Field	Function
31-24	DAT[7:0] line signal level
DLSL	DAT[7:0] line signal level. This status is used to check the DAT line level to recover from errors, and for debugging.This is especially useful in detecting the busy signal level from DAT[0]. The reset value is effected by the external pull-up/pull-down resistors. By default, the read value of this bit field after reset is 8'b11111111 DLSL[0]: Data 0 line signal level DLSL[1]: Data 1 line signal level DLSL[2]: Data 2 line signal level DLSL[3]: Data 3 line signal level DLSL[4]: Data 4 line signal level

Table continues on the next page...

Field	Function
	DLSL[5]: Data 5 line signal level DLSL[6]: Data 6 line signal level DLSL[7]: Data 7 line signal level
23 CLSL	CMD line signal level. This status is used to check the CMD line level to recover from errors, and for debugging. The reset value is effected by the external pull-up/pull-down resistor, by default, the read value of this bit after reset is 1, when the command line is pulled up.
22-20 —	Reserved
19 WPS	Write protect state Write protect state. The write protect switch is supported for memory and combo cards. This bit reflects the write protect state depending on value of SDHC_WP pin of the card socket. A software reset does not affect this bit. The reset value is effected by the external write protect switch. If the SDHC_WP pin is not used, it should be tied low, so that the reset value of this bit is high and write is enabled. 0b - Write protected (SDHC_WP =1) 1b - Write enabled (SDHC_WP =0)
18 CDS	Card detect state Card detect state. This bit reflects the card inserted/removed state depending on value of the SDHC_CD_B pin for the card socket. Debouncing is not performed on this bit. This bit may be valid, but is not guaranteed, because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. A software reset does not effect this bit. A write to the force event register (FEVT) does not affect this bit. The reset value is effected by the external card detection pin. This bit shows the value on the SDHC_CD_B pin (that is, when a card is inserted in the socket, it is 0 on the SDHC_CD_B input, and consequently the CDS reads 1.) 0b - No card present (SDHC_CD_B =1) 1b - Card present SDHC_CD_B =0)
17 —	Reserved
16 CINS	Card inserted. This bit indicates whether a card has been inserted. The eSDHC debounces this signal so that the host driver will not need to wait for it to stabilize. Changing from a 0 to 1 generates a card insertion interrupt in the interrupt status register (IRQSTAT). Changing from a 1 to 0 generates a card removal interrupt in the interrupt status register (IRQSTAT). A write to the force event register (FEVT) does not effect this bit. The software reset for all in the system control register (SYSCTL) does not effect this bit. A software reset does not effect this bit. 0b - Power on reset or no card 1b - Card inserted
15-12 —	Reserved
11 BREN	Buffer read enable. This status bit is used for non-DMA read transfers. The eSDHC may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this bit is high, valid data greater than the watermark level exist in the buffer. A change of this bit from 1 to 0 occurs when any read from the buffer is made. A change of this bit from 0 to1 occurs when there is enough valid data ready in the buffer and the buffer read ready interrupt has been generated and enabled. 0b - Read disable 1b - Read enable

Table continues on the next page...

Field	Function
10 BWEN	Buffer write enable. This status bit is used for non-DMA write transfers. The eSDHC can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, valid data greater than the watermark level can be written to the buffer. A change of this bit from 1 to 0 occurs when any write to the buffer is made. A change of this bit from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the buffer write ready interrupt is generated and enabled. 0b - Write disable 1b - Write enable

9
RTA
	Read transfer active. This status bit is used for detecting completion of a read transfer.
	• This bit is set for either of the following conditions:
	  • After the end bit of the read command.
	  • When read operation is restarted by writing a 1 to PROCTL[CREQ].
	• This bit is cleared for either of the following conditions:
	  • When the last data block as specified by block length is transferred to the system.
	  • In the case of ADMA2, end of read operation is designated by descriptor table.
	  • When all valid data blocks in the host controller have been transferred to the system and no current block transfers are being sent as a result of the stop at block gap request being set to 1.
	A transfer complete interrupt is generated when this bit changes to 0.
	0b - No valid data
	1b - Transferring data

8
WTA
	Write transfer active. This status bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the eSDHC.
	• This bit is set in either of the following cases:
	  • After the end bit of the write command.
	  • When write operation is restarted by writing a 1 to PROCTL[CREQ].
	• This bit is cleared in either of the following cases:
	  • After getting the CRC status of the last data block as specified by the transfer count (single and multiple). In case of ADMA2, transfer count is designated by descriptor table.
	  • After getting the CRC status of any block where data transmission is about to be stopped by a stop at block gap request.
	During a write transaction, a block gap event interrupt is generated when this bit is changed to 0, as result of the stop at block gap request being set. This status is useful for the host driver in determining when to issue commands during write busy state.
	0b - No valid data
	1b - Transferring data

7 SDOFF	SD clock gated off internally. This status bit indicates that the SD clock is internally gated off, because of buffer over/under-run or read pause without read wait assertion. 0b - SD clock is active 1b - SD clock is gated off
6-4 —	Reserved
3 SDSTB	SD clock stable. This status bit indicates that the internal card clock is stable. This bit is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear SYSCTL[SDCLKEN] to remove glitch on the card clock when the frequency is changing. 0b - Clock is changing frequency and not stable 1b - Clock is stable
2	Data line active. This status bit indicates whether one of the DAT lines on the SD bus is in use.

Table continues on the next page...

Field	Function

DLA	In the case of read transactions:
	• This status indicates if a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a block gap event interrupt in the interrupt status register (IRQSTAT).
	• This bit will be set in either of the following cases:
	  • After the end bit of the read command.
	  • When writing a 1 to PROCTL[CREQ] to restart a read transfer.
	• This bit should be cleared in either of the following cases:
	  • When the end bit of the last data block is sent from the SD bus to the host controller. In case of ADMA2, the last block is designated by the last transfer of descriptor table.
	  • When a read transfer is stopped at the block gap initiated by a stop at block gap request.
	• The eSDHC will wait at the next block gap by driving read wait at the start of the interrupt cycle. If the read wait signal is already driven (data buffer cannot receive data), the eSDHC can wait for a current block gap by continuing to drive the read wait signal. It is necessary to support read wait in order to use the suspend/resume function. This bit will remain 1 during read wait.
	
	In the case of write transactions:
	• This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a transfer complete interrupt in the interrupt status register (IRQSTAT).
	• This bit will be set in either of the following cases:
	  • After the end bit of the write command.
	  • When writing to 1 to PROCTL[CREQ] to continue a write transfer.
	• This bit will be cleared in either of the following cases:
	  • When the SD card releases write busy of the last data block. If SD card does not drive busy signal for 8 SD Clocks, the host controller should consider the card drive "Not Busy". In case of ADMA2, the last block is designated by the last transfer of descriptor table.
	  • When the SD card releases write busy, prior to waiting for write transfer, and as a result of a stop at block gap request.
	
	In the case of command with busy pending:
	• Command with busy. This status indicates whether a command indicates busy (for example, erase command for memory) is executing on the SD bus. This bit is set after the end bit of the command with busy and cleared when busy is de-asserted.
	
	Changing this bit from 1 to 0 generate a transfer complete interrupt in the interrupt status register (IRQSTAT).
	0b - DAT line inactive
	1b - DAT line active

1 CDIHB	Command inhibit (DAT). This status bit is generated if either the DAT line active or the read transfer active is set to 1. If this bit is 0, it indicates the eSDHC can issue the next SD/eMMC Command. Commands with busy signal belong to command inhibit (DAT) (for example, R1b, R5b type). Changing from 1 to 0 generates a transfer complete interrupt in the interrupt status register (IRQSTAT). **NOTE:** The SD host driver can save registers for a suspend transaction after this bit has changed from 1 to 0. 0b - Can issue command which uses the DAT line 1b - Cannot issue command which uses the DAT line
0 CIHB	Command inhibit (CMD). If this status bit is 0, it indicates that the CMD line is not in use and the eSDHC can issue a SD/eMMC Command using the CMD line. This bit is set also immediately after the transfer type register (XFERTYP) is written. This bit is cleared when the command response is received. Even if the command inhibit (DAT) is set to 1, Commands using only the CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a command complete interrupt in the interrupt status register (IRQSTAT). If the eSDHC cannot issue the command

Field	Function
	because of a command conflict error (Refer to command CRC error) or because of a command not issued by Auto CMD12 Error, this bit will remain 1 and the command complete is not set. The status of issuing an Auto CMD12 does not show on this bit. 0b - Can issue command using only CMD line 1b - Cannot issue command

14.3.12 Protocol control register (PROCTL)

14.3.12.1 Offset

Register	Offset
PROCTL	28h

14.3.12.2 Function

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether the eSDHC issues a suspend command or the SD card accepts the suspend command.

1. If the host driver does not issue a suspend command, the continue request should be used to restart the transfer.
2. If the host driver issues a suspend command and the SD card accepts it, a resume command should be used to restart the transfer.
3. If the host driver issues a suspend command and the SD card does not accept it, the continue request should be used to restart the transfer.

Any time a stop at block gap request stops the data transfer, the host driver should wait for a transfer complete (in the interrupt status register, IRQSTAT), before attempting to restart the transfer. When restarting the data transfer by continue request, the host driver should clear the stop at block gap request before or simultaneously.

14.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					WECRM	WECINS	WECINT	Reserved				IABG	RWCTL	CREQ	SABGREQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					VOLT_SEL	DMAS		CDSS	CDTL	EMODE		Reserved	DTW		Reserved
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

14.3.12.4 Fields

Field	Function
31-27 —	Reserved
26 WECRM	Wakeup event enable on SD card removal Wakeup event enable on SD card removal. This bit enables a wakeup event, via a card removal, in the interrupt status register (IRQSTAT). FN_WUS (wake up support) in CIS of SDIO card does not effect this bit. When this bit is set, the card removal status and the eSDHC interrupt can be asserted without SDHC_CLK toggling. When the wakeup feature is not enabled, the SDHC_CLK must be active in order to assert the card removal status and the eSDHC interrupt. 0b - Disable 1b - Enable
25 WECINS	Wakeup event enable on SD card insertion Wakeup event enable on SD card insertion. This bit enables a wakeup event, via a card insertion, in the interrupt status register (IRQSTAT). FN_WUS (wake up support) in CIS of the SDIO card does not affect this bit. When this bit is set, the card insertion status and the eSDHC interrupt can be asserted without SDHC_CLK toggling. When the wakeup feature is not enabled, the SDHC_CLK must be active in order to assert the card insertion status and the eSDHC interrupt. 0b - Disable 1b - Enable
24 WECINT	Wakeup event enable on card interrupt Wakeup event enable on card interrupt. This bit enables a wakeup event, via a card interrupt, in the interrupt status register (IRQSTAT). This bit can be set to 1 if FN_WUS (wake up support) in CIS of SDIO card is set to 1. When this bit is set, the card interrupt status and the eSDHC interrupt can be asserted without SDHC_CLK toggling. When the wakeup feature is not enabled, the SDHC_CLK must be active in order to assert the card interrupt status and the eSDHC interrupt. 0b - Disable 1b - Enable

Table continues on the next page...

Field	Function
23-20 —	Reserved
19 IABG	Interrupt at block gap. This bit is valid only in 4-bit mode of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO card insertion, it should set this bit according to the CCCR of the card. 0b - Disable 1b - Enable
18 RWCTL	Read wait control. The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise the eSDHC has to stop the SD Clock to hold read data, which restricts commands generation. When the host driver detects an SDIO card insertion, it should set this bit according to the CCCR of the card. If the card does not support read wait, this bit should never be set to 1, otherwise DAT line conflicts may occur. If this bit is set to 0, stop at block gap during read operation is also supported, but the eSDHC will stop the SD clock to pause reading operation. 0b - Disable read wait control, and stop SD clock at block gap when SABGREQ bit is set 1b - Enable read wait control, and assert read wait without stopping SD clock at block gap when SABGREQ bit is set
17 CREQ	Continue request. This bit is used to restart a transaction, which was stopped using the stop at block gap request. To cancel stop at the block gap, set stop at block gap request to 0 and set this bit 1 to restart the transfer. The eSDHC automatically clears this bit in either of the following cases: • In the case of a read transaction, the DAT line active changes from 0 to 1 as a read transaction restarts. • In the case of a write transaction, the write transfer active changes from 0 to 1 as the write transaction restarts. Therefore, it is not necessary for host driver to set this bit to 0. If stop at block gap request is set to 1, any write to this bit is ignored. 0b - No effect 1b - Restart
16 SABGREQ	Stop at block gap request Stop at block gap request. This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the host driver should leave this bit set to 1. Clearing both the stop at block gap request and continue request does not cause the transaction to restart. Read wait is used to stop the read transaction at the block gap. The eSDHC will honor the stop at block gap request for write transfers, but for read transfers it requires that the SDIO card support read wait. Therefore, the host driver should not set this bit during read transfers unless the SDIO card supports read wait and has set the read wait control to 1, otherwise the eSDHC will stop the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the data port register, the host driver should set this bit after all block data is written. If this bit is set to 1, the host driver should not write data to the data port register after a block is sent. Once this bit is set, the host driver should not clear this bit before the transfer complete bit in (IRQSTAT)is set, otherwise the eSDHCs behavior is undefined. This bit effects read transfer active, write transfer active, PRSSTAT[DLA] and PRSSTAT[CDIHB]. 0b - Transfer 1b - Stop
15-11 —	Reserved

Table continues on the next page...

Field	Function
10 VOLT_SEL	Voltage selection Voltage selection. Change the value of output signal SDHC_VS , to control the SD bus supply voltage for external card. There must be a control circuit out of eSDHC to change the voltage. 0b - Change the SD Bus Supply voltage to high voltage range, around 3.0V 1b - Change the SD bus supply voltage to low voltage range, around 1.8V
9-8 DMAS	DMA select DMA select. This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00b - Single DMA is selected 01b - ADMA1 is selected 10b - 11b - 64-bit address ADMA2 is selected
7 CDSS	Card detect signal selection. This bit selects the source for the card detection. 0b - SD CD pin is selected (for normal purposes) 1b - Card detection test level is selected (for test purposes)
6 CDTL	Card detect test level. This is bit is enabled while the card detection signal selection is set to 1 and it indicates card insertion. 0b - Card detect test level is 0, no card inserted 1b - Card detect test level is 1, card inserted
5-4 EMODE	Endian mode. The eSDHC supports little and big endian mode for transferring between buffer port register and data buffer. 00b - Big endian mode 01b - Reserved 10b - Little endian mode 11b - Reserved
3 —	Reserved
2-1 DTW	Data transfer width. This bit selects the data width of the SD bus for a data transfer. The host driver should set it to match the data width of the card. Possible Data transfer widths are 1-bit, 4-bits, and 8-bits. 00b - 1-bit mode 01b - 4-bit mode 10b - 8-bit mode 11b - Reserved
0 —	Reserved

14.3.13 System Control Register when ESDHCCTL[CRS=0] (SYSCTL_ESDHCCTL_CRS_0)

14.3.13.1 Offset

Register	Offset
SYSCTL_ESDHCCTL_CRS_0	2Ch

14.3.13.2 Function

The clock divider mode(16-27 bits in this register) is selected according to Clock Register Select field in eSDHC Control register. Other bits of the register remain unaffected by clock register select value.

14.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved				INITA	RSTD	RSTC	RSTA	Reserved				DTOCV			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	SDCLKFS								DVS				SDCLKEN	Reserved		
Reset	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

14.3.13.4 Fields

Field	Function
31-28 —	Reserved
27 INITA	Initialization active. When this bit is set, 80 SD clocks are sent to the card. After the 80 clocks are sent, this bit is self cleared. This bit is very useful during the card power-up period when 74 SD-Clocks are needed. Writing 1 to this bit when this bit is already 1 has no effect. Writing 0 to this bit at any time has no effect. When either PRSSTAT[CIHB] or PRSSTAT[CDIHB] are set, writing 1 to this bit is ignored (that is, when command line or data lines are active, write to this bit is not allowed). On the other hand, when this bit is set, that is, during intialization active period, it is allowed to issue command, and the command bit

Table continues on the next page...

Field	Function
	stream will appear on the CMD pad after all 80 clock cycles are done. So when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs send 80 cycles to the card and does not want to wait till this bit is self cleared.

26
RSTD
	Software reset for DAT line. Only part of the data circuit is reset. DMA circuit is also reset.
	The following registers and bits are cleared by this bit:
	• Data port register
	  • Buffer is cleared and initialized
	• Present state register (PRSSTAT)
	  • Buffer read enable
	  • Buffer write enable
	  • Read transfer active
	  • Write transfer active
	  • DAT line active
	  • Command inhibit (DAT)
	• Protocol control register (PROCTL)
	  • Continue request
	  • Stop at block gap request
	• Interrupt status register (IRQSTAT)
	  • Buffer read ready
	  • Buffer write ready
	  • DMA interrupt
	  • Block gap event
	  • Transfer complete
	NOTE: This bit will be self cleared by eSDHC when Software Reset for Data is complete, so Software should poll for it to be cleared after setting it.
	0b - No reset
	1b - Reset

25 RSTC	Software reset for CMD line. Only part of the command circuit is reset. The following registers and bits are cleared by this bit: • PRSSTAT[CIHB] • IRQSTAT[CC] **NOTE:** This bit will be self cleared by eSDHC when Software Reset for Command is complete, so software should poll for it to be cleared after setting it. 0b - No reset 1b - Reset
24 RSTA	Software reset for all. This reset effects the entire host controller except for the card detection circuit. Register bits of type R, RW, RW1C are cleared. During its initialization, the host driver should set this bit to 1 to reset the eSDHC. The eSDHC should reset this bit to 0 when the capabilities registers are valid and the host driver can read them. Additional use of software reset for all does not affect the value of the capabilities registers. After this bit is set, it is recommended that the host driver reset the external card and re-initialize it. **NOTE:** The Software Reset For All in the System Control register clears Card Insertion and Card Removal bits in Interrupt Status Register. Software should issue partial reset(Reset for Command and Reset for Data) instead of Reset for All, if it wants to reset eSDHC without clearing these Interrupt Status Register bits. This bit will be self cleared by eSDHC when Software Reset for All is complete, so Software should poll for it to be cleared after setting it. 0b - No reset 1b - Reset
23-20	Reserved

Table continues on the next page...

Field	Function
—	
19-16 DTOCV	Data timeout counter value Data timeout counter value. This value determines the interval by which DAT line timeouts are detected. Refer to the data timeout error bit in the Interrupt status enable register (IRQSTATEN) for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the SDCLK value by this value. The host driver can clear IRQSTATEN[DTOESEN] to prevent inadvertent time-out events. 0000b - SDCLK x 2^{13} 0001b - SDCLK x 2^{14} 1110b - SDCLK x 2^{27} 1111b - Reserved
15-8 SDCLKFS	SDCLK frequency select. This register is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this register holds the prescaler (this register) and divisor (next register) of the base clock frequency register. Base clock can be selected by programming ESDHCCTL[PCS]. It selects between platform clock and peripheral clock / 2 . Setting 0x00 bypasses the frequency prescaler of the SD clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of the base clock and the following divisor bits. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, if the base clock frequency is 96 MHz and the target frequency is 25 MHz, then choosing the prescaler value of 0x01 and divisor value of 0x1 will yield 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 KHz, the prescaler value of 0x08 and divisor value of 0xE yields the exact clock value of 400 KHz. The reset value of this bit field is 0x80, so if the input base clock is about 96 MHz, the default SD clock after reset is 375 KHz. The programmed SD Clock frequency shall never exceed maximum SD clock supported by the card. **NOTE:** Both DVS and SDCLKFS fields should not be programmed 0 simultaneously. Only the following settings are allowed: 00000000b - Base clock 00000001b - Base clock divided by 2 00000010b - Base clock divided by 4 00000100b - Base clock divided by 8 00001000b - Base clock divided by 16 00010000b - Base clock divided by 32 00100000b - Base clock divided by 64 01000000b - Base clock divided by 128 10000000b - Base clock divided by 256
7-4 DVS	Divisor. This register is used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisor without deterioration of duty cycle. The settings are as following: 0000b - Divisor by 1 0001b - Divisor by 2 ... 1110b - Divisor by 15 1111b - Divisor by 16

Table continues on the next page...

Field	Function
3 SDCLKEN	SD clock enable. the host controller should stop SDCLK when writing this bit to 0. SDCLK frequency can be changed when this bit is 0. Then, the host controller should maintain the same clock frequency until SDCLK is stopped (stop at SDCLK=0). If PRSSTAT[CINS] is cleared, this bit should be cleared by the host driver to save power.
2-0 —	Reserved

14.3.14 System Control Register when ESDHCCTL[CRS=1] (SYSCTL_ESDHCCTL_CRS_1)

14.3.14.1 Offset

Register	Offset
SYSCTL_ESDHCCTL_CRS_1	2Ch

14.3.14.2 Function

System Control Register when ESDHCCTL[CRS=1]

14.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				INITA	RST D	RST C	RST A	Reserved				DTOCV			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDCLKFS								USDCLKFS		CGS	Reserved	SDCLKEN	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.14.4 Fields

Field	Function
31-28 —	Reserved
27 INITA	Initialization active. When this bit is set, 80 SD clocks are sent to the card. After the 80 clocks are sent, this bit is self cleared. This bit is very useful during the card power-up period when 74 SD-Clocks are needed. Writing 1 to this bit when this bit is already 1 has no effect. Writing 0 to this bit at any time has no effect. When either PRSSTAT[CIHB] or PRSSTAT[CDIHB] are set, writing 1 to this bit is ignored (that is, when command line or data lines are active, write to this bit is not allowed). On the other hand, when this bit is set, that is, during intialization active period, it is allowed to issue command, and the command bit stream will appear on the CMD pad after all 80 clock cycles are done. So when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs send 80 cycles to the card and does not want to wait till this bit is self cleared.

26
RSTD
	Software reset for DAT line. Only part of the data circuit is reset. DMA circuit is also reset.
	The following registers and bits are cleared by this bit:
	• Data port register
	  • Buffer is cleared and initialized
	• Present state register (PRSSTAT)
	  • Buffer read enable
	  • Buffer write enable
	  • Read transfer active
	  • Write transfer active
	  • DAT line active
	  • Command inhibit (DAT)
	• Protocol control register (PROCTL)
	  • Continue request
	  • Stop at block gap request
	• Interrupt status register (IRQSTAT)
	  • Buffer read ready
	  • Buffer write ready
	  • DMA interrupt
	  • Block gap event
	  • Transfer complete
	NOTE: This bit will be self cleared by eSDHC when Software Reset for Data is complete, so Software should poll for it to be cleared after setting it.
	0b - No reset
	1b - Reset

25 RSTC	Software reset for CMD line. Only part of the command circuit is reset. The following registers and bits are cleared by this bit: • PRSSTAT[CIHB] • IRQSTAT[CC] **NOTE:** This bit will be self cleared by eSDHC when Software Reset for Command is complete, so software should poll for it to be cleared after setting it. 0b - No reset 1b - Reset
24 RSTA	Software reset for all. This reset effects the entire host controller except for the card detection circuit. Register bits of type R, RW, RW1C are cleared. During its initialization, the host driver should set this bit to 1 to reset the eSDHC. The eSDHC should reset this bit to 0 when the capabilities registers are valid

Table continues on the next page...

Field	Function
	and the host driver can read them. Additional use of software reset for all does not affect the value of the capabilities registers. After this bit is set, it is recommended that the host driver reset the external card and re-initialize it. **NOTE:** The software reset for all in the system control register clears card insertion and card removal bits in interrupt status register. Software should issue partial reset (reset for command and reset for data) instead of reset for all, if it wants to reset eSDHC without clearing these interrupt status register bits. This bit will be self cleared by eSDHC when software reset for all is complete, so software should poll for it to be cleared after setting it. 0b - No reset 1b - Reset
23-20 —	Reserved
19-16 DTOCV	Data timeout counter value Data timeout counter value. This value determines the interval by which DAT line timeouts are detected. Refer to the data timeout error bit in the Interrupt status enable register (IRQSTATEN) for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the SDCLK value by this value. The host driver can clear IRQSTATEN[DTOESEN] to prevent inadvertent time-out events. 0000b - SDCLK x 2^{13} 0001b - SDCLK x 2^{14} 1110b - SDCLK x 2^{27} 1111b - Reserved
15-8 SDCLKFS	SDCLK Frequency Select: This register is used to select the frequency of the SDCLK pin. 10-bit divisor value is formed by concatenating USDCLKFS(2-bit) and SDCLKFS(8-bit), that is, Divisor = {USDCLKFS[0:1], SDCLKFS[0:7]} Following Divisor definition is selected depending on Clock Generation Select value: **10-bit Divided Clock Mode** 0x3FF Base clock divided by 2048 0x04 Base clock divided by 8 0x03 Base clock divided by 6 0x02 Base clock divided by 4 0x01 Base clock divided by 2 0x00 Reserved **10-bit Programmable Clock Mode** 0x3FF Base clock divided by 1024 0x04 Base clock divided by 5 0x03 Base clock divided by 4 0x02 Base clock divided by 3 0x01 Base clock divided by 2 0x00 Reserved The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (divisor)

Table continues on the next page...

Field	Function
7-6 USDCLKFS	Upper bits of SDCLK frequency select. This field is used to expand SDCLKFS to 10 bits. These two bits occupies most significant portion of 10-bit SDCLKFS.
5 CGS	Clock Generator Select. This field selects 10-bit SDCLKFS clock mode. 0b - Divided clock mode is selected 1b - Programmable clock mode is selected
4 —	Reserved
3 SDCLKEN	SD clock enable. The host controller shall stop SDCLK when writing this bit to 0. SDCLK frequency can be changed when this bit is 0. Then, the host controller shall maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the card inserted in the present state register is cleared, this bit should be cleared by the host driver to save power.
2-0 —	Reserved

14.3.15 Interrupt status register (IRQSTAT)

14.3.15.1 Offset

Register	Offset
IRQSTAT	30h

14.3.15.2 Function

An interrupt is generated when the normal interrupt signal enable is enabled and at least one of the status bits is set to 1. For most bits, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For card interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the card driver services the interrupt condition, otherwise the CINT bit will be asserted again

Table 14-5. eSDHC Status for Command Timeout Error/Command Complete Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

Table 14-6. eSDHC Status for Data Timeout Error/Transfer Complete Bit Combinations

Transfer Complete	Data Timeout Error	Meaning of the Status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data Transfer Complete

Table 14-7. eSDHC Status for Command CRC Error/Command Timeout Error Bit Combinations

Command CRC Error	Command Timeout Error	Meaning of the Status
0	0	No error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

14.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		RTOE	DMAE	Reserved	TNE	ADMAE	AC12E	Reserved	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W			W1C	W1C		W1C	W1C	W1C		W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			RTE	Reserved			CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W				W1C					W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.15.4 Fields

Field	Function
31-30 —	Reserved
29	Register access timeout error. This bit indicate that register access has timed-out. 0b - No timeout error

Table continues on the next page...

Field	Function
RTOE	1b - Timeout error
28 DMAE	DMA error DMA error. This bit indicates that DMA (SDMA or ADMA) transfer has failed. 0b - No error 1b - Error
27 —	Reserved
26 TNE	Tuning Error. This bit is set when an unrecoverable error is detected in a tuning circuit except during tuning procedure (Occurrence of an error during tuning procedure is indicated by Sampling Clock Select). By detecting Tuning Error, Host Driver needs to abort a command executing and perform tuning. The Tuning Error is higher priority than the other error interrupts generated during data transfer. By detecting Tuning Error, the Host Driver should discard data transferred by a current read/write command and retry data transfer after the Host Controller retrieved from tuning circuit error. This bit might not be set in some cases, but SD command or Data error might be set; Driver should consider it as tuning circuit error and perform tuning procedure. 0b - No error 1b - Error
25 ADMAE	ADMA error. This bit is set when the host controller detects errors during ADMA operation. The state of the ADMA at an error occurrence is saved in the ADMA error status register. 0b - No error 1b - Error
24 AC12E	Auto CMD12 error. Occurs when detecting that one of the bits in the Auto CMD12 error status register (AUTOC12ERR) has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur, but also when the Auto CMD12 is not executed due to the previous command error. 0b - No error 1b - Error
23 —	Reserved
22 DEBE	Data end bit error. Occurs either when detecting 0 at the end bit position of read data, which uses the DAT line, or at the end bit position of the CRC. 0b - No error 1b - Error
21 DCE	Data CRC error. Occurs when detecting a CRC error when transferring read data, which uses the DAT line, or when detecting the write CRC status having a value other than 010. 0b - No error 1b - Error
20 DTOE	Data timeout error. Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after write CRC status • Write CRC status time-out • Read data time-out 0b - No error 1b - Time out
19 CIE	Command index error. Occurs if a command index error occurs in the command response. 0b - No error 1b - Error
18	Command end bit error. Occurs when detecting that the end bit of a command response is 0.

Table continues on the next page...

Field	Function
CEBE	0b - No error 1b - End bit error generated
17 CCE	Command CRC error. A command crc error is generated in two cases: • If a response is returned and the command timeout error is set to 0 (indicating no time-out), this bit is set when detecting a CRC error in the command response. • The eSDHC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the eSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then the eSDHC should abort the command (stop driving CMD line) and set this bit to 1. The command timeout error should also be set to 1 to distinguish CMD line conflict. 0b - No error 1b - CRC error generated
16 CTOE	Command timeout error Command timeout error. Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the eSDHC detects a CMD line conflict, in which case a command CRC error should also be set (as shown in), this bit should be set without waiting for 64 SDCLK cycles. This is because the command will be aborted by the eSDHC. 0b - No error 1b - Time out
15-13 —	Reserved
12 RTE	Re-tuning event. This status is set if re-tuning request in the eSDHC control register changes from 0 to 1. eSDHC requests host driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 0b - Re-tuning is not required 1b - Re-tuning should be performed
11-9 —	Reserved
8 CINT	Card interrupt. Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the host controller should detect the card interrupt without SD clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the host system. It is necessary to define how to handle this delay. When this status has been set and the host driver needs to start this interrupt service, IRQSTATEN[CINTSEN] should be set to 0 in order to clear the card interrupt statuses latched in the eSDHC and to stop driving the interrupt signal to the host system. After completion of the card interrupt service (It should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set the card interrupt status enable bit to 1 and start sampling the interrupt signal again. 0b - No card interrupt 1b - Generate card interrupt
7 CRM	Card removal. This status is set if the PRSSTAT[CINS] changes from 1 to 0. When the host driver writes this bit to 1 to clear this status, the status of the PRSSTAT[CINS] should be confirmed. Because the card detect state may possibly be changed when the host driver clear this bit and interrupt event may not be generated. **NOTE:** The Software Reset For All in the System Control register clears this bit. Software should issue partial reset(Reset for Command and Reset for Data) instead of Reset for All, if it wants to reset eSDHC without clearing this bit. eSDHC does not implement de-bouncing circuit on card detect pin, so Software should check Card Inserted in Present State register to confirm card insertion/removal status. 0b - Card state unstable or inserted

Table continues on the next page...

Field	Function
	1b - Card removed
6 CINS	Card insertion. This status is set if PRSSTAT[CINS] changes from 0 to 1. When the host driver writes this bit to 1 to clear this status, the status of the PRSSTAT[CINS] should be confirmed. Because the card detect state may possibly be changed when the host driver clear this bit and interrupt event may not be generated. **NOTE:** The Software Reset For All in the System Control register clears this bit. Software should issue partial reset(Reset for Command and Reset for Data) instead of Reset for All, if it wants to reset eSDHC without clearing this bit. eSDHC does not implement de-bouncing circuit on card detect pin, so Software should check Card Inserted in Present State register to confirm card insertion/removal status. 0b - Card state unstable or removed 1b - Card inserted
5 BRR	Buffer read ready Buffer read ready. This status bit is set if PRSSTAT[BREN] changes from 0 to 1. Refer to the description of the buffer read enable bit in Present state register (PRSSTAT) for additional information. 0b - Not ready to read buffer 1b - Ready to read buffer
4 BWR	Buffer write ready Buffer write ready. This status bit is set if the PRSSTAT[BWEN] changes from 0 to 1. Refer to the description of the buffer write enable bit in Present state register (PRSSTAT) for additional information. 0b - Not ready to write buffer 1b - Ready to write buffer
3 DINT	DMA interrupt DMA interrupt. Occurs only when the DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this bit will not be set. Instead, the DMAE bit will be set. Either single DMA or ADMA finishes data transferring, this bit will be set. 0b - No DMA Interrupt 1b - DMA Interrupt is generated
2 BGE	Block gap event. If PROCTL[SABGREQ] is set, this bit is set when a read or write transaction is stopped at a block gap. If PROCTL[SABGREQ] is not set to 1, this bit is not set to 1. • In the case of a read transaction, this bit is set at the falling edge of the DAT Line active status (when the transaction is stopped at SD bus timing). The read wait must be supported in order to use this function. • In the case of a write transaction, this bit is set at the falling edge of write transfer active status (after getting CRC status at SD Bus timing). 0b - No block gap event 1b - Transaction stopped at block gap
1 TC	Transfer complete. This bit is set when a read/write transfer and a command with busy is completed. While performing tuning procedure (Execute Tuning is set to 1), Transfer Complete is not set for CMD19 execution. 0b - Transfer not complete 1b - Transfer complete
0 CC	Command complete. This bit is set when the end bit of the command response is received (except Auto CMD12). Refer to PRSSTAT[CIHB]. 0b - Command not complete 1b - Command complete

14.3.16 Interrupt status enable register (IRQSTATEN)

14.3.16.1 Offset

Register	Offset
IRQSTATEN	34h

14.3.16.2 Function

Setting the bits to 1 in the IRQSTATEN enables the corresponding interrupt status to be set by the specified event. If any bit is cleared, the corresponding interrupt status bit is also cleared (that is, when the bit in this register is cleared, the corresponding bit in interrupt status register, IRQSTAT, is always 0).

NOTE

- Depending on how PROCTL[IABG] is set, eSDHC may be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There will be some delays on the card interrupt, asserted from the card, to the time the host system is informed.
- To detect a CMD line conflict, the host driver must set both IRSTAT[CTOESEN] and IRSTAT[CCESEN] to 1.

14.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R / W	Reserved		RTOESEN	DMAESEN	Reserved	TNESEN	ADMAESEN	AC12ESEN	Reserved	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN
Reset	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	Reserved			RTESEN	Reserved			CINTSEN	CRMSEN	CINSEN	BRRSEN	BWRSEN	DINTSEN	BGESEN	TCSEN	CCSEN
Reset	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1

14.3.16.4 Fields

Field	Function
31-30 —	Reserved
29 RTOESEN	Register access timeout status enable 0b - Masked 1b - Enabled
28 DMAESEN	DMA error status enable. 0b - Masked 1b - Enabled
27 —	Reserved
26 TNESEN	Tuning error status enable 0b - Masked 1b - Enabled
25 ADMAESEN	ADMA error status enable. 0b - Masked 1b - Enabled
24 AC12ESEN	Auto CMD12 error status enable. 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBESEN	Data end bit error status enable. 0b - Masked 1b - Enabled
21 DCESEN	Data CRC error status enable. 0b - Masked 1b - Enabled
20 DTOESEN	Data timeout error status enable. 0b - Masked 1b - Enabled
19 CIESEN	Command index error status enable. 0b - Masked 1b - Enabled
18 CEBESEN	Command end bit error status enable. 0b - Masked 1b - Enabled
17 CCESEN	Command CRC error status enable. 0b - Masked 1b - Enabled

Table continues on the next page...

Field	Function
16 CTOESEN	Command timeout error status enable. 0b - Masked 1b - Enabled
15-13 —	Reserved
12 RTESEN	Re-tuning event status enable. 0b - Masked 1b - Enabled
11-9 —	Reserved
8 CINTSEN	Card interrupt status enable. If this bit is set to 0, the eSDHC will clear the interrupt request to the system. The card interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The host driver should clear the card interrupt status enable before servicing the card interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0b - Masked 1b - Enabled
7 CRMSEN	Card removal status enable. 0b - Masked 1b - Enabled
6 CINSEN	Card insertion status enable. 0b - Masked 1b - Enabled
5 BRRSEN	Buffer read ready status enable. 0b - Masked 1b - Enabled
4 BWRSEN	Buffer write ready status enable. 0b - Masked 1b - Enabled
3 DINTSEN	DMA interrupt status enable. 0b - Masked 1b - Enabled
2 BGESEN	Block gap event status enable. 0b - Masked 1b - Enabled
1 TCSEN	Transfer complete status enable. 0b - Masked 1b - Enabled
0 CCSEN	Command complete status enable. 0b - Masked 1b - Enabled

14.3.17 Interrupt signal enable register (IRQSIGEN)

14.3.17.1 Offset

Register	Offset
IRQSIGEN	38h

14.3.17.2 Function

The IRQSIGEN is used to select which interrupt status is indicated to the host system as the interrupt. These status bits all share the same interrupt line. Setting any of these bits to 1 enables interrupt generation. The corresponding status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

14.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		RTOEIEN	DMAEIEN	Reserved	TNEIEN	ADMAEIEN	AC12EIEN	Reserved	DEBEIEN	DCEIEN	DTOEIEN	CIEIEN	CEBEIEN	CCEIEN	CTOEIEN
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			RTEIEN	Reserved			CINTIEN	CRMIEN	CINSIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

14.3.17.4 Fields

Field	Function
31-30 —	Reserved
29	Register timeout error interrupt enable 0b - Masked

Table continues on the next page...

Field	Function
RTOEIEN	1b - Enable
28 DMAEIEN	DMA error interrupt enable. 0b - Masked 1b - Enabled
27 —	Reserved
26 TNEIEN	Tuning error interrupt enable 0b - Masked 1b - Enable
25 ADMAEIEN	ADMA Error Interrupt Enable.
24 AC12EIEN	Auto CMD12 error interrupt enable. 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBEIEN	Data end bit error interrupt enable. 0b - Masked 1b - Enabled
21 DCEIEN	Data CRC error interrupt enable. 0b - Masked 1b - Enabled
20 DTOEIEN	Data timeout error interrupt enable. 0b - Masked 1b - Enabled
19 CIEIEN	Command index error interrupt enable. 0b - Masked 1b - Enabled
18 CEBEIEN	Command end bit error interrupt enable. 0b - Masked 1b - Enabled
17 CCEIEN	Command CRC error interrupt enable. 0b - Masked 1b - Enabled
16 CTOEIEN	Command timeout error interrupt enable. 0b - Masked 1b - Enabled
15-13 —	Reserved
12 RTEIEN	Re-tuning event interrupt enable 0b - Masked 1b - Enabled
11-9 —	Reserved

Table continues on the next page...

Field	Function
8 CINTIEN	Card interrupt interrupt enable. 0b - Masked 1b - Enabled
7 CRMIEN	Card removal interrupt enable. 0b - Masked 1b - Enabled
6 CINSIEN	Card insertion interrupt enable. 0b - Masked 1b - Enabled
5 BRRIEN	Buffer read ready interrupt enable. 0b - Masked 1b - Enabled
4 BWRIEN	Buffer write ready interrupt enable. 0b - Masked 1b - Enabled
3 DINTIEN	DMA interrupt enable. 0b - Masked 1b - Enabled
2 BGEIEN	Block gap event interrupt enable. 0b - Masked 1b - Enabled
1 TCIEN	Transfer complete interrupt enable. 0b - Masked 1b - Enabled
0 CCIEN	Command complete interrupt enable. 0b - Masked 1b - Enabled

14.3.18 Auto CMD Error Status Register / System Control 2 Register (AUTOCERR_SYSCTL2)

14.3.18.1 Offset

Register	Offset
AUTOCERR_SYSCTL2	3Ch

14.3.18.2 Function

When the Auto CMD12 error status bit in the AUTOC12ERR is set, the host driver checks this register to identify the kind of error indicated by the Auto CMD12. This register is valid only when the auto CMD12 error status bit is set.

Table 14-8. Relationship Between Command CRC Error and Command Timeout Error for Auto CMD12

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Type of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Changes in Auto CMD12 error status register (AUTOC12ERR) can be classified in three scenarios:

1. When the eSDHC is going to issue an Auto CMD12
   - Set bit 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
   - Set bit 0 to 0 if the Auto CMD12 is issued
2. At the end bit of an Auto CMD12 response
   - Check errors correspond to bits 1-4
   - Set bits 1-4 corresponding to detected errors
   - Clear bits 1-4 corresponding to detected errors
3. Before reading the Auto CMD12 error status bit 7
   - Set bit 7 to 1 if there is a command that cannot be issued
   - Clear bit 7 if there is no command to issue

The timing for generating the Auto CMD12 error and writing to the command register are asynchronous. After that, bit 7 should be sampled when the driver is not writing to the command register. So it is suggested to read this register only when IRQSTAT[AC12E] is set. An Auto CMD12 error interrupt is generated when one of the error bits (0-4) is set to 1. The command not issued by Auto CMD12 Error does not generate an interrupt.

14.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	AIE	Reserved						SMPCLKSEL	EXTN	Reserved			UHSM		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								CNIBAC12E	Reserved		AC12IE	AC12EBE	AC12CE	AC12TOE	AC12NE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.18.4 Fields

Field	Function
31 —	Reserved
30 AIE	Asynchronous Interrupt Enable. This bit can be set to 1 if a card supports asynchronous interrupts and asynchronous interrupt support is set to 1 in capability register. If this bit is set to 1, host driver can stop SDCLK during asynchronous inrrupt period to save power. During this period, eSDHC continues to deliver the card interrupt to the system when it is assrted by card. 0b - Disabled 1b - Enabled
29-24 —	Reserved
23 SMPCLKSEL	Sampling clock select. This bit is set by eSDHC during tuning procedure and valid after the completion of tuning (when execute tuning is cleared). Setting 1 by eSDHC means that tuning is completed successfully and setting 0 means that tuning is failed. Host driver should not write to this bit. Change of this bit is not allowed while eSDHC is receiving response or a read data block. 0b - Tuning procedure unsuccessful 1b - Tuning procedure completed successfully
22 EXTN	Execute Tuning. This bit is set to 1 by host driver to start tuning procedure and automatically cleared by eSDHC when tuning procedure is completed. The result of tuning is indicated to sampling clock select field. Tuning procedure is aborted by writing 0. 0b - Not tuned or tuning not compelted 1b - Execute tuning
21-19 —	Reserved

Table continues on the next page...

Field	Function
18-16 UHSM	UHS mode select UHS mode select. This field is used to select one of UHS-1 modes for SD 3.0 card; and select HS200 or DDR mode for eMMC device. Host driver should reset SDCLKEN in system control register before changing this field, and then set SDCLKEN again. 000b - SDR12 for SD, or max 52 MHz mode for SD 2.0 / eMMC 4.2 or older spec 001b - SDR25 for SD 010b - SDR50 for SD 011b - SDR104 for SD, HS200 for eMMC 100b - DDR 101b - Rest all the fields are reserved
15-8 —	Reserved
7 CNIBAC12E	Command not issued by Auto CMD12 error. Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04-D01) in this register. 0b - No error 1b - Not Issued
6-5 —	Reserved
4 AC12IE	Auto CMD index error. Occurs if the command index error occurs in response to a command. 0b - No error 1b - Error, the CMD index in response is not CMD12
3 AC12EBE	Auto CMD end bit error. Occurs when detecting that the end bit of command response is 0 which should be 1. 0b - No error 1b - End bit error generated
2 AC12CE	Auto CMD CRC error. Occurs when detecting a CRC error in the command response. 0b - No CRC error 1b - CRC error met in Auto CMD12 response
1 AC12TOE	Auto CMD timeout error. Occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to1, the other error status bits (2-4) have no meaning. 0b - No error 1b - Time out
0 AC12NE	Auto CMD12 not executed. If memory multiple block data transfer is not started, due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12. Setting this bit to 1 means the eSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (1-4) have no meaning. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. 0b - Executed 1b - Not executed

14.3.19 Host controller capabilities register (HOSTCAPBLT)

14.3.19.1 Offset

Register	Offset
HOSTCAPBLT	40h

14.3.19.2 Function

The HOSTCAPBLT provides the host driver with information specific to the eSDHC implementation. The value in this register is the power-on-reset value, and does not change with a software reset. Any write to this register is ignored.

14.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		AIS	SBS64B	Reserved	VS18	VS30	VS33	SRS	DMAS	HSS	ADMAS	Reserved	MBL		
W																
Reset	0	0	1	1	0	1	0	1	1	1	1	1	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.19.4 Fields

Field	Function
31-30 —	Reserved
29 AIS	Asynchronous Interrupt Support. This bit indicates whether the eSDHC supports SDIO asynchronous interrupt. Refer to SDIO Specification Version 3.0 0b - Asynchronous interrupt not supported 1b - Asynchronous interrupt supported
28 SBS64B	64-bit system bus support. This bit indicates that system supports 64-bit address descriptor mode and is connected to 64-bit adress system bus. 0b - 64-bit system bus not supported 1b - 64-bit system bus supported

Table continues on the next page...

Field	Function
27 —	Reserved
26 VS18	Voltage support 1.8V. This bit should depend on the host system ability. 0b - 1.8V not supported 1b - 1.8V supported
25 VS30	Voltage Support 3.0V. This bit shall depend on the Host System ability. 0b - 3.0V not supported 1b - 3.0V supported
24 VS33	Voltage support 3.3V. This bit should depend on the host system ability. 0b - 3.3V not supported 1b - 3.3V supported
23 SRS	Suspend/resume support. This bit indicates whether the eSDHC supports suspend/resume functionality. If this bit is 0, the suspend and resume mechanism, as well as the read wait, are not supported, and the host driver should not issue either suspend or resume commands. 0b - Not supported 1b - Supported
22 DMAS	DMA support. This bit indicates whether the eSDHC is capable of using the DMA to transfer data between system memory and the data buffer directly. 0b - DMA not supported 1b - DMA supported
21 HSS	High speed support. This bit indicates whether the eSDHC supports high speed mode and the host system can supply a SD clock frequency from 25-50 MHz. 0b - High speed not supported 1b - High speed supported
20 ADMAS	ADMA support. This bit indicates whether the eSDHC supports the ADMA feature. 0b - Advanced DMA not supported 1b - Advanced DMA supported
19 —	Reserved
18-16 MBL	Maximum block length. This value indicates the maximum block size that the host driver can read and write to the buffer in the eSDHC. The buffer should transfer block size without wait cycles. 000b - 512 bytes 001b - 1024 bytes 010b - 2048 bytes 011-111b - Reserved
15-0 —	Reserved

14.3.20 Watermark level register (WML)

14.3.20.1 Offset

Register	Offset
WML	44h

14.3.20.2 Function

Both write and read watermark levels (FIFO threshold) are configurable in the WML. They can range from 1- 128 words. Both write and read burst lengths are also configurable. They can range from 1- 16 beats.

14.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				WR_BRST_LEN				Reserved	WR_WML						
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				RD_BRST_LEN				Reserved	RD_WML						
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

14.3.20.4 Fields

Field	Function
31-28 —	Reserved
27-24 WR_BRST_LEN	Max write burst length. Burst length desirable for write on system bus when DMA is used. This is the maximum burst length, actual burst length may be less than this depending on other factors, such as block boundary

Table continues on the next page...

Field	Function
	0000b - 16 transfers in a single burst 0001b - 1 transfer in a single burst 0010b - 2 transfers in a single burst 1111b - 15 transfers in a single burst
23 —	Reserved
22-16 WR_WML	Write watermark level. The number of words (32-bit) used as the watermark level (FIFO threshold) for SD write in CPU polling mode. 0000000b - 128 words 0000001b - 1 word 0000010b - 2 words 1111111b - 127 words
15-12 —	Reserved
11-8 RD_BRST_LEN	Max read burst length. Burst length desirable for read on system bus when DMA is used. This is the maximum burst length, actual burst length may be less than this depending on other factors, such as block boundary 0000b - 16 transfers in a single burst 0001b - 1 transfers in a single burst 0010b - 2 transfers in a single burst 1111b - 15 transfers in a single burst
7 —	Reserved
6-0 RD_WML	Read watermark level. The number of words (32-bit) used as the watermark level (FIFO threshold) for SD read in CPU polling mode. 0000000b - 128 words 0000001b - 1 word 0000010b - 2 words 1111111b - 127 words

14.3.21 Force event register (FEVT)

14.3.21.1 Offset

Register	Offset
FEVT	50h

14.3.21.2 Function

The FEVT is not a physically implemented register. Rather, it is an address at which the interrupt status register (IRQSTAT) can be written if the corresponding bit of the interrupt status enable register (IRQSTATEN) is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register actually sets the corresponding bit of interrupt status register (IRQSTAT). A read from this register always results in zeros.

14.3.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				Reserved				Reserved							
W				FEVTDMAE			FEVTADMAE	FEVTAC12E		FEVTDEBE	FEVTDCE	FEVTDTOE	FEVTCIE	FEVTCEBE	FEVTCCE	FEVTCTOE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									Reserved						
W									FEVTCNIBAC12E			FEVTAC12IE	FEVTAC12EBE	FEVTAC12CE	FEVTAC12TOE	FEVTAC12NE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.21.4 Fields

Field	Function
31-29 —	Reserved
28 FEVTDMAE	Force event DMA error. Forces the IRQSTAT[DMAE] to be set.
27-26 —	Reserved
25	Force event ADMA error. Forces the IRQSTAT[ADMAE] to be set.

Table continues on the next page...

Field	Function
FEVTADMAE	
24 FEVTAC12E	Force event Auto CMD12 error. Forces IRQSTAT[AC12E] to be set.
23 —	Reserved
22 FEVTDEBE	Force event data end bit error. Forces IRQSTAT[DEBE] to be set.
21 FEVTDCE	Force event data CRC error. Forces IRQSTAT[DCE] to be set.
20 FEVTDTOE	Force event data time out error. Forces IRQSTAT[DTOE] to be set.
19 FEVTCIE	Force event command index error. Forces the IRQSTAT[CCE] to be set.
18 FEVTCEBE	Force event command end bit error. Forces IRQSTAT[CEBE] to be set.
17 FEVTCCE	Force event command CRC error. Forces IRQSTAT[CCE] to be set.
16 FEVTCTOE	Force event command time out error. Forces IRQSTAT[CTOE] to be set.
15-8 —	Reserved
7 FEVTCNIBAC12E	Force event command not executed by Auto CMD12 error. Forces AUTOC12ERR[CNIBAC12E] to be set.
6-5 —	Reserved
4 FEVTAC12IE	Force event Auto CMD12 index error. Forces AUTOC12ERR[AC12IE] to be set.
3 FEVTAC12EBE	Force event Auto CMD12 end bit error. Forces AUTOC12ERR[AC12EBE] to be set.
2 FEVTAC12CE	Force event Auto CMD12 CRC error. Forces AUTOC12ERR[AC12CE] to be set.
1 FEVTAC12TOE	Force event Auto CMD12 time out error. Forces the AUTOC12ERR[AC12TOE] to be set.
0 FEVTAC12NE	Force event Auto CMD12 not executed. Forces AUTOC12ERR[AC12NE] to be set.

14.3.22 ADMA error status register (ADMAES)

14.3.22.1 Offset

Register	Offset
ADMAES	54h

14.3.22.2 Function

When an ADMA error interrupt occurs, the ADMA error states field in the ADMAES holds the ADMA state and the ADMA system address register (ADSADDR) holds the address around the error descriptor.

For recovering from this 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 (ADSADDR) is the error descriptor address
- ST_FDS: Current location set in the ADMA system address register (ADSADDR) is the error descriptor address
- ST_CADR: This state is never set because it only increments the descriptor pointer and does not generate an ADMA error
- ST_TFR: Previous location set in the ADMA system address register (ADSADDR) is the error descriptor address

In case of a write operation, the host driver should use the ACMD22 to get the number of the 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) in the ST_FDS state. The host driver can distinguish this error by reading the valid bit of the error descriptor.

Table 14-9. ADMA Error State Coding

D30-D31	ADMA Error State (When Error Has Occurred)	Contents of ADMA System Address Register
00	IDLE (idle)	Current descriptor address on which ADMA error occured
01	FETCH_DESC (fetch descriptor)	Current descriptor address on which ADMA error occured
10	DATA_XFER (data transfer)	Current descriptor address on which ADMA error occured
11	WAIT_STOP (Wait for ADMA to stop)	Current descriptor address on which ADMA error occured

14.3.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											ADMAIBE	ADMADCE	ADMALME	ADMAES	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.22.4 Fields

Field	Function
31-5 —	Reserved
4 ADMAIBE	ADMA internal bus error. This bit indicates that a system bus error occurred while ADMA transaction was underway. It is set when error response is received on the system bus. 0b - No error 1b - Error
3 ADMADCE	ADMA descriptor error. This error occurs when invalid descriptor fetched by ADMA. 0b - No error 1b - Error
2 ADMALME	ADMA length mismatch error. This error occurs in the following two cases: • While XFERTYP[BCEN] 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 can not be divided by the block length 0b - No error 1b - Error
1-0 ADMAES	ADMA error state (when ADMA error is occurred). This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. Refer to the table above for more details.

14.3.23 ADMA system address low register (ADSADDRL)

14.3.23.1 Offset

Register	Offset
ADSADDRL	58h

14.3.23.2 Function

The ADSADDRL contains the physical system memory address used for ADMA transfers.

14.3.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADS_ADDR_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADS_ADDR_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.23.4 Fields

Field	Function
31-0 ADS_ADDR_LOW	ADMA system address low. This register holds lower 32-bit address of executing command of the descriptor table. At the start of ADMA, the host driver should set start address of the descriptor table. The ADMA increments this register address, which points to next line, when every fetching a descriptor line. When the ADMA error interrupt is generated, this register should hold valid descriptor address depending on the ADMA state. The host driver should program descriptor table on 32-bit boundary and set 32-bit boundary address to this register. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). The host driver should initialize this register before starting an ADMA transaction.

14.3.24 ADMA system address high register (ADSADDRH)

14.3.24.1 Offset

Register	Offset
ADSADDRH	5Ch

14.3.24.2 Function

The ADSADDRH contains the physical system memory address used for ADMA transfers.

14.3.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADS_ADDR_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADS_ADDR_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.24.4 Fields

Field	Function
31-0 ADS_ADDR_HIGH	ADMA system address high. This register holds higher 32-bit address of executing command of the descriptor table. At the start of ADMA, the host driver should set start address of the descriptor table. The ADMA increments this register address, which points to next line, when every fetching a descriptor line. When the ADMA error interrupt is generated, this register should hold valid descriptor address depending on the ADMA state. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). The host driver should initialize this register before starting an ADMA transaction.

14.3.25 Host controller version register (HOSTVER)

14.3.25.1 Offset

Register	Offset
HOSTVER	FCh

14.3.25.2 Function

The HOSTVER contains the vendor host controller version information. All bits are read only and will read the same as the power-reset value.

14.3.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VVN								SVN							
W																
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0

14.3.25.4 Fields

Field	Function
31-16 —	Reserved
15-8 VVN	Vendor version number. These status bits are reserved for the vendor version number. The host driver should not use this status. Patterns not shown are reserved. 00000000b - eSDHC Version 1.0 00010000b - eSDHC Version 2.0

Table continues on the next page...

Field	Function
	00010001b - eSDHC Version 2.1 00010010b - eSDHC Version 2.2 00010011b - eSDHC Version 2.3 00100000b - eSDHC Version 3.0 00100001b - eSDHC Version 3.1 00100010b - eSDHC Version 3.2
7-0 SVN	Specification version number. These status bits indicate the host controller specification version. Patterns not shown are reserved. 00000000b - SD Host Specification Version 1.0 00000001b - SD Host Specification Version 2.0, supports test event register , and ADMA 00000010b - SD Host Specification Version 3.0

14.3.26 DMA error address low register (DMAERRADDRL)

14.3.26.1 Offset

Register	Offset
DMAERRADDRL	104h

14.3.26.2 Function

The DMAERRADDRL contains the address of the transaction on which DMA error occured.

14.3.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_ADDRn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_ADDRn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.26.4 Fields

Field	Function
31-0 DMA_ADDRn	DMA error address. This field contains the system low address of the transaction on which DMA error occured.

14.3.27 DMA error address high register (DMAERRADDRH)

14.3.27.1 Offset

Register	Offset
DMAERRADDRH	108h

14.3.27.2 Function

The DMAERRADDRH contains the address of the transaction on which DMA error occured.

14.3.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_ADDR_HIGHn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_ADDR_HIGHn															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.27.4 Fields

Field	Function
31-0 DMA_ADDR_HIGHn	DMA error address. This field contains the system high address of the transaction on which DMA error occured.

14.3.28 DMA error attribute register (DMAERRATTR)

14.3.28.1 Offset

Register	Offset
DMAERRATTR	10Ch

14.3.28.2 Function

The DMAERRATTR contains attributes of the transaction on which DMA error occured.

14.3.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									DMA_SIZE			DMA_LEN			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.28.4 Fields

Field	Function
31-7 —	Reserved
6-4 DMA_SIZE	System bus burst size. This field contains burst size of the transaction on which DMA error occured.
3-0 DMA_LEN	System bus burst length. This field contains burst length of the transaction on which DMA error occured.

14.3.29 Host controller capabilities register 2 (HOSTCAPBLT2)

14.3.29.1 Offset

Register	Offset
HOSTCAPBLT2	114h

14.3.29.2 Function

This register provides the host driver with information specific to the eSDHC implementation. The value in this register is the power-on-reset value, and does not change with a software reset. Any write to this register is ignored.

14.3.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RTM		UTSDR50	Reserved	TCRT				Reserved	DTDS	DTCS	DTAS	Reserved	DDR50	SDR104	SDR50
W																
Reset	1	0	1	0	1	1	1	1	0	0	0	0	0	1	1	1

14.3.29.4 Fields

Field	Function
31-16 —	Reserved
15-14 RTM	Re-tuning modes. This bit indicates the supported re-tuning modes for UHS. 00b - Mode 1 - Software Timer 01b - Mode 2 - Software Timer and Re-tuning request 10b - Mode 3 - Software Timer, and Auto Re-tuning during data transfer 11b - Reserved
13 UTSDR50	Use tuning for SDR50. This bit indicates whether the host support tuning for SDR50 mode. 0b - Tuning for SDR50 mode not supported 1b - Tuning for SDR50 mode supported
12 —	Reserved
11-8 TCRT	Timer Count for Re-Tuning : This field indicates an initial value of Re-Tuning timer for retuning mode 1 to 3. 0000b - Re-Tuning timer disabled 0001b - 1 second 0010b - 2 seconds 0011b - 4 seconds 1011b - 1024 seconds 1100-1110b - Reserved 1111b - Get timer information from other source.
7 —	Reserved
6 DTDS	Driver type D support. This bit indicates whether the system is capable of using driver type D. 0b - Driver Type D not supported 1b - Driver Type D Supported

Table continues on the next page...

Field	Function
5 DTCS	Driver type C support. This bit indicates whether the system is capable of using driver type C. 0b - Driver Type C not supported 1b - Driver Type C Supported
4 DTAS	Driver type A support. This bit indicates whether the system is capable of using driver type A. 0b - Driver Type A not supported 1b - Driver Type A Supported
3 —	Reserved
2 DDR50	DDR50 support. This bit indicates whether the eSDHC supports the DDR mode. 0b - DDR mode not supported 1b - DDR mode supported
1 SDR104	SDR104 Support. This bit indicates whether the eSDHC supports the SDR104. 0b - SDR104 not supported 1b - SDR104 supported
0 SDR50	SDR50 support. This bit indicates whether the eSDHC supports the SDR50. 0b - SDR50 not supported 1b - SDR50 supported

14.3.30 Tuning block control register (TBCTL)

14.3.30.1 Offset

Register	Offset
TBCTL	120h

14.3.30.2 Function

This register contains fields for controlling the tuning block.

NOTE

Writing or reading to reserved fields of this register does not guarantee specific values will be written or read.

14.3.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	Reserved	Reserved	Reserved					Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved		Reserved				Reserved	Reserved		Reserved	Reserved	TB_EN	TB_MODE	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

14.3.30.4 Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 —	Reserved
28-24 —	Reserved
23-16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved
11-8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4	Reserved

Table continues on the next page...

Field	Function
—	
3 —	Reserved
2 TB_EN	Tuning block enabled. Tuning block should be enabled for high speed SDR mode more than 50 MHz SD clock frequency. This bit is not reset by software reset for all. 0b - Tuning block is disabled 1b - Tuning block is enabled
1-0 TB_MODE	Tuning Mode Tuning Mode. Selects tuning mode when tuning block is enabled. Refer to re-tuning modes in Table 14-11 00b - Mode 1 - Software Timer 01b - Mode 2 - Software Timer, and Re-tuning request 10b - Mode 3 - Software Timer, and Auto Re-tuning during data transfer 11b - SW tuning mode - Software tuning mode where start and end point of data window needs to be programmed in TBPTR register and use software timer for re-tuning

14.3.31 Tuning block status register (TBSTAT)

14.3.31.1 Offset

Register	Offset
TBSTAT	124h

14.3.31.2 Function

This register contains the status of the tuning block.

14.3.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TB_STATUS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TB_STATUS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.31.4 Fields

Field	Function
31-0 TB_STATUS	Tuning Status.

14.3.32 Tuning block pointer register (TBPTR)

14.3.32.1 Offset

Register	Offset
TBPTR	128h

14.3.32.2 Function

This register contains fields for controlling tuning block pointers.

14.3.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	TB_WNDW_STRT_PTR							Reserved	TB_WNDW_END_PTR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.32.4 Fields

Field	Function
31-15 —	Reserved
14-8 TB_WNDW_STRT_PTR	Tuning window start pointer. Selects window start pointer for software tuning mode (when TBCTL[TB_MODE]=3).
7 —	Reserved
6-0 TB_WNDW_END_PTR	Tuning window start pointer. Selects window end pointer for software tuning mode (when TBCTL[TB_MODE]=3)

14.3.33 SD direction control register (SDDIRCTL)

14.3.33.1 Offset

Register	Offset
SDDIRCTL	140h

14.3.33.2 Function

This register contains control for CMD and DAT lines direction.

14.3.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													DIR_CTL		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

14.3.33.4 Fields

Field	Function
31-3 —	Reserved
2-0 DIR_CTL	Direction control Specify the turnaround time required for external transceiver after the assertion of direction pins in number of SD clocks 000b - No turnaround time required 001b - 1 SD clock period for turnaround 010b - 2 SD clock periods for turnaround 111b - 7 SD clock periods for turnaround

14.3.34 SD Clock Control Register (SDCLKCTL)

14.3.34.1 Offset

Register	Offset
SDCLKCTL	144h

14.3.34.2 Function

This register contains fileds for controlling SD external and loopback clock.

14.3.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LPBK_CLK_SEL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_CLK_CTL	Reserved	Reserved		Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.34.4 Fields

Field	Function
31 LPBK_CLK_SEL	SD Loopback Clock Select. This field specifies whether SD clock is loopbacked from external pin or from internal pad. 0b - SD card clock is loopbacked from internal pad 1b - SD card clock is loopbacked from external pin
30-16 —	Reserved
15	Command Logic Clock Control. This field specifies controls clock to the command logic.

Table continues on the next page...

Field	Function
CMD_CLK_CTL	0b - Command logic clock is same as data logic clock 1b - Command logic clock is 25% shifted early from data logic clock
14 —	Reserved
13-12 —	Reserved
11-0 —	Reserved

14.3.35 eSDHC control register (ESDHCCTL)

14.3.35.1 Offset

Register	Offset
ESDHCCTL	40Ch

14.3.35.2 Function

This register contains fields for controlling DMA transfers.

14.3.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved										RTOCV		PCS	FAF	RTR	CRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			RD_PRFTCH_BLKCNT					PAD_DIS	SNOOP	Reserved		WR_BUF	RD_SAFE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.3.35.4 Fields

Field	Function
31-22 —	Reserved
21-20 RTOCV	Register timeout count value. This field define the timeout value for register access. 00b - Timeout count value for register access is 2^10 clocks 01b - Timeout count value for register access is 2^11 clocks 10b - Timeout count value for register access is 2^12 clocks 11b - Timeout count value for register access is 2^13 clocks
19 PCS	Peripheral clock select. This bit selects the clock used for generating SD clock. This bit is not reset by software reset for all. 0b - Platform clock is used 1b - Peripheral/2 clock is used
18 FAF	Flush asynchronous FIFO. This bit is used to flush asynchronous FIFO. It can be set by software and will be auto cleared by hardware. 0b - No Flush 1b - Flush async FIFO
17 RTR	Re-tuning request. This bit indicates re-tuning request status. eSDHC may request host driver to execute re-tuning sequence, by setting this bit, when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This bit is cleared when a command is issued with setting execute tuning in the system control 2 register. Changing of this bit from 0 to 1 generates re-tuning event. Refer to interrupt status registers for more detail. This bit isn't set to 1 if tuning block enable in the tuning control register is set to 0 (using fixed sampling clock). This is read-only field. 0b - No re-tuning request 1b - Re-tuning request is set
16 CRS	Clock register select. This bit selects the clock division format of SDCLKFS and DVS fields of system control register.

Table continues on the next page...

Field	Function
	0b - SDCLKFS is defined as 8-bit field, and DVS is active in system control register 1b - SDCLKFS is defined as 10-bit field, CGS is active in system control register
15-13 —	Reserved
12-8 RD_PRFTCH_BLKCNT	Read prefetch block count. For DMA read (SD write), this field specifies the maximum number of SD blocks that the host can prefetch from the system memory. It can have following values: 00000b - No prefetch 00001b - 1 SD block prefetch 00010b - 2 SD block prefetch 11111b - 31 SD block prefetch
7 PAD_DIS	Pad disable Pad disable. Padding disable control for DMA transaction with non-word (4-bytes) aligned block size. For more details refer to Data buffer and block size. 0b - DMA will pad data at the end each block transfer. 1b - DMA will not pad data at the end of each block transfer.
6 SNOOP	Snoop attribute. Snoop enable for DMA transaction. 0b - DMA transactions are not snooped by the CPU data cache. 1b - DMA transactions are snooped by the CPU data cache.
5-4 —	Reserved
3 WR_BUF	Write bufferable. DMA always initiate write transaction on System bus corresponding to last in every SD block as non-bufferable. This bit specifies whether all non-last write transactions will be bufferable or not. 0b - Non-last write transactions are not bufferable. 1b - Non-last write transactions are bufferable.
2 RD_SAFE	Read safe. This bit should be set only if the target of read dma operation is a well behaved memory which is not affected by the read operation and will return the same data if read again from the same location. This means that unaligned reading operation can be rounded up to enable more efficient read operations. 0b - It is not safe to read more bytes that were intended. 1b - It is safe to read more bytes that were intended.
1-0 —	Reserved

14.4 Functional description

The eSDHC block is partitioned in five major sub-blocks as shown in Figure 14-2.

- SD interface and control unit

  This block interfaces with the SD bus. It is mainly responsible for controlling the SD bus operation and transferring data from Data Buffer and SD bus. It also interacts with System Interface and Control Unit.

- System interface and control unit

This block interfaces with the system interfaces (that is, the system bus in DMA mode and register bus in CPU polling mode) and transfers the card data from the data buffer and system.

- Register bank

  This block interfaces with register bus and contains all the registers. It controls the overall operation of eSDHC and also provides status through various registers.

- Clock and reset

  This block generates divided clock for SD interface and provides appropriate reset to all the blocks.

- SD monitor

  This block monitors the SD bus and provides status to register banks, such as card interrupt, write protect and card detect.

14.4.1 System interface and control unit (SysICU)

The SysICU block is further partitioned into three major sub-blocks:

- System control block

  This sub-block receives data transfer request from the register bank, controls data transfer for whole transaction and generates control signals for DMA and buffer control operation on per block basis.

- Buffer control block

  This sub-block handles buffer port register access in CPU polling mode and contains the buffer FIFO to store transfer data. It controls the data transfer per-block basis in the CPU polling mode.

- DMA block

  This sub-block generates DMA transactions on system bus to transfer the data between system memory and data buffer.

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA system interface, register bank as well as IP bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

14.4.1.1 Data buffer

The eSDHC uses one configurable data buffer, so that data can be transferred between the system/register bus and SD card in an optimized manner to maximize throughput between the two clock domains. See the figure below for illustration of the buffer scheme.

The buffer is used as temporary storage for data being transferred between the host system and the card.

The watermark levels for read and write are both configurable, and can be any number from 1-128 words for CPU polling mode. The burst lengths for read and write are also configurable, and can be any number from 1-16 for DMA mode.

Figure 14-3. eSDHC buffer scheme

There are two transfer modes to access the data buffer:

- CPU polling mode

  Data is transferred over register bus. For a host read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, then by monitoring (polling or interrupt) the BRR bit the host driver can read the buffer data port register (DATPORT) to fetch the amount of words set in the RD_WML

register from the buffer. For block size integral multiple of watermark level value set in watermark level register (WML), driver must access buffer data port register (DATPORT) exactly the same number of times as the watermark level. However, if the block size is not the times of the watermark level value, the software must access exactly the remained number of words at the end of each block. For example, for read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block count is 2, then the access times to Data Buffer Port Register for the burst sequence in the whole transfer process must be 4, 4, 2(for first block); 4, 4, 2(for second block). The write operation is similar.

- DMA mode (includes simple and advanced DMA accesses)

  Data is transferred over system bus. DMA interrupt is generated after all the data is transferred to/from the buffer.

14.4.1.1.1 Write operation sequence

There are two ways to write data into the buffer when the user transfers data to the card:

- By processor core polling through IRQSTAT[BWR] (interrupt or polling)
- By using the DMA

When the DMA is not used, (CPU polling mode, that is, the XFERTYP[DMAEN] is not set when the command is sent), and when the amount of buffer space exceeds the value set in the WR_WML register, PRSSTAT[BWR] is set. The buffer write ready interrupt is generated if it is enabled by software.

When DMA is used, the eSDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the eSDHC aborts the data transfer and abandons the current block. If the current data transfer is in multi block mode, the eSDHC does not automatically send CMD12, even though the XFERTYP[AC12EN] is set. The host driver needs to send CMD12 in this scenario. It is recommended that a Software Reset for Data be applied before the transfer is re-started after error recovery.

14.4.1.1.2 Read operation sequence

There are two ways to read data from the buffer when the user transfers data to the card:

- By processor core polling through IRQSTAT[BRR] (interrupt or polling)
- By using the DMA

When DMA is not used (CPU polling mode, that is, the XFERTYP[DMAEN] is not set when the command is sent), and when the amount of data exceeds the value set in the RD_WML register, that is available and ready for system fetching data, PRSSTAT[BRR] is set. The buffer read ready interrupt will be generated if it is enabled by software.

When DMA is used, the eSDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the eSDHC will abort the data transfer and abandon the current block. If the current data transfer is in multi block mode, the eSDHC will not automatically send CMD12, even though the XFERTYP[AC12EN] is set. The host driver should send CMD12 in this scenario. It is recommended that a software reset for data be applied before the transfer is re-started after error recovery.

14.4.1.1.3 Data buffer and block size

In eSDHC, the data buffer can hold up to 128 words (32-bit).

The watermark levels for both write and read can be configured for CPU polling mode. The watermark level can be from one word to a maximum of 128 words. For both DMA read and write, the burst length can be configured from one to a maximum of 16. The host driver may configure watermark level and burst length value according to the system situation and requirement in the watermark level register (WML).

During a multi-block data transfer, the block length may be set to any value between 1 and 2048 bytes inclusive that satisfies the requirements of the external card. The only restriction to it can be from the external card, which may not support that large a block or partial access to block (which is not an integer times of 512 bytes).

For CPU polling mode, when block size not a multiple of four; that is, not word aligned, eSDHC requires stuff bytes at the end of each block, as defined in the host specification. For example, if the block size is 5 bytes and there are two blocks to write, there must be two register bus writes to buffer data port register (DATPORT) for each block; and for each block, the ending non-word aligned bytes should be stuffed in buffer data port register (DATPORT) write to make it word aligned. For this example, 3 bytes should be stuffed. eSDHC will transfer only the required number of bytes to the card and ignore the stuff bytes as shown in the figure below. Read operation is similar.

Figure 14-4. Byte stuffing for CPU polling mode

For DMA mode, when block size not a multiple of four; that is, not word aligned, eSDHC requires stuff bytes depending on the PAD_DIS field programmed in DMA Control register. The data transfer with byte stuffing enabled (when DMACTL[PAD_DIS]=0) is shown in Figure 14-5. And, data transfer with byte stuffing disabled (when DMACTL[PAD_DIS]=1) is shown in Figure 14-6. Transfer with byte stuffing disabled eliminates the software overhead of byte stuffing. Driver may program DMA Control register accordingly.

Figure 14-5. Byte stuffing for DMA mode when DMACTL[PAD_DIS]=0

Figure 14-6. No byte stuffing for DMA mode when DMACTL[PAD_DIS]=1

14.4.1.1.4 Dividing large data transfer

This SDIO command CMD53 definition, limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in block size units. If the total data length cannot be divided evenly into a multiple of the block size, then based on the function and the card design, there are two ways to transfer the data. First option is for the host driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Second option is to add dummy data in the last block to fill the block size provided the card manages the removal of the dummy data.

Figure below shows an example which explains the dividing of large data transfers. In this figure, assume a kind of WLAN SDIO card that only supports block size up to 64 bytes. Although the eSDHC supports a block size of up to 2048 bytes, the SDIO can only accept a block size less than 64 bytes. Thus, the data must be divided (see example below).

Figure 14-7. Example for dividing large data transfers

14.4.1.1.5 Byte order (endianness) of buffer data port register

For CPU polling mode, sequential and contiguous access is necessary to ensure the pointer address value is correct. Random or skipped access is not possible.

The byte order of buffer data port register (DATPORT), by reset is little endian mode. The data buffer is always little endian. The data is swapped inside the buffer, according to the endian mode configured by software in PROCTL[EMODE].

In little endian mode, the data is transferred between data buffer and buffer data port register (DATPORT) without any swapping. In big endian mode, the data between data buffer and buffer data port register (DATPORT) is byte-swapped as shown in the figure given below. For an SD write operation, byte order is swapped after data is fetched from the buffer port register and sent to data buffer. For a host read operation, byte order is swapped after the data is read from data buffer and sent to buffer port register.

Figure 14-8. Data swap between buffer data port register and data buffer in big endian mode

14.4.1.2 DMA system interface

The DMA implements a DMA engine and the system master.

The eSDHC supports both SDMA and ADMA (ADMA1 and ADMA2). Figure below illustrates the DMA system interface block.

Figure 14-9. DMA system interface block

14.4.1.2.1 DMA burst length

The actual burst length on system bus depends on the shortest of following factors:

- Burst length configured in the burst length field of the watermark level register (WML)
- Block size boundary
- Data boundary configured in the current descriptor (if the ADMA is active)

14.4.1.2.2 System master interface

The System DMA engine can at times fail during data transfer.

When this error occurs:

- The DMA engine stops the transfer and goes to the error state.
- The internal data buffer stops accepting incoming data.
- The IRQSTAT[DMAE] is set to inform the driver.

Once the DMAE interrupt is received, the software sends a CMD12 to abort the current transfer and read the DMAERRADDRH/L[DMA_ADDR] bits to get the starting address of the corrupted block. For error recovery, the software issues a data reset and re-starts the transfer from this address to recover the corrupted block.

14.4.1.3 Single DMA (SDMA)

SDMA is single operation DMA, that is, data is transferred for single operation only (unlike ADMA which has chained descriptor table) from the starting address programmed in SDMA system address register (DSADDR) for entire data transfer size (block count x block size) as programmed in the block attributes register (BLKATTR) if XFERTYP[BCEN] is set.

14.4.1.3.1 SDMA error

SDMA will stop whenever an error is encountered on the system bus.

DMA error address low register latches the transaction address on which the error occured.

14.4.1.4 Advanced DMA (ADMA)

Advanced DMA (ADMA) is a new DMA transfer algorithm, which is defined in the SD Host Controller Standard.

For single DMA, the data can be transferred for single operation only. The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. Higher speed DMA transfers are realized because the host MCU intervention is not needed during long DMA based data transfers.

The host controller has two types of ADMA: ADMA1 and ADMA2. ADMA1 can support data transfer of 4KB aligned data in system memory. ADMA2 improves the restriction so that data of any location and any size can be transferred in system memory. Their formats of descriptor table are different.

ADMA can recognize all kinds of descriptors defined in SD Host Controller Standard and if 'End' flag is detected in the descriptor, ADMA stops after this descriptor is processed.

eSDHC supports ADMA1 and ADMA2 with both 32-bit and 64-bit addressing.

14.4.1.4.1 ADMA concept and descriptor format

ADMA1 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and End flag in descriptor

ADMA2 includes the following descriptors:

- Valid/Invalid descriptor
- Nop descriptor
- Rsv descriptor
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and End flag in descriptor

Figure 14-10 explains the ADMA1 descriptor table format.

Figure 14-12 explains the ADMA2 descriptor table format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it will ignore the high 32-bit. Address field should be set on word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

ADMA starts read/write operation after it reaches the Tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before Tran type descriptor. Every Tran type triggers a transfer, and the transfer data length is extracted from the most recent Set type descriptor. If there is no set type descriptor after the previous Trans descriptor, the data length is the value for previous transfer, or 4 Kbyte if no set descriptor is ever met.

For ADMA2, Tran type descriptor contains both data length and transfer data address. Thus, only a Tran type descriptor can start a data transfer.

Address/Page Field		Address/Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Addressor Data Length		000000		Act2	Act1	0	Int	End	Valid

Act2	Act1	Symbol	Comment	31-28	27-12
0	0	NOP	No Operation	Do not care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid=1 indicates this line of descriptor is effective. If Valid=0, generate ADMA error interrupt and stop ADMA.
End	End=1 indicates current descriptor is the ending descriptor.
Int	Int=1 generates DMA Interrupt when this descriptor is processed.

Figure 14-10. Format of the 32-bit address ADMA1 descriptor table

Figure 14-11. Concept and access method of ADMA1 descriptor table

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	6	5	4	3	2	1	0
32-bit Address		16-bit Length		0000000000		Act2	Act1	0	Int	End	Valid

Act2	Act1	Symbol	Comment	Operation
0	0	NOP	No Operation	Do not care
0	1	Rsv	Reserved	Same as NOP. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid=1 indicates this line of descriptor is effective. If Valid=0, generate ADMA error interrupt and stop ADMA.
End	End=1 indicates current descriptor is the ending descriptor.
Int	Int=1 generates DMA Interrupt when this descriptor is processed.

Figure 14-12. Format of the 32-bit address ADMA2 descriptor table

Table 14-10. Format of the 64-bit address ADMA2 descriptor table

Address field		Length		Reserved			Attribute field				
95	32	31	16	15	06	05	04	03	02	01	00
64-bit Address		16-bit length		0000000000		Act2	Act1	0	Int	End	Valid

Figure 14-13. Concept and access method of ADMA2 descriptor table

14.4.1.4.2 ADMA interrupt

If the 'interrupt' flag of descriptor is set, ADMA will generate an interrupt, IRQSTAT[DINT] whenever the data transfer for the particular descriptor line is completed.

14.4.1.4.3 ADMA error

The ADMA stops whenever any of the following error is encountered:

- Fetching descriptor error
- System response error
- Data length mismatch error

ADMA descriptor error is generated when it fails to detect 'Valid' flag in the descriptor. If ADMA descriptor error occurs, the interrupt is not generated even if the 'Interrupt' flag of this descriptor is set.

When the BLKCNTEN bit is set, the data transfer length set in buffer must be equal to the whole data transfer length set in the descriptor nodes, otherwise data length mismatch error is generated.

If the BLKCNTEN bit is not set, the whole data transfer length set in descriptor should be times of block length, otherwise, when all data set in the descriptor nodes are done, the data length mismatch error will occur.

The DMA error address low/high register (DMAERRADDRL/H) latches the transaction address on which the error occured. The ADMA system address low/high register (ADSADDRL/H) latches the current descriptor line address which encountered the error.

14.4.2 SD interface and control unit (SDICU)

This block is partitioned into six major sub-blocks as shown in Figure 14-2.

- Transfer control block: This sub-block receives command request from the register bank and overall controls other SDICU sub-blocks for SD bus operation on transaction level.
- SD command block: This sub-block controls the SD CMD line for sending command out. It receives the command request from Transfer Control block and sends it on SD CMD line.
- SD response block: This sub-block monitors the SD CMD line for receiving response and detecting line errors. It sends the command and response status to transfer control for transaction operation.
- SD data out block: This sub-block controls the SD DAT lines for sending out the block write data from the SD data out FIFO.
- SD data in block: This sub-block monitors the SD DAT line for detecting read and write data block end and busy period end for transfer control module. It also detects data line error and card interrupt.
- Tuning block: This sub-block receives Tuning block data from card and tunes IP. It then forwards the sampled data to Async FIFO for further operation.

14.4.2.1 Command CRC

See the figure below for an illustration of the structure for the command CRC shift register.

Figure 14-14. Command CRC shift register

The CRC polynomials for the CMD are as follows:

```
Generator polynomial: G(x) = x^7 + x^3 + 1
M(x) = (first bit) * x^n + (second bit) * x^(n-1) +...+ (last bit) * x^0
CRC[6:0] = Remainder [(M(x) * x^7) / G(x)]
```

14.4.2.2 Data CRC

The CRC polynomials for the DAT are as follows:

```
Generator polynomial: G(x) = x^16 + x^12 + x^5 +1 M(x)
      = (first bit) * x^n + (second bit) * x^(n-1) +...+ (last bit) * x^0
     CRC[15:0] = Remainder [(M(x) * x^16) / G(x)]
```

14.4.2.3 Tuning block (SDICU)

Tuning block is used for SD UHS SDR50, SDR104 and eMMC HS200 modes.

Tuning block tunes the IP on tuning command (CMD19 for SD, CMD21 for eMMC) data sent by card and the sampled data is sent to async FIFO for further operation.

This block oversamples data from card and selects particular sampling point after completion of tuning procedure, refer to Tuning block procedure for tuning block usage. Various re-tuning modes are supported by eSDHC as described below.

Table 14-11. Re-tuning modes

Re-Tuning Mode	Re-Tuning Method	Data Length
1	Software timer	4MB (Max.)
2	Software timer, and re-tuning request	4MB (Max.)
3	Software timer, and auto re-tuning during data transfer	Any

There are two re-tuning timings: Re-tuning request controlled by eSDHC and expiration of a re-tuning timer(software timer) controlled by the host driver. By receiving either timing, the host driver executes the re-tuning procedure just before a next command issue.

The maximum data length per read/write command is restricted so that re-tuning procedures can be inserted during data transfers.

Re-tuning mode 1

eSDHC do not generate re-tuning request. In this case, the host driver should maintain all re-tuning timings by using a re-tuning timer. To enable inserting the re-tuning procedure during data transfers, the data length per read/write command shall be limited up to 4 MB.

Re-tuning mode 2

eSDHC indicates the re-tuning timing by re-tuning request during data transfers. Then the data length per read/write command shall be limited up to 4 MB. During non data transfer, re-tuning timing is determined by re-tuning timer.

Re-tuning mode 3

eSDHC takes care of the re-tuning during data transfer (auto re-tuning). Re-tuning request will not be generated and there is no limitation to data length per read/write command. During non data transfer, re-tuning timing is determined by re-tuning timer.

Re-tuning timer control example for re-tuning mode 1

The software timer starts counting by loading the initial value. When the timer expires, the host driver marks an expiration flag. On receiving a command request, the host driver checks the expiration flag. If the expiration flag is set, then the host driver should perform the re-tuning procedure before issuing a command. If the expiration flag is not set, then the host driver issues a command without performing the re-tuning procedure. Every time the re-tuning procedure is performed, the timer loads the new initial value and the expiration flag is cleared.

Re-tuning timer control example for re-tuning mode 2 and mode 3

The software timer control is almost the same as re-tuning mode 1 except the timer loads the new initial value after data transfer (when receiving transfer complete). in case of mode 3, timer count for re-tuning is set either smaller value: tuning effective time after re-tuning procedure or after data transfer.

If a host system goes into power down mode, the host driver should stop the re-tuning timer and set the expiration flag to 1 when the host system resumes from power down mode.

14.4.3 Register bank

This block interfaces with register bus and it contains all the registers. It controls the overall operation of eSDHC and also provides status through various registers.

Register accesses is actually on the register bank. See the figure below for the block diagram.

Figure 14-15. Register bank diagram

Partial access is not allowed on any register; that is, it should be 32-bit access only.

14.4.4 Clock and reset module

Clock and reset module generates divided clock for SD interface and provides appropriate reset to other modules.

There are four kinds of reset signals within eSDHC:

- Hardware reset
- Software reset for all
- Software reset for the data part
- Software reset for the command part

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules.

If the internal data buffer is in danger, and the SD clock must be gated off to avoid buffer over/under-run, this module asserts the gate of the output SD clock to shut the clock off. After the buffer danger has recovered, and when the system access of the buffer catches up, the clock gate of this module opens and the SD clock becomes active again.

14.4.4.1 Clock generator

The clock generator generates the SDHC_CLK by source clock in two stages.

The figure below illustrates the structure of the divider. The term "Base" represents the frequency of the source clock.

Figure 14-16. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV), which can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler, and outputs the actual clock (SDHC_CLK). This clock is the driving clock for sub modules SD Command and SD Data Out in SD Interface and Control Unit (refer to Figure 14-2) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage, can be DIV, DIV/2, DIV/4,..., or DIV/256. Thus, the highest frequency of the SDHC_CLK is Base, and the next highest is Base/2, while the lowest frequency is Base/4096.

The figure below illustrates the sequence for changing the SD clock frequency.

Figure 14-17. SD clock frequency change sequence

Base clock can be selected by programming ESDHCCTL[PCS]. It selects between platform clock and peripheral clock / 2. Base clock is divided by two of peripheral clocks when PCS=1.

1. Clear SYSCTL[SDCLKEN]
2. Wait for PRSSTAT[SDSTB] to be set
3. Program appropriate value of ESDHCCTL[PCS]
4. Set SYSCTL[SDCLKEN]
5. Wait for PRSSTAT[SDSTB] to be set

14.4.5 SD monitor

The module detects the CD_B (card detection) as well as the DAT3 signal. The transceiver reports the card insertion state according to the CD_B state, the signal level on the DAT3 signal.

NOTE

Do not use DAT3 pin as a CD pin.

The module detects the WP (write protect) signal. With the information of the WP state, the register bank ignores the command, accompanied by a write operation, when the WP switch is on.

14.4.5.1 SDIO card interrupt

14.4.5.1.1 Interrupts in 1-bit mode

In this case the DAT[1] pin is dedicated to providing the interrupt function.

An interrupt is asserted by pulling the DAT[1] low from the SDIO card, until the interrupt service is finished to clear the interrupt.

14.4.5.1.2 Interrupt in 4-bit mode

Since the interrupt and DAT[1] share pin 8 in 4-bit mode, an interrupt is sent by the card and recognized by the host only during a specific time.

This is known as the interrupt period. eSDHC only samples the level on pin 8 during the interrupt period. At all other times, the host ignores the level on pin 8, and treats it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in 4-bit mode, there is only a limited period of time that the Interrupt Period can be active due to the limited period of data line availability between the multiple blocks of data. This requires a more strict definition of the Interrupt Period. For this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DAT[1] line is held low for one clock cycle with the last clock cycle pulling DAT[1] high. On completion of the interrupt period, the card releases the DAT[1] line into the high Z state. eSDHC samples DAT[1] during the interrupt period when the PROCTL[IABG] is set.

Refer to SDIO card specification v2.0 for further information about the SDIO card interrupt.

14.4.5.1.3 Card interrupt handling

When the CINTIEN bit in the interrupt signal enable register is set to 0, the eSDHC clears the interrupt request to the host system.

The host driver should clear this bit before servicing the SDIO interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

In 1-bit mode, eSDHC detects the SDIO interrupt with or without the SD clock (to support wakeup). In 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the host system interrupt controller. When the SDIO status is set, and the host driver needs to service this interrupt, so the SDIO bit in the interrupt control register of SDIO card is cleared. This is required to clear the SDIO interrupt status latched in the eSDHC and to stop driving the interrupt signal to the system interrupt controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO interrupt enable bit is set to 1, and the eSDHC starts sampling the interrupt signal again.

The figure below illustrates the SDIO card interrupt scheme and sequences of software and hardware events that take place during a card interrupt handling procedure.

Figure 14-18. Card interrupt scheme and card interrupt detection and handling procedure

14.4.5.2 Card insertion and removal detection

When the DAT[3] pin is not used for card detection (for example, it is implemented in GPIO), the CD pin must be connected for card detection. Whether DAT[3] is configured for card detection or not, the CD pin is always a reference for card detection. Whether the DAT[3] pin or the CD pin is used to detect card insertion, the eSDHC sends an interrupt (if enabled) to inform the host system that a card is inserted.

14.4.5.3 Power management and wake up events

If no operation is expected to happen between eSDHC and the card through the SD bus, the user can completely disable the IP in the chip-level clock control module to save power.

When the user needs to use the eSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to the eSDHC are disabled, for instance, when the system is in low power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The eSDHC can generate these interrupt even when there are no clocks enabled. The three interrupts which can be used as wake up events are:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO card

The eSDHC offers a power management feature. By clearing the clock enabled bits in the system control register (SYSCTL), the clocks are gated in the low position to the eSDHC. For maximum power saving, the user can disable all the clocks to the eSDHC when there is no operation in progress.

These three wake up events (or wakeup interrupts) can also be used to wake up the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to the eSDHC are disabled or when the whole system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol control register (PROCTL) for more information.

14.4.5.3.1 Setting wake up events

For eSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters sleep mode.

Before the software disables the host clock, it should ensure that all of the following conditions are met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

14.5 Initialization/application of eSDHC

All communication between the system and the cards are controlled by the host.

The host sends commands of two types: Broadcast and Addressed (point-to-point).

Broadcast commands are intended for all cards, such as "GO_IDLE_STATE", "SEND_OP_COND", "ALL_SEND_CID" and so on. In Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for /SD/SDIO and eMMC device for the commands of bc and bcr categories.

After the Broadcast command CMD3 is issued, the cards enter standby mode. Addressed type commands are used from this point. In this mode, the CMD/DAT I/O pads will turn to push-pull mode, to have the driving capability for maximum frequency operation. Refer to Commands for /SD/SDIO and eMMC device, for the commands of ac and adtc categories.

14.5.1 Command send and response receive basic operation

Assuming the data type WORD is an unsigned 32-bit integer, the below flow is a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into transfer type register
(XFERTYP),
it is recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) >> 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
          set MSBSEL bit;
          if (finite block number) {
                    set BCEN bit;
                    if (auto12 command is to use) set AC12EN bit;
          }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set transfer type register (XFERTYP) as wCmd value to issue the
command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here using polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. By doing this, make sure the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and go to the Standby State, no response to the Host when CMD2 is sent. The host driver should deal with 'fake' errors like this with caution.

14.5.2 Card identification mode

When a card is inserted to the socket or the card was reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the relative card address (RCA) .For eMMC devices, set the relative card address. All data communications in the card identification mode use the command line (CMD) only.

14.5.2.1 Card detect

The figure below illustrates a flow diagram showing the detection of , SDIO, and SD cards using the eSDHC.

Figure 14-19. Flow diagram for card detection

The card detect sequence is as follows:

1. Set the IRQSTAT[CINSIEN] bit and IRQSIGEN[CINSIEN] bit to enable card detection interrupt.
2. When an interrupt from the eSDHC is received in IRQSTAT[CINS], check PRSSTAT[CINS] to confirm if it was caused by card insertion.

NOTE

This section is not valid for embedded devices.

14.5.2.2 Reset

The host consists of three types of resets:

- Hardware reset (card and host) which is driven by POR (power on reset).

- Software reset (Host Only) is proceed by the write operation on SYSCTL[RSTD], SYSCTL[RSTC], or SYSCTL[RSTA] bits to reset the data part, command part, or all parts of the host controller, respectively.
- Card reset (card only). The command, "Go_Idle_State" (CMD0), is the software reset command for all types of eMMC devices and SD memory cards. This command sets each card into the idle state regardless of the current card state. For an SD I/O Card, CMD52 is used to write an I/O reset in the CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. Figure below illustrates the software flow to reset both the eSDHC and the card.

Figure 14-20. Flow chart for reset of the eSDHC and SD I/O card

```
software_reset()
{
set_bit(SYSCTRL, RSTA); // software reset the Host
set DTOCV and SDCLKFS bit fields to get the SDHC_CLK of frequency around 400 KHz
configure IO pad to set the power voltage of external card to around 3.0V
poll bits CIHB and CDIHB bits of PRSSTAT to wait both bits are cleared
set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0
or send_command(CMD_IO_RW_DIRECT, <other parameters>);
}
```

14.5.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification.

However, the supported minimum and maximum values for Vdd are defined in the operation conditions register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer Vdd conditions. This means, if the host and card have non-common Vdd ranges, the card will not be able to complete the identification cycle, nor will it be able to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for eMMC), SD_Send_Op_Cont (ACMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SD I/O) is used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards which do not match the Vdd range(s) desired by the host. This is accomplished by the host sending the desired Vdd voltage window as the operand of this command. Cards that cannot perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the inactive state. This query should be used if the host is able to select a common voltage range or if a notification should be sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement)
{
label the card as UNKNOWN;

send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
operation voltage, command argument is zero

if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
          if (0 < number of IO functions) {
                    label the card as SDIO;
                    IORDY = 0;
                    while (!(IORDY in IO OCR response)) { // set voltage range for each IO
function
                              send_command(IO_SEND_OP_COND, <voltage range>, <other
parameter>);
                              wait_for_response(IO_SEND_OP_COND);
                    } // end of while ...
          } // end of if (0 < ...
          if (memory part is present inside SDIO card) Label the card as SDCombo; // this is
an
SD-Combo card
} // end of if (RESP_TIMEOUT ...
if (the card is labelled as SDIO card) return; // card type is identified and voltage range
is
set, so exit the function;
send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application specific
CMD
prefix
if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
          send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set voltage
range
for memory part or SD card
          wait_for_response(SD_APP_OP_COND); // voltage range is set
          if (card type is UNKNOWN) label the card as SD;
          return; //
} // end of if (no error ...
else if (errors other than time-out occur) { // command/response pair is corrupted
          deal with it by program specific manner;
} // of else if (response time-out
else { // CMD55 is refuse, it must be eMMC device
          if (card is already labelled as SDCombo) { // change label
                    re-label the card as SDIO;
                    ignore the error or report it;
                    return; // card is identified as SDIO card
          } // of if (card is ...
          send_command(SEND_OP_COND, <voltage range>, <...>);
          if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
```

```
either
                    label the card as UNKNOWN;
                    return;
          } // of if (RESP_TIMEOUT ...
          else label the card as eMMC;
} // of else
}
```

14.5.2.4 Card registry

Card registry for the SD/SDIO/SD combo cards and eMMC devices are different.

For the SD card, the identification process starts at a clock rate lower than 400 KHz and the power voltage higher than 2.7 V (as defined by the card specification). At this time, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host will request the card to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command should be send to all of the new cards in the system. Incompatible cards are put into the inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the ready state), send their CID number as the response. After the CID is sent by the card, the card goes into the identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA will be used to address the card for future data transfer operations. Once the RCA is received, the card changes its state to the standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in system.

For eMMC operation, the host starts the card identification process in open-drain mode with the identification clock rate lower than 400 KHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired OR" operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop

sending their CID immediately and must wait for the next identification cycle. Since the CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into the Identification State. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to the card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react in further identification cycles, and its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

```
card_registry()
{
do { // decide RCA for each card until response time-out
          if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
                    send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to
publish its
RCA
                    retrieve RCA from response;
          } // end if (card is labelled as SDCombo ...
          else if (card is labelled as SD) { // for SD card
                    send_command(ALL_SEND_CID, <...>);
                    if (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
                    send_command(SET_RELATIVE_ADDR, <...>);
                    retrieve RCA from response;
          } // else if (card is labelled as SD ...
          else if (card is labelled as eMMC) {
                    send_command(ALL_SEND_CID, <...>);
                    rca = 0x1; // arbitrarily set RCA, 1 here for example
                    send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper
16
bits
          } // end of else if (card is labelled as eMMC ...
} while (response is not time-out);
}
```

14.5.3 Card access

This section describes access to the card.

14.5.3.1 clocking constraints

Table 14-12. Clocking constraints

Mode name	Frequency	Source clock option	Minimum clock divisor	Maximum clock divisor
Default Speed	0-25 MHz	IPG clock, IPG PER clock	Card clock should not exceed 25 MHz.	No restriction
High Speed	0-50 MHz	IPG clock, IPG PER clock	Card clock should not exceed 50 MHz.	No restriction
SDR12	0-25 MHz	IPG clock, IPG PER clock	Card clock should not exceed 25 MHz.	No restriction
SDR25	0-50 MHz	IPG clock, IPG PER clock	Card clock should not exceed 50 MHz.	No restriction

Table continues on the next page...

Table 14-12. Clocking constraints (continued)

Mode name	Frequency	Source clock option	Minimum clock divisor	Maximum clock divisor
eMMC DDR50	0-52 MHz	IPG clock, IPG PER clock	Card clock should not exceed 52 MHz. Odd clock division in DDR mode is supported only with IPG PER clock.	No restriction. Odd clock division in DDR mode is supported only with IPG PER clock.
DDR50	0-50 MHz	IPG clock, IPG PER clock	Card clock should not exceed 50 MHz. Odd clock division in DDR mode is supported only with IPG PER clock.	No restriction. Odd clock division in DDR mode is supported only with IPG PER clock.
SDR50 (without tuning)	0-100 MHz	IPG clock, IPG PER clock	Card clock should not exceed 100 MHz.	No restriction
SDR50 (with tuning)	0-100 MHz	IPG PER clock	3	16
SDR104	0-208 MHz	IPG PER clock	3	16
HS200	0-200 MHz	IPG PER clock	3	16
HS400	0-200 MHz	IPG PER clock	3	16

NOTE

IPG PER CLK is the clock selected through RCW[HWA_CGA_M2_CLK_SEL] and the IPG clock is platform clock.

14.5.3.2 Block write

14.5.3.2.1 Normal write

During a block write (CMD24 - 27), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host.

If the CRC fails, the card should indicate the failure on the DAT line. The transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write protected area.

For eMMC devices and SD cards, programming of the CID and CSD registers do not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card will respond with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby State and release the DAT line without interrupting the write operation. When re-selecting the card, it will reactivate the busy indication by pulling DAT to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card using DMA mode is as follows:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   a. For SD cards and eMMC devices, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1-7)
3. Set the eSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the eSDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the eSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or some other error that occurred during the auto12 command sending and response receiving.

The software flow to write to a card using CPU Polling mode is as follows:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   a. For SD cards and eMMC devices, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1-7)

3. Set the eSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the eSDHC number block register (NOB), nob is 5 (for instance).
5. Issue the command with data transfer. The AC12EN bit should also be set.
6. Wait for IRQSTAT[BWR] interrupt to be set.
7. Write to Buffer port register for no. of times programmed in WML[WR_WML] considering restriction mentioned in Software polling procedure.
8. Clear IRQSTAT[BWR].
9. Repeat 6-8 steps for rest of the data transfer.
10. Wait for the Transfer Complete interrupt.
11. Check the status bit to see if a write CRC error occurred, or some other error that occurred during the auto12 command sending and response receiving.

14.5.3.2.2 Write with pause

The write operation can be paused during the transfer.

Instead of stopping the SDHC_CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the PROCTL[SABGREQ] to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DAT0 line is not required to de-assert to release the busy state, no suspend command is needed.

Like in the flow described in Normal write, the write with pause is shown with the same kind of write operation:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD cards and eMMC devices, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1-7)
3. Set the eSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the eSDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the eSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.

10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of BLKATTR[BLKCNT]. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The driver should read the value of BLKCNT after the transfer is paused and the transfer complete interrupt is received.

It is also possible the last block has begun when the stop at block gap request is sent to the buffer. In this case, the next block gap is actually the end of the transfer. These types of requests are ignored and the driver should treat this as a non-pause transfer and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the host system is not stopped, and the transfer is active until the data buffer is full. Because of this (if not needed), it is recommended to avoid using the suspend command for the SDIO card. This is because, when such a command is sent, the eSDHC thinks that the system will switch to another function on the SDIO card, and flush the data buffer. The eSDHC takes the resume command as a normal command with data transfer, and it is left for the driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, XFERTYP[MSBSEL] and XFERTYP[BCEN] are set as well as XFERTYP[AC12EN]. However, the eSDHC will automatically send a CMD12 to mark the end of the multi-block transfer.

14.5.3.3 Block read

14.5.3.3.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification.

A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the transfer state. For multi-blocks read, data blocks will be continuously transferred until a stop command is issued.

The software flow to read from a card using DMA mode is as follows:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:

   a. For SD cards and eMMC devices, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1-7)
3. Set the eSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the eSDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer read ready interrupt, configure the DMA settings and enable the eSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or some other error, occurred during the auto12 command sending and response receiving.

The software flow to read from a card using CPU polling mode is as follows:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD cards and eMMC devices, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1-7)
3. Set the eSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the eSDHC number block register (NOB), nob is 5 (for instance).
5. Issue the command with data transfer. The AC12EN bit should also be set.
6. Wait for IRQSTAT[BRR] interrupt to be set.
7. Read from Buffer port register for number of times programmed in WML[RD_WML] considering restriction mentioned in Software polling procedure.
8. Clear IRQSTAT[BRR].
9. Repeat 6-8 steps for rest of the data transfer.
10. Wait for the Transfer Complete interrupt.
11. Check the status bit to see if a read CRC error occurred, or some other error, occurred during the auto12 command sending and response receiving.

14.5.3.3.2 Read with pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the read wait feature can pause during the read operation.

If the SDIO card support read wait (SRW bit in CCCR register is 1), the driver can set the PROCTL[SABGREQ] to pause the transfer between the data blocks. Before setting the SABGREQ bit, make sure the PROCTL[RWCTL] is set, otherwise the eSDHC will not assert the read wait signal during the block gap and data corruption occurs. It is recommended to set the RWCTL bit once the read wait capability of the SDIO card is recognized.

Like in the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm the card supports read wait.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   a. For SD cards and eMMC devices, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1-7)
5. Set the eSDHC block length register to be the same as the block length set for the card in Step 4.
6. Set the eSDHC number block register (NOB), nob is 5 (for instance).
7. Disable the buffer read ready interrupt, configure the DMA setting and enable the eSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
8. Set the SABGREQ bit.
9. Wait for the transfer complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the transfer complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or some other error, occurred during the auto12 command sending and response receiving.

Like the write operation, it is possible to meet the ending block of the transfer when paused. In this case, the eSDHC will ignore the stop at block gap request and treat it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause will be transferred to the Host System. Even if the Suspend Command is sent or not, the internal data buffer is not flushed.

If the Suspend Command is sent and the transfer is later resumed by means of a Resume Command, the eSDHC takes the command as a normal one accompanied with data transfer. It is left for the driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, XFERTYP[MSBSEL] and XFERTYP[BCEN] are set, as well as XFERTYP[AC12EN]. However, the eSDHC will automatically send the CMD12 to mark the end of multi-block transfer.

14.5.3.4 Suspend resume

The eSDHC supports the suspend resume operations of SDIO cards, although slightly different than the suggested implementation of suspend in the SDIO card specification.

14.5.3.4.1 Suspend

After setting the SABGREQ bit, the host driver may send a Suspend command to switch to another function of the SDIO card. The eSDHC does not monitor the content of the response, so it doesn't know if the Suspend command succeeded or not.

Accordingly, it doesn't de-assert Read Wait for read pause. To solve this problem, the driver should not mark the Suspend command as a "Suspend", (that is, setting the CMDTYP bits to 01). Instead, the driver should send this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the driver send another command marked as "Suspend" to inform the eSDHC that the current transfer is suspended. As shown in the following sequence for Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, save the context registers in the system memory for later use, including the SDMA system address register (DSADDR) for DMA operation, and the block attributes register (BLKATTR).
3. Send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
4. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the driver strategy.
5. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so the eSDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, the eSDHC stops driving DAT2 and goes to the idle state.
6. Begin operation for another function on the SDIO card.

14.5.3.4.2 Resume

To resume the data transfer, a resume command should be issued:

1. To resume the suspended function, restore the context register with the saved value in step #2 of the suspend operation above.
2. Send the resume command. In the transfer type register (XFERTYP), all bit fields are set to the value as if this were another ordinary data transfer, instead of a transfer resume (except the CMDTYP is set to 2'b10).
3. If the resume command has responded, the data transfer will be resumed.

14.5.3.5 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command.

To accomplish this:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4 Kbyte address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1-3.
4. Repeat steps 1-3 until all descriptors are created.
5. In the last descriptor, set the end flag to 1 and make sure the total length of all descriptors match the product of the block size and block number configured in the block attributes register (BLKATTR).
6. Set the ADMA system address register (ADSADDR) to the address of the first descriptor and set PROCTL[DMAS] to 01 to select the ADMA.
7. Issue a write or read command with XFERTYP[DMAEN] set.

Steps 1-5 are independent of step 6, so step 6 can finish before steps 1-5. Regarding the descriptor configuration, it is recommended not to use the link descriptor as it requires extra system memory access.

14.5.3.6 Tuning block procedure

Tuning block is used for SD UHS SDR50, SDR104 and eMMC HS200 modes.

Host driver should execute tuning procedure before initiating data transfer with these speed modes.

The steps for using tuning block are:

1. Clear SYSCTL[SDCLKEN]
2. Wait for PRSSTAT[SDSTB] to be set
3. Set ESDHCCTL[FAF]
4. Wait for ESDHCCTL[FAF] to be cleared
5. Set appropriate AUTOCERR[UHSM]
6. Set TBCTL[TB_EN] and program appropriate TBCTL[TB_MODE]
7. Set SYSCTL[SDCLKEN]
8. Wait for PRSSTAT[SDSTB] to be set
9. Execute tuning procedure

While tuning procedure is being performed, eSDHC does not generate any other command or data interrupt except buffer read ready in IRQSTAT register.

When tuning error is received, host driver should abort the current data transfer and execute the tuning procedure.

ESDHCCTL[PCS] (peripheral clock select), and TBCTL[TB_EN] (tuning block enable) should always be set whenever using tuning block. PCS should be set before TB_EN.

NOTE

- Similar steps needs to be performed to disable tuning block, except programming different values in 5 and 6. Also 9 need not to be performed.
- For tuning mode operation, the SD clock divisor value must be within 3 to 16.
- Tuning data timeout occur with default value of SYSCTL[DTOCV] when SD card send tuning data after 450 µs. According to SD host controller specification a card may take up to 150 ms (or at least 100 ms) to respond to the first tuning block.

14.5.3.6.1 Tuning procedure for hardware tuning modes

The steps for tuning procedure, when TBCTL[TB_MODE] is programmed to either Mode 1, Mode 2 or Mode 3, are:

1. Set SYSCTL2[EXTN], execute tuning.
2. Issue SEND_TUNING_BLK Command (CMD19 for SD, CMD21 for eMMC).
3. Wait for IRQSTAT[BRR], buffer read ready, to be set.
4. Clear IRQSTAT[BRR].
5. Check SYSCTL2[EXTN] to be cleared.
6. Repeat steps 2-5, if EXTN is not cleared.

7. Check SYSCTL2[SMPCLKSEL], sampling clock select. It's set value indicates tuning procedure success, and clears indicate failure. In case of tuning failure, fixed sampling scheme could be used by clearing TBCTL[TB_EN], tuning block enable bit.

14.5.3.6.2 Tuning procedure for software tuning mode

The steps for tuning procedure, when TBCTL[TB_MODE] is set to SW tuning mode, are:

1. Program the start and end pointer for the data window in the TPR register.
2. Set SYSCTL2[EXTN] and SYSCTL2[SMPCLKSEL], execute tuning.
3. Issue SEND_TUNING_BLK Command (CMD19 for SD, CMD21 for eMMC).
4. Wait for IRQSTAT[BRR], buffer read ready, to be set.
5. Clear IRQSTAT[BRR].
6. Check SYSCTL2[EXTN] to be cleared.
7. Check SYSCTL2[SMPCLKSEL], sampling clock select. It's set value indicates tuning procedure success, and clears indicate failure. In case of tuning failure, fixed sampling scheme could be used by clearing TBCTL[TB_EN], tuning block enable bit.

14.5.3.7 DDR

The steps for using DDR mode are:

1. Clear SYSCTL[SDCLKEN]
2. Wait for PRSSTAT[SDSTB] to be set
3. Program AUTOCERR[UHSM] to 4
4. If required, change the clock division ratio in SYSCTL register
5. Set SDCLKCTL[CMD_CLK_CTL] and SDCLKCTL[LPBK_CLK_SEL] for DDR mode
6. Set SYSCTL[SDCLKEN]
7. Wait for PRSSTAT[SDSTB] to be set
8. Again clear SYSCTL[SDCLKEN]
9. Wait for PRSSTAT[SDSTB] to be set
10. Set ESDHCCTL[FAF]
11. Wait for ESDHCCTL[FAF] to be cleared
12. Set SYSCTL[SDCLKEN]
13. Wait for PRSSTAT[SDSTB] to be set

NOTE

1. SD clock divisor should be even for DDR mode.
2. Similar steps needs to be performed to disable, except programming different value in 3

14.5.3.8 Transfer error

14.5.3.8.1 CRC transfer error

It is possible at the end of a block transfer, that a write CRC status error or read CRC error occurs.

For this type of error the latest block received should be discarded.

This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one. For a multi-block transfer, the host driver should issue a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, the eSDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 will be sent by the eSDHC. In this case, the driver should re-send or re-obtain the last block with a single block transfer.

14.5.3.8.2 DMA transfer error

During the data transfer with internal single DMA, if the DMA engine encounters some error on the System bus, the DMA operation is aborted and DMA error interrupt is sent to the host system.

When acknowledged by such an interrupt, the driver should calculate the start address of data block in which the error occurs. The start address can be calculated by:

Read the DMA error address register (DMAERRADDRL/H). Taking the block size and the start address intially preogrammed in SDMA Adress register, it is straight forward to obtain the start address of the corrupted block.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and re-start the transfer from the corrupted block to recover from the error.

14.5.3.8.3 ADMA transfer error

There are three kinds of possible ADMA errors; The System transfer, invalid descriptor, and data-length mismatch errors.

Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set. For acknowledging the status, the host driver should recover the error as shown below and re-transfer from the place of interruption.

1. System transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or the next block if no block is corrupted.
2. Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and re-create the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
3. Data-length mismatch error: It is similar to recover from this error. The host driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain, and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

14.5.3.8.4 Auto CMD12 error

After the last block of the multi block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, the eSDHC automatically sends a CMD12 to the card to stop the transfer.

When errors with this command occur, it is recommended to the driver to deal with the situations in the following manner:

1. Auto CMD12 response time-out. It is not certain whether the command is accepted by the card or not. The driver should clear the Auto CMD12 error status bits and re-send the CMD12 until it is accepted by the card.
2. Auto CMD12 response CRC error. Since card responds to the CMD12, the card will abort the transfer. The driver may ignore the error and clear the error status bit.
3. Auto CMD12 conflict error or not sent. The command is not sent, so the driver should send a CMD12 manually.

14.5.3.9 Card interrupt

The external cards can inform the host controller by means of some special signals. For the SDIO card, it can be the low level on the DAT[1] line during some special period.

The eSDHC only monitors the DAT[1] line and supports the SDIO interrupt.

When the SDIO interrupt is captured by the eSDHC, and the Host System is informed by the eSDHC asserting the eSDHC interrupt line, the interrupt service from the host driver is called.

As the interrupt factor is controlled by the external card, the interrupt from the SDIO card must be served before the CINT bit is cleared by written 1. Refer to Card interrupt handling for the card interrupt handling flow.

14.5.4 Switch function

The Switch command is used to enable high speed and change the bus width for eMMC devices.

For SDIO cards, the high speed mode is enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high speed mode is queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For eMMC devices, the high speed mode is queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The 4-bit and 8-bit bus width of the eMMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO cards, it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following flowcharts do not show current capability check, which is recommended in the function switch process.

14.5.4.1 Query, enable and disable SDIO high speed mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
```

```
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

14.5.4.2 Query, enable and disable SD high speed mode

```
enable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFF1 and read 64 bytes of data accompanying the R1 response;
wait data transfer done bit is set;
check if the bit 401 of received 512 bit is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
send CMD6, with argument 0x80FFFFF1 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

14.5.4.3 Query, enable and disable eMMC high speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of eMMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the eMMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of eMMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this eMMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of eMMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report eMMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
```

```
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of eMMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

14.5.4.4 Set eMMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of eMMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the eMMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

14.5.5 ADMA operation

14.5.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB align);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

14.5.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

14.6 Interfacing Card

eSDHC and card IO voltage might be different. External on board voltage translator can be used to interface the card in that case.

Following diagram illustrates direct interfacing of eSDHC with card when voltage level is same at both ends. SD_CLK_SYNC_OUT and SD_CLK_SYNC_IN should be routed as close as possible to card, with minimum skew with respect to SD_CLK.

Figure 14-21. Direct Interfacing of eSDHC with Card

Below figure illustrates interfacing card with eSDHC through voltage translator when the voltages are different. CMD/DATA DIR and SD_VS pins might be required when interfacing with voltage translator that support DDR, and SDR more than 50 MHz modes. eSDHC SD_VS, or GPIO could be used to control SEL pin of the translator.

Figure 14-22. Card Interfacing with eSDHC through Voltage Translator

NOTE

CD, WP and DAT4-7 are not shown in above figures for simplicity.

14.7 Commands for /SD/SDIO and eMMC device

See the table below for the list of commands for the /SD/SDIO cards and eMMC device.

Refer the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr), response from all cards simultaneously
- Addressed (point-to-point) commands (ac), no data transfer on the DAT
- Addressed (point-to-point) data transfer commands (adtc)

The access bits for the EXT_CSD access modes are shown in Table 14-14.

Table 14-13. Commands for /SD/SDIO cards and eMMC device

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all eMMC device and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all eMMC device and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.

Table continues on the next page...

Table 14-13. Commands for /SD/SDIO cards and eMMC device (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD3[1]	ac	[31:16] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 - 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.
CMD8[4]	bcr	[31:12] reserved bits [11:8] supply voltage (VHS) [7:0] check pattern	R7	SEND_IF_COND	Sends SD memory card interface condition, which includes host supply voltage information and asks the card whether card supports voltage. Reserved bits shall be set to 0.
CMD8[5]	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.

Table continues on the next page...

Table 14-13. Commands for /SD/SDIO cards and eMMC device (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD11	ac	[31:0] stuff bits	R1	VOLTAGE_SWITCH	Switch to 1.8V bus signalling level. Applicable for UHS SD card.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	adtc	[31:0] stuff bits	R1	SEND_TUNING_BLOCK	64 bytes tuning pattern is sent for SDR50 and SDR104. Applicable for UHS SD card.
CMD20	Reserved				
CMD21	adtc	[31:0] stuff bits	R1	SEND_TUNING_BLOCK	128 clocks of tuning pattern (64 byte in 4-bit mode or 128 byte in 8-bit mode) is sent for HS200 optimal sampling point detection. Applicable for eMMC device.
CMD22	Reserved				
CMD23	ac	[31:16] set to 0 [15:0] block count	R1	SET_BLOCKCOUNT	Defines the number of blocks which are going to be transferred in the immediately succeeding multiple block read or write command. If the argument is all 0s, the subsequent read/write operations are openended.
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command should be issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

Table continues on the next page...

Table 14-13. Commands for /SD/SDIO cards and eMMC device (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK_START	Sets the address of the first write block to be erased in SD card.
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group in eMMC device.
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK_END	Sets the address of the last write block of the continuous range to be erased in SD card.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase in eMMC device.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection in eMMC device.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase in eMMC device.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase in eMMC device.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection in eMMC device.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the eMMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode in eMMC device.

Table continues on the next page...

Table 14-13. Commands for /SD/SDIO cards and eMMC device (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/ unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43-51	Reserved				
CMD52	ac	As per SDIO spec CMD52 format	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	As per SDIO spec CMD53 format	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-63	Reserved				
ACMD6	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[6]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD18[6]	adtc	[31:0] stuff bits	R1	SECURE_READ_MULTI_BLOCK	Protected Area Access Command: Reads continuously transfer data blocks from protected area of SD memory card. Refer Security Specification Version 2.00 for more details.
ACMD22[6]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[6]	ac	[31:0] stuff bits	R1	SET_WR_BLK_ERASE_COUNT	In SD cards, set the number of write blocks to be pre-erased before writing (to be used for faster multiple block write command) Default value is 1 (one write block).
ACMD25[6]	adtc	[31:0] stuff bits	R1	SECURE_WRITE_MULTI_BLOCK	Protected Area Access Command:

Table continues on the next page...

Table 14-13. Commands for /SD/SDIO cards and eMMC device (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
					Writes continuously transfer data blocks to protected area of SD memory card. Refer Security Specification Version 2.00 for more details.
ACMD26[6]	adtc	[31:0] stuff bits	R1	SECURE_WRITE_MKB	System Area Access Command: Overwrite the existing media key block (MKB) on the system area of SD Memory Card with new MKB. This command is used in dynamic update MKB scheme. Refer Security Specification Version 2.00 for more details.
ACMD38[6]	ac	[31:0] stuff bits	R1b	SECURE_ERASE	Protected Area Access Command: Erase a specified region of the Protected Area of SD Memory Card. Refer Security Specification Version 2.00 for more details.
ACMD41[6]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[6]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect [1]/Disconnect [0] the 50KΩ pull-up resistor on CD/DAT3 of SD card.
ACMD43[6]	adtc	[31:24]Unit_Count: [23:16] MKB_ID: [15:0]Unit_Offset:	R1	GET_MKB	Reads Media Key Block from the System Area of SD Memory Card. -Unit_Count specifies the Number of units to read. (Here, a unit=512 byte (fixed).) - MKB_ID specifies the application unique number. - Unit_Offset specifies the start address(offset) to read. Refer Security Specification Version 2.00 for more details.
ACMD44[6]	adtc	[31:0] stuff bits	R1	GET_MID	Reads Media ID from the system area of SD memory card. Refer Security Specification Version 2.00 for more details.
ACMD45[6]	adtc	[31:0] stuff bits	R1	SET_CER_RN1	AKE Command: Writes random number RN1 as challenge1 in AKE process. Refer Security Specification Version 2.00 for more details.
ACMD46[6]	adtc	[31:0] stuff bits	R1	GET_CER_RN2	AKE Command:

Table continues on the next page...

Table 14-13. Commands for /SD/SDIO cards and eMMC device (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
					Reads random number RN2 as challenge2 in AKE process. Refer security specification version 2.00 for more details.
ACMD47[6]	adtc	[31:0] stuff bits	R1	SET_CER_RES2	AKE Command: Writes RES2 as response2 to RN2 in AKE process Refer Security Specification Version 2.00 for more details.
ACMD48[6]	adtc	[31:0] stuff bits	R1	GET_CER_RES1	AKE Command: Reads RES1 as response1 to RN1 in AKE process. Refer Security Specification Version 2.00 for more details.
ACMD49[6]	ac	[31:0] stuff bits	R1b	CHANGE_SECURE_AREA	Protected Area Access Command: Change size of the protected area. Refer Security Specification Version 2.00 for more details.
ACMD51[6]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD configuration register (SCR).

1. CMD3 differs for eMMC devices and SD cards. For eMMC device, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed eMMC devices and high speed SD cards. Command SWITCH_FUNC is for high speed SD cards.
3. Command SWITCH is for high speed eMMC devices. The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 2.
4. CMD8 for SD stands for SEND_IF_COND.
5. CMD8 for eMMC stands for SEND_EXT_CSD.
6. ACMDs should be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).

Table 14-14. EXT_CSD access modes

Bits	Access name	Operation
00	Command set	The command set is changed according to the Cmd set field of the argument.
01	Set Bits	The bits in the pointed byte are set, according to the 1 bit in the Value field.
10	Clear bits	The bits in the pointed byte are cleared, according to the 1 bit in the Value field.
11	Write Byte	The Value field is written into the pointed byte.

14.8 Software restrictions

This section covers software restrictions.

14.8.1 Software polling procedure

For polling read or write, once the software begins a buffer read or write, it must access the buffer data port register (DATPORT) exactly the number of times as watermark level value set in the watermark level register (WML).

However, if the block size is not same as of the value in watermark level register (read and write respectively), the software must access exactly the remaining number of words at the end of each block. For example, for read operations, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block count is 2, then the access times to data buffer port register for the burst sequence in the whole transfer process must be 4, 4, 2 (for first block); 4, 4, 2 (for second block).

14.8.2 Suspend operation

In order to suspend the data transfer, the software must inform eSDHC that the suspend command is successfully accepted.

To achieve this, after the suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as "01") to inform eSDHC that the transfer is suspended.

If software needs to resume the suspended transfer, it should read the value in BLKCNT register to save the remained number of blocks before sending the normal command marked as suspend, otherwise on sending such "suspend" command, eSDHC will regard the current transfer a aborted and change the BLKCNT register to its original value, instead of keeping the remaining number of blocks.

14.8.3 Data port access

Data port does not support parallel access.

For example, during an external DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or external DMA. Otherwise, the data would be corrupted inside the eSDHC buffer.

14.8.4 Multi-block read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for eMMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or SD security read commands, or whatever multi-block read without abort command at card side; soft reset for data(SYSCTL[RSTD]) is required after the transfer is complete to drive the internal state machine to idle mode.

14.8.5 ADMA address

For ADMA1/ADMA2 operation, the ADMA system address register (ADSADDRL/H) and address defined in the address field of the descriptor table should be 4-byte aligned.

14.8.6 Allowed operations after stop at block gap

Only one of these operations is allowed after stop at block gap event:

- Continue data transfer by setting SYSCTL[CREQ]
- Issue suspend command
- Abort data transfer by issued abort command

No other command can be issued until one of the operations listed above is performed.

14.8.7 SDIO card interrupt during soft reset

The host driver should disable SDIO card interrupts in IRQSTAT[CINT] before issuing soft reset for data or all (generally used for error recovery) in the system control register (SYSCTL), and enable it after error recovery sequence has been completed.

14.8.8 Soft reset for data not allowed when SD clock is disabled

Soft reset for data and CMD (SYSCTL[RSTD]/SYSCTL[RSTC]) should not be issued when SD clock is disabled; that is, when SYSCTL[SDCLKEN] is cleared.

Instead, the host driver may issue soft reset for all (SYSCTL[RSTA]).

14.8.9 Data transfer with Auto CMD12 Enable

When Auto CMD12 is enabled for data transfer, it generates IRQSTAT[TC] for data transfer completion but does not generate TC for Auto CMD12(command with busy).

Host Driver needs make sure that card has reached to “trans“ state before issuing any new data command. CMD13(SEND_STATUS) could be sent to check the card status.

Chapter 15
FlexTimer Module

15.1 FlexTimer as Implemented on the chip

The following table describes the FlexTimer module integration into this chip:

Table 15-1. FlexTimer module integration

Module	Module Base address
FlexTimer Module 1	280_0000
FlexTimer Module 2	281_0000
FlexTimer Module 3	282_0000
FlexTimer Module 4	283_0000

The FlexTimer is a two-to-eight channel 16-bit timer. Each Flextimer instance has one counter. The Flextimers on the chip have no I/O. FTM is available only to support PWM, and there are no external FTM pins. Flextimers can be cascaded through logic outside the Flextimer module such that under software control (controlled by DCFG_FTMCR register fields with the default being not cascaded), rather than four 16-bit timers, software may have 2 32-bit “counters” or one 64-bit “counter”. These are not true 32-bit or 64-bit counters however, because the lower Flextimer(s) is effectively a prescaler for the upper prescaler. Therefore, while the cascaded Flextimer can operate with the maximum range equivalent to that which would be reached with a 32-bit or 64-bit counter, it cannot (for example) generate an interrupt when the count reaches any arbitrary 32-bit or 64-bit value. Rather, it can only generate events when the most significant 16 bit Flextimer reaches a specified value. The cascading is implemented by:

- Counters 2 and 1 can be cascaded with counter 1 holding LSB
- Counters 4 and 3 can be cascaded with counter 3 holding the LSB
- Counters 4 through 1 can be cascaded with counter 1 holding the LSB

Flextimer counters can be synchronized through the Global Timer Base (GTB) functionality of the Flextimer module. Flextimer 1 provides the only source for the Flextimer global time base. The other Flextimer instances can share the time base.

The remainder of this chapter refers to a single FlexTimer module. Notes are included to indicate variations for multiple instantiations.

NOTE

The fixed frequency clock is connected to the RTC pin.

15.2 Introduction

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

15.2.1 FlexTimer philosophy

The FlexTimer is built upon a simple timer, the Timer PWM Module – TPM, used for many years on our HCS08 family of 8-bit microcontrollers. The FlexTimer extends the functionality to meet the demands of motor control, digital lighting solutions, and power conversion, while providing low cost and backwards compatibility with the TPM module.

Several key enhancements are made:

- Signed up counter
- Deadtime insertion hardware
- Enhanced triggering functionality
- Initialization and polarity control

All of the features common with the TPM have fully backwards compatible register assignments. The FlexTimer can also use code on the same core platform without change to perform the same functions.

Motor control and power conversion features have been added through a dedicated set of registers and defaults turn off all new features. The new features, such as hardware deadtime insertion, polarity, and output forcing and masking, greatly reduce loading on the execution software and are usually each controlled by a group of registers.

FlexTimer input triggers can be from comparators, ADC, or other submodules to initiate timer functions automatically. These triggers can be linked in a variety of ways during integration of the sub modules so please note the options available for used FlexTimer configuration.

More than one FlexTimers may be synchronized to provide a larger timer with their counters incrementing in unison, assuming the initialization, the input clocks, the initial and final counting values are the same in each FlexTimer.

All main user access registers are buffered to ease the load on the executing software. A number of trigger options exist to determine which registers are updated with this user defined data.

15.2.2 Features

The FTM features include:

- FTM source clock is selectable.
  - The source clock can be the system clock, the fixed frequency clock, or an external clock
  - Fixed frequency clock is an additional clock input to allow the selection of an on chip clock source other than the system clock
  - Selecting external clock connects FTM clock to a chip level input pin therefore allowing to synchronize the FTM counter with an off chip clock source
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit counter
  - It can be a free-running counter or a counter with initial and final value
  - The counting can be up or up-down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In Input Capture mode:
  - The capture can occur on rising edges, falling edges or both edges
  - An input filter can be selected for some channels

- In Output Compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of match triggers
- Initialization trigger
- Software control of PWM outputs
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- Synchronized loading of write buffered FTM registers
- Write protection for critical registers
- Backwards compatible with TPM
- Testing of input captures for a stuck at zero and one conditions
- Dual edge capture for pulse and period width measurement

15.2.3 Modes of operation

When the chip is in an active Debug mode, the FTM temporarily suspends all counting until the chip returns to normal user operating mode. During Stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During Wait mode, the FTM continues to operate normally. If the FTM does not need to produce a real time reference or provide the interrupt sources needed to wake the chip from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

15.2.4 Block diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–7).

The following figure shows the FTM structure. The central component of the FTM is the 16-bit counter with programmable initial and final values and its counting can be up or up-down.

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

CLKS
FTMEN
QUADEN
no clock selected
(FTM counter disable)
system clock
external clock
phase A
phase B
synchronizer
Quadrature decoder
PS
prescaler
(1, 2, 4, 8, 16, 32, 64 or 128)
QUADEN
CPWMS
CAPTEST
FTM counter
CNTIN
MOD
INITTRIGEN
initialization trigger
TOIE
TOF
timer overflow interrupt
TOFDIR
QUADIR
pair channels 0 - channels 0 and 1
DECAPEN
COMBINE0
CPWMS
MS0B:MS0A
ELS0B:ELS0A
dual edge capture mode logic
CH0IE
CH0F
channel 0 interrupt
CH0TRIG
channel 0 match trigger
channel 0 input
input capture mode logic
C0V
output modes logic
(generation of channels 0 and 1 outputs signals in output compare, EPWM, CPWM and combine modes according to initialization, complementary mode, inverting, software output control, deadtime insertion, output mask, and polarity control)
channel 0 output signal
channel 1 output signal
channel 1 input
input capture mode logic
C1V
DECAPEN
COMBINE0
CPWMS
MS1B:MS1A
ELS1B:ELS1A
CH1F
CH1IE
channel 1 interrupt
CH1TRIG
channel 1 match trigger
pair channels 3 - channels 6 and 7
DECAPEN
COMBINE3
CPWMS
MS6B:MS6A
ELS6B:ELS6A
dual edge capture mode logic
CH6IE
CH6F
channel 6 interrupt
CH6TRIG
channel 6 match trigger
channel 6 input
input capture mode logic
C6V
output modes logic
(generation of channels 6 and 7 outputs signals in output compare, EPWM, CPWM and combine modes according to initialization, complementary mode, inverting, software output control, deadtime insertion, output mask, and polarity control)
channel 6 output signal
channel 7 output signal
channel 7 input
input capture mode logic
C7V
DECAPEN
COMBINE3
CPWMS
MS7B:MS7A
ELS7B:ELS7A
CH7F
CH7IE
channel 7 interrupt
CH7TRIG
channel 7 match trigger

Figure 15-1. FTM block diagram

15.3 FTM signal descriptions

Table 15-2 shows the user-accessible signals for the FTM.

Table 15-2. FTM signal descriptions

Signal	Description	I/O	Function
EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I	The external clock input signal is used as the FTM counter clock if selected by CLKS[1:0] bits in the SC register. This clock signal must not exceed 1/4 of system clock frequency. The FTM counter prescaler selection and settings are also used when an external clock is selected.
CHn	FTM channel (n), where n can be 7-0	I/O	Each FTM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.

15.4 Memory map and register definition

15.4.1 Memory map

This section presents a high-level summary of the FTM registers and how they are mapped.

The registers and bits of an unavailable function in the FTM remain in the memory map and in the reset value, but they have no active function.

Note

Do not write in the region from the CNTIN register through the PWMLOAD register when FTMEN = 0.

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

15.4.2 Register descriptions

Accesses to reserved addresses result in transfer errors. Registers for absent channels are considered reserved.

FTM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
280_0000	Status And Control (FTM1_SC)	32	R/W	0000_0000h	15.4.3/938
280_0004	Counter (FTM1_CNT)	32	R/W	0000_0000h	15.4.4/940
280_0008	Modulo (FTM1_MOD)	32	R/W	0000_0000h	15.4.5/940
280_000C	Channel (n) Status And Control (FTM1_C0SC)	32	R/W	0000_0000h	15.4.6/941
280_0010	Channel (n) Value (FTM1_C0V)	32	R/W	0000_0000h	15.4.7/943
280_0014	Channel (n) Status And Control (FTM1_C1SC)	32	R/W	0000_0000h	15.4.6/941
280_0018	Channel (n) Value (FTM1_C1V)	32	R/W	0000_0000h	15.4.7/943
280_001C	Channel (n) Status And Control (FTM1_C2SC)	32	R/W	0000_0000h	15.4.6/941
280_0020	Channel (n) Value (FTM1_C2V)	32	R/W	0000_0000h	15.4.7/943
280_0024	Channel (n) Status And Control (FTM1_C3SC)	32	R/W	0000_0000h	15.4.6/941
280_0028	Channel (n) Value (FTM1_C3V)	32	R/W	0000_0000h	15.4.7/943
280_002C	Channel (n) Status And Control (FTM1_C4SC)	32	R/W	0000_0000h	15.4.6/941
280_0030	Channel (n) Value (FTM1_C4V)	32	R/W	0000_0000h	15.4.7/943
280_0034	Channel (n) Status And Control (FTM1_C5SC)	32	R/W	0000_0000h	15.4.6/941
280_0038	Channel (n) Value (FTM1_C5V)	32	R/W	0000_0000h	15.4.7/943
280_003C	Channel (n) Status And Control (FTM1_C6SC)	32	R/W	0000_0000h	15.4.6/941
280_0040	Channel (n) Value (FTM1_C6V)	32	R/W	0000_0000h	15.4.7/943
280_0044	Channel (n) Status And Control (FTM1_C7SC)	32	R/W	0000_0000h	15.4.6/941
280_0048	Channel (n) Value (FTM1_C7V)	32	R/W	0000_0000h	15.4.7/943
280_004C	Counter Initial Value (FTM1_CNTIN)	32	R/W	0000_0000h	15.4.8/944
280_0050	Capture And Compare Status (FTM1_STATUS)	32	R/W	0000_0000h	15.4.9/945
280_0054	Features Mode Selection (FTM1_MODE)	32	R/W	0000_0004h	15.4.10/ 947
280_0058	Synchronization (FTM1_SYNC)	32	R/W	0000_0000h	15.4.11/ 948
280_005C	Initial State For Channels Output (FTM1_OUTINIT)	32	R/W	0000_0000h	15.4.12/ 951
280_0060	Output Mask (FTM1_OUTMASK)	32	R/W	0000_0000h	15.4.13/ 952
280_0064	Function For Linked Channels (FTM1_COMBINE)	32	R/W	0000_0000h	15.4.14/ 954
280_0068	Deadtime Insertion Control (FTM1_DEADTIME)	32	R/W	0000_0000h	15.4.15/ 958
280_006C	FTM External Trigger (FTM1_EXTTRIG)	32	R/W	0000_0000h	15.4.16/ 959

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
280_0070	Channels Polarity (FTM1_POL)	32	R/W	0000_0000h	15.4.17/ 961
280_0074	Fault Mode Status (FTM1_FMS)	32	R/W	0000_0000h	15.4.18/ 963
280_0078	Input Capture Filter Control (FTM1_FILTER)	32	R/W	0000_0000h	15.4.19/ 964
280_0084	Configuration (FTM1_CONF)	32	R/W	0000_0000h	15.4.20/ 965
280_008C	Synchronization Configuration (FTM1_SYNCONF)	32	R/W	0000_0000h	15.4.21/ 966
280_0090	FTM Inverting Control (FTM1_INVCTRL)	32	R/W	0000_0000h	15.4.22/ 968
280_0094	FTM Software Output Control (FTM1_SWOCTRL)	32	R/W	0000_0000h	15.4.23/ 969
280_0098	FTM PWM Load (FTM1_PWMLOAD)	32	R/W	0000_0000h	15.4.24/ 971
281_0000	Status And Control (FTM2_SC)	32	R/W	0000_0000h	15.4.3/938
281_0004	Counter (FTM2_CNT)	32	R/W	0000_0000h	15.4.4/940
281_0008	Modulo (FTM2_MOD)	32	R/W	0000_0000h	15.4.5/940
281_000C	Channel (n) Status And Control (FTM2_C0SC)	32	R/W	0000_0000h	15.4.6/941
281_0010	Channel (n) Value (FTM2_C0V)	32	R/W	0000_0000h	15.4.7/943
281_0014	Channel (n) Status And Control (FTM2_C1SC)	32	R/W	0000_0000h	15.4.6/941
281_0018	Channel (n) Value (FTM2_C1V)	32	R/W	0000_0000h	15.4.7/943
281_001C	Channel (n) Status And Control (FTM2_C2SC)	32	R/W	0000_0000h	15.4.6/941
281_0020	Channel (n) Value (FTM2_C2V)	32	R/W	0000_0000h	15.4.7/943
281_0024	Channel (n) Status And Control (FTM2_C3SC)	32	R/W	0000_0000h	15.4.6/941
281_0028	Channel (n) Value (FTM2_C3V)	32	R/W	0000_0000h	15.4.7/943
281_002C	Channel (n) Status And Control (FTM2_C4SC)	32	R/W	0000_0000h	15.4.6/941
281_0030	Channel (n) Value (FTM2_C4V)	32	R/W	0000_0000h	15.4.7/943
281_0034	Channel (n) Status And Control (FTM2_C5SC)	32	R/W	0000_0000h	15.4.6/941
281_0038	Channel (n) Value (FTM2_C5V)	32	R/W	0000_0000h	15.4.7/943
281_003C	Channel (n) Status And Control (FTM2_C6SC)	32	R/W	0000_0000h	15.4.6/941
281_0040	Channel (n) Value (FTM2_C6V)	32	R/W	0000_0000h	15.4.7/943
281_0044	Channel (n) Status And Control (FTM2_C7SC)	32	R/W	0000_0000h	15.4.6/941
281_0048	Channel (n) Value (FTM2_C7V)	32	R/W	0000_0000h	15.4.7/943
281_004C	Counter Initial Value (FTM2_CNTIN)	32	R/W	0000_0000h	15.4.8/944
281_0050	Capture And Compare Status (FTM2_STATUS)	32	R/W	0000_0000h	15.4.9/945
281_0054	Features Mode Selection (FTM2_MODE)	32	R/W	0000_0004h	15.4.10/ 947
281_0058	Synchronization (FTM2_SYNC)	32	R/W	0000_0000h	15.4.11/ 948

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
281_005C	Initial State For Channels Output (FTM2_OUTINIT)	32	R/W	0000_0000h	15.4.12/ 951
281_0060	Output Mask (FTM2_OUTMASK)	32	R/W	0000_0000h	15.4.13/ 952
281_0064	Function For Linked Channels (FTM2_COMBINE)	32	R/W	0000_0000h	15.4.14/ 954
281_0068	Deadtime Insertion Control (FTM2_DEADTIME)	32	R/W	0000_0000h	15.4.15/ 958
281_006C	FTM External Trigger (FTM2_EXTTRIG)	32	R/W	0000_0000h	15.4.16/ 959
281_0070	Channels Polarity (FTM2_POL)	32	R/W	0000_0000h	15.4.17/ 961
281_0074	Fault Mode Status (FTM2_FMS)	32	R/W	0000_0000h	15.4.18/ 963
281_0078	Input Capture Filter Control (FTM2_FILTER)	32	R/W	0000_0000h	15.4.19/ 964
281_0084	Configuration (FTM2_CONF)	32	R/W	0000_0000h	15.4.20/ 965
281_008C	Synchronization Configuration (FTM2_SYNCONF)	32	R/W	0000_0000h	15.4.21/ 966
281_0090	FTM Inverting Control (FTM2_INVCTRL)	32	R/W	0000_0000h	15.4.22/ 968
281_0094	FTM Software Output Control (FTM2_SWOCTRL)	32	R/W	0000_0000h	15.4.23/ 969
281_0098	FTM PWM Load (FTM2_PWMLOAD)	32	R/W	0000_0000h	15.4.24/ 971
282_0000	Status And Control (FTM3_SC)	32	R/W	0000_0000h	15.4.3/938
282_0004	Counter (FTM3_CNT)	32	R/W	0000_0000h	15.4.4/940
282_0008	Modulo (FTM3_MOD)	32	R/W	0000_0000h	15.4.5/940
282_000C	Channel (n) Status And Control (FTM3_C0SC)	32	R/W	0000_0000h	15.4.6/941
282_0010	Channel (n) Value (FTM3_C0V)	32	R/W	0000_0000h	15.4.7/943
282_0014	Channel (n) Status And Control (FTM3_C1SC)	32	R/W	0000_0000h	15.4.6/941
282_0018	Channel (n) Value (FTM3_C1V)	32	R/W	0000_0000h	15.4.7/943
282_001C	Channel (n) Status And Control (FTM3_C2SC)	32	R/W	0000_0000h	15.4.6/941
282_0020	Channel (n) Value (FTM3_C2V)	32	R/W	0000_0000h	15.4.7/943
282_0024	Channel (n) Status And Control (FTM3_C3SC)	32	R/W	0000_0000h	15.4.6/941
282_0028	Channel (n) Value (FTM3_C3V)	32	R/W	0000_0000h	15.4.7/943
282_002C	Channel (n) Status And Control (FTM3_C4SC)	32	R/W	0000_0000h	15.4.6/941
282_0030	Channel (n) Value (FTM3_C4V)	32	R/W	0000_0000h	15.4.7/943
282_0034	Channel (n) Status And Control (FTM3_C5SC)	32	R/W	0000_0000h	15.4.6/941
282_0038	Channel (n) Value (FTM3_C5V)	32	R/W	0000_0000h	15.4.7/943
282_003C	Channel (n) Status And Control (FTM3_C6SC)	32	R/W	0000_0000h	15.4.6/941

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
282_0040	Channel (n) Value (FTM3_C6V)	32	R/W	0000_0000h	15.4.7/943
282_0044	Channel (n) Status And Control (FTM3_C7SC)	32	R/W	0000_0000h	15.4.6/941
282_0048	Channel (n) Value (FTM3_C7V)	32	R/W	0000_0000h	15.4.7/943
282_004C	Counter Initial Value (FTM3_CNTIN)	32	R/W	0000_0000h	15.4.8/944
282_0050	Capture And Compare Status (FTM3_STATUS)	32	R/W	0000_0000h	15.4.9/945
282_0054	Features Mode Selection (FTM3_MODE)	32	R/W	0000_0004h	15.4.10/ 947
282_0058	Synchronization (FTM3_SYNC)	32	R/W	0000_0000h	15.4.11/ 948
282_005C	Initial State For Channels Output (FTM3_OUTINIT)	32	R/W	0000_0000h	15.4.12/ 951
282_0060	Output Mask (FTM3_OUTMASK)	32	R/W	0000_0000h	15.4.13/ 952
282_0064	Function For Linked Channels (FTM3_COMBINE)	32	R/W	0000_0000h	15.4.14/ 954
282_0068	Deadtime Insertion Control (FTM3_DEADTIME)	32	R/W	0000_0000h	15.4.15/ 958
282_006C	FTM External Trigger (FTM3_EXTTRIG)	32	R/W	0000_0000h	15.4.16/ 959
282_0070	Channels Polarity (FTM3_POL)	32	R/W	0000_0000h	15.4.17/ 961
282_0074	Fault Mode Status (FTM3_FMS)	32	R/W	0000_0000h	15.4.18/ 963
282_0078	Input Capture Filter Control (FTM3_FILTER)	32	R/W	0000_0000h	15.4.19/ 964
282_0084	Configuration (FTM3_CONF)	32	R/W	0000_0000h	15.4.20/ 965
282_008C	Synchronization Configuration (FTM3_SYNCONF)	32	R/W	0000_0000h	15.4.21/ 966
282_0090	FTM Inverting Control (FTM3_INVCTRL)	32	R/W	0000_0000h	15.4.22/ 968
282_0094	FTM Software Output Control (FTM3_SWOCTRL)	32	R/W	0000_0000h	15.4.23/ 969
282_0098	FTM PWM Load (FTM3_PWMLOAD)	32	R/W	0000_0000h	15.4.24/ 971
283_0000	Status And Control (FTM4_SC)	32	R/W	0000_0000h	15.4.3/938
283_0004	Counter (FTM4_CNT)	32	R/W	0000_0000h	15.4.4/940
283_0008	Modulo (FTM4_MOD)	32	R/W	0000_0000h	15.4.5/940
283_000C	Channel (n) Status And Control (FTM4_C0SC)	32	R/W	0000_0000h	15.4.6/941
283_0010	Channel (n) Value (FTM4_C0V)	32	R/W	0000_0000h	15.4.7/943
283_0014	Channel (n) Status And Control (FTM4_C1SC)	32	R/W	0000_0000h	15.4.6/941
283_0018	Channel (n) Value (FTM4_C1V)	32	R/W	0000_0000h	15.4.7/943

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
283_001C	Channel (n) Status And Control (FTM4_C2SC)	32	R/W	0000_0000h	15.4.6/941
283_0020	Channel (n) Value (FTM4_C2V)	32	R/W	0000_0000h	15.4.7/943
283_0024	Channel (n) Status And Control (FTM4_C3SC)	32	R/W	0000_0000h	15.4.6/941
283_0028	Channel (n) Value (FTM4_C3V)	32	R/W	0000_0000h	15.4.7/943
283_002C	Channel (n) Status And Control (FTM4_C4SC)	32	R/W	0000_0000h	15.4.6/941
283_0030	Channel (n) Value (FTM4_C4V)	32	R/W	0000_0000h	15.4.7/943
283_0034	Channel (n) Status And Control (FTM4_C5SC)	32	R/W	0000_0000h	15.4.6/941
283_0038	Channel (n) Value (FTM4_C5V)	32	R/W	0000_0000h	15.4.7/943
283_003C	Channel (n) Status And Control (FTM4_C6SC)	32	R/W	0000_0000h	15.4.6/941
283_0040	Channel (n) Value (FTM4_C6V)	32	R/W	0000_0000h	15.4.7/943
283_0044	Channel (n) Status And Control (FTM4_C7SC)	32	R/W	0000_0000h	15.4.6/941
283_0048	Channel (n) Value (FTM4_C7V)	32	R/W	0000_0000h	15.4.7/943
283_004C	Counter Initial Value (FTM4_CNTIN)	32	R/W	0000_0000h	15.4.8/944
283_0050	Capture And Compare Status (FTM4_STATUS)	32	R/W	0000_0000h	15.4.9/945
283_0054	Features Mode Selection (FTM4_MODE)	32	R/W	0000_0004h	15.4.10/ 947
283_0058	Synchronization (FTM4_SYNC)	32	R/W	0000_0000h	15.4.11/ 948
283_005C	Initial State For Channels Output (FTM4_OUTINIT)	32	R/W	0000_0000h	15.4.12/ 951
283_0060	Output Mask (FTM4_OUTMASK)	32	R/W	0000_0000h	15.4.13/ 952
283_0064	Function For Linked Channels (FTM4_COMBINE)	32	R/W	0000_0000h	15.4.14/ 954
283_0068	Deadtime Insertion Control (FTM4_DEADTIME)	32	R/W	0000_0000h	15.4.15/ 958
283_006C	FTM External Trigger (FTM4_EXTTRIG)	32	R/W	0000_0000h	15.4.16/ 959
283_0070	Channels Polarity (FTM4_POL)	32	R/W	0000_0000h	15.4.17/ 961
283_0074	Fault Mode Status (FTM4_FMS)	32	R/W	0000_0000h	15.4.18/ 963
283_0078	Input Capture Filter Control (FTM4_FILTER)	32	R/W	0000_0000h	15.4.19/ 964
283_0084	Configuration (FTM4_CONF)	32	R/W	0000_0000h	15.4.20/ 965
283_008C	Synchronization Configuration (FTM4_SYNCONF)	32	R/W	0000_0000h	15.4.21/ 966
283_0090	FTM Inverting Control (FTM4_INVCTRL)	32	R/W	0000_0000h	15.4.22/ 968
283_0094	FTM Software Output Control (FTM4_SWOCTRL)	32	R/W	0000_0000h	15.4.23/ 969

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
283_0098	FTM PWM Load (FTM4_PWMLOAD)	32	R/W	0000_0000h	15.4.24/ 971

15.4.3 Status And Control (FTMx_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, FTM configuration, clock source, and prescaler factor. These controls relate to all channels within this module.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TOF	TOIE	CPWMS	CLKS		PS		
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTMx_SC field descriptions (continued)

Field	Description
7 TOF	Timer Overflow Flag Set by hardware when the FTM counter passes the value in the MOD register. The TOF bit is cleared by reading the SC register while TOF is set and then writing a 0 to TOF bit. Writing a 1 to TOF has no effect. If another FTM overflow occurs between the read and write operations, the write operation has no effect; therefore, TOF remains set indicating an overflow has occurred. In this case, a TOF interrupt request is not lost due to the clearing sequence for a previous TOF. 0 FTM counter has not overflowed. 1 FTM counter has overflowed.
6 TOIE	Timer Overflow Interrupt Enable Enables FTM overflow interrupts. 0 Disable TOF interrupts. Use software polling. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the FTM to operate in Up-Down Counting mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 FTM counter operates in Up Counting mode. 1 FTM counter operates in Up-Down Counting mode.
4–3 CLKS	Clock Source Selection Selects FTM counter clock sources. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 No clock selected. This in effect disables the FTM counter. 01 System clock 10 Fixed frequency clock 11 External clock
PS	Prescale Factor Selection Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 000 Divide by 1 001 Divide by 2 010 Divide by 4 011 Divide by 8 100 Divide by 16 101 Divide by 32 110 Divide by 64 111 Divide by 128

15.4.4 Counter (FTMx_CNT)

The CNT register contains the FTM counter value.

Reset clears the CNT register. Writing any value to COUNT updates the counter with its initial value, CNTIN.

When Debug is active, the FTM counter is frozen. This is the value that you may read.

Address: Base address + 4h offset

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

FTMx_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Counter Value

15.4.5 Modulo (FTMx_MOD)

The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock, and the next value of FTM counter depends on the selected counting method; see Counter.

Writing to the MOD register latches the value into a buffer. The MOD register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the SC register whether Debug is active or not.

Initialize the FTM counter, by writing to CNT, before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Address: Base address + 8h offset

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

FTMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved.
MOD	Modulo Value

15.4.6 Channel (n) Status And Control (FTM*x*_C*n*SC)

CnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function.

Table 15-3. Mode, edge, and level selection

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	X	X	XX	00	Pin not used for FTM—revert the channel pin to general purpose I/O or other peripheral control	
0	0	0	00	01	Input Capture	Capture on Rising Edge Only
				10		Capture on Falling Edge Only
				11		Capture on Rising or Falling Edge
			01	01	Output Compare	Toggle Output on match
				10		Clear Output on match
				11		Set Output on match
			1X	10	Edge-Aligned PWM	High-true pulses (clear Output on match)
				X1		Low-true pulses (set Output on match)
		1	XX	10	Center-Aligned PWM	High-true pulses (clear Output on match-up)
				X1		Low-true pulses (set Output on match-up)

Table continues on the next page...

Table 15-3. Mode, edge, and level selection (continued)

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
	1	0	XX	10	Combine PWM	High-true pulses (set on channel (n) match, and clear on channel (n+1) match)
				X1		Low-true pulses (clear on channel (n) match, and set on channel (n +1) match)
1	0	0	X0	See the following table (Table 15-4).	Dual Edge Capture	One-Shot Capture mode
			X1			Continuous Capture mode

Table 15-4. Dual Edge Capture mode — edge polarity selection

ELSnB	ELSnA	Channel Port Enable	Detected Edges
0	0	Disabled	No edge
0	1	Enabled	Rising edge
1	0	Enabled	Falling edge
1	1	Enabled	Rising and falling edges

Address: Base address + Ch offset + (8d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	0
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_C*n*SC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CHF	Channel Flag Set by hardware when an event occurs on the channel. CHF is cleared by reading the CSC register while CHnF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect.

Table continues on the next page...

FTMx_C*n*SC field descriptions (continued)

Field	Description
	If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF. 0 No channel event has occurred. 1 A channel event has occurred.
6 CHIE	Channel Interrupt Enable Enables channel interrupts. 0 Disable channel interrupts. Use software polling. 1 Enable channel interrupts.
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 15-3 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
4 MSA	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 15-3 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
3 ELSB	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 15-3 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
2 ELSA	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 15-3 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

15.4.7 Channel (n) Value (FTMx_C*n*V)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.

In Input Capture, Capture Test, and Dual Edge Capture modes, any write to a CnV register is ignored.

In output modes, writing to a CnV register latches the value into a buffer. A CnV register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the CnSC register whether Debug mode is active or not.

Address: Base address + 10h offset + (8d × i), where i=0d to 7d

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

FTMx_C*n*V field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
VAL	Channel Value Captured FTM counter value of the input modes or the match value for the output modes

15.4.8 Counter Initial Value (FTMx_CNTIN)

The Counter Initial Value register contains the initial value for the FTM counter.

Writing to the CNTIN register latches the value into a buffer. The CNTIN register is updated with the value of its write buffer according to Registers updated from write buffers.

When the FTM clock is initially selected, by writing a non-zero value to the CLKS bits, the FTM counter starts with the value 0x0000. To avoid this behavior, before the first write to select the FTM clock, write the new value to the the CNTIN register and then initialize the FTM counter by writing any value to the CNT register.

Address: Base address + 4Ch offset

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

FTMx_CNTIN field descriptions

Field	Description
31–16 Reserved	This field is reserved.
INIT	Initial Value Of The FTM Counter

15.4.9 Capture And Compare Status (FTMx_STATUS)

The STATUS register contains a copy of the status flag CHnF bit in CnSC for each FTM channel for software convenience.

Each CHnF bit in STATUS is a mirror of CHnF bit in CnSC. All CHnF bits can be checked using only one read of STATUS. All CHnF bits can be cleared by reading STATUS followed by writing 0x00 to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. CHnF is cleared by reading STATUS while CHnF is set and then writing a 0 to the CHnF bit. Writing a 1 to CHnF has no effect.

If another event occurs between the read and write operations, the write operation has no effect; therefore, CHnF remains set indicating an event has occurred. In this case, a CHnF interrupt request is not lost due to the clearing sequence for a previous CHnF.

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W									0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_STATUS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTMx_STATUS field descriptions (continued)

Field	Description
7 CH7F	Channel 7 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
6 CH6F	Channel 6 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
5 CH5F	Channel 5 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
4 CH4F	Channel 4 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
3 CH3F	Channel 3 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
2 CH2F	Channel 2 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
1 CH1F	Channel 1 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

15.4.10 Features Mode Selection (FTMx_MODE)

This register contains the global enable bit for FTM-specific features and the control bits used to configure:

- Capture Test mode
- PWM synchronization
- Write protection
- Channel output initialization

These controls relate to all channels within this module.

Address: Base address + 54h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			CAPTEST	PWMSYNC	WPDIS	INIT	FTMEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

FTMx_MODE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 CAPTEST	Capture Test Mode Enable Enables the capture test mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Capture test mode is disabled. 1 Capture test mode is enabled.
3 PWMSYNC	PWM Synchronization Mode Selects which triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. See PWM synchronization. The PWMSYNC bit configures the synchronization when SYNCMODE is 0.

Table continues on the next page...

FTMx_MODE field descriptions (continued)

Field	Description
	0 No restrictions. Software and hardware triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. 1 Software trigger can only be used by MOD and CnV synchronization, and hardware triggers can only be used by OUTMASK and FTM counter synchronization.
2 WPDIS	Write Protection Disable When write protection is enabled (WPDIS = 0), write protected bits cannot be written. When write protection is disabled (WPDIS = 1), write protected bits can be written. The WPDIS bit is the negation of the WPEN bit. WPDIS is cleared when 1 is written to WPEN. WPDIS is set when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPDIS has no effect. 0 Write protection is enabled. 1 Write protection is disabled.
1 INIT	Initialize The Channels Output When a 1 is written to INIT bit the channels output is initialized according to the state of their corresponding bit in the OUTINIT register. Writing a 0 to INIT bit has no effect. The INIT bit is always read as 0.
0 FTMEN	FTM Enable This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 TPM compatibility. Free running counter and synchronization compatible with TPM. 1 Free running counter and synchronization are different from TPM behavior.

15.4.11 Synchronization (FTMx_SYNC)

This register configures the PWM synchronization.

A synchronization event can perform the synchronized update of MOD, CV, and OUTMASK registers with the value of their write buffer and the FTM counter initialization.

NOTE

The software trigger, SWSYNC bit, and hardware triggers TRIG0, TRIG1, and TRIG2 bits have a potential conflict if used together when SYNCMODE = 0. Use only hardware or software triggers but not both at the same time, otherwise unpredictable behavior is likely to happen.

The selection of the loading point, CNTMAX and CNTMIN bits, is intended to provide the update of MOD, CNTIN, and CnV registers across all enabled channels simultaneously. The

use of the loading point selection together with SYNCMODE = 0 and hardware trigger selection, TRIG0, TRIG1, or TRIG2 bits, is likely to result in unpredictable behavior.

The synchronization event selection also depends on the PWMSYNC (MODE register) and SYNCMODE (SYNCONF register) bits. See PWM synchronization.

Address: Base address + 58h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SWSYNC	TRIG2	TRIG1	TRIG0	SYNCHOM	REINIT	CNTMAX	CNTMIN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SYNC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 SWSYNC	PWM Synchronization Software Trigger Selects the software trigger as the PWM synchronization trigger. The software trigger happens when a 1 is written to SWSYNC bit. 0 Software trigger is not selected. 1 Software trigger is selected.
6 TRIG2	PWM Synchronization Hardware Trigger 2 Enables hardware trigger 2 to the PWM synchronization. Hardware trigger 2 happens when a rising edge is detected at the trigger 2 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
5 TRIG1	PWM Synchronization Hardware Trigger 1 Enables hardware trigger 1 to the PWM synchronization. Hardware trigger 1 happens when a rising edge is detected at the trigger 1 input signal. 0 Trigger is disabled. 1 Trigger is enabled.

Table continues on the next page...

FTMx_SYNC field descriptions (continued)

Field	Description
4 TRIG0	PWM Synchronization Hardware Trigger 0 Enables hardware trigger 0 to the PWM synchronization. Hardware trigger 0 occurs when a rising edge is detected at the trigger 0 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
3 SYNCHOM	Output Mask Synchronization Selects when the OUTMASK register is updated with the value of its buffer. 0 OUTMASK register is updated with the value of its buffer in all rising edges of the system clock. 1 OUTMASK register is updated with the value of its buffer only by the PWM synchronization.
2 REINIT	FTM Counter Reinitialization By Synchronization (FTM counter synchronization) Determines if the FTM counter is reinitialized when the selected trigger for the synchronization is detected. The REINIT bit configures the synchronization when SYNCMODE is zero. 0 FTM counter continues to count normally. 1 FTM counter is updated with its initial value when the selected trigger is detected.
1 CNTMAX	Maximum Loading Point Enable Selects the maximum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMAX is 1, the selected loading point is when the FTM counter reaches its maximum value (MOD register). 0 The maximum loading point is disabled. 1 The maximum loading point is enabled.
0 CNTMIN	Minimum Loading Point Enable Selects the minimum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMIN is one, the selected loading point is when the FTM counter reaches its minimum value (CNTIN register). 0 The minimum loading point is disabled. 1 The minimum loading point is enabled.

15.4.12 Initial State For Channels Output (FTMx_OUTINIT)

Address: Base address + 5Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7OI	CH6OI	CH5OI	CH4OI	CH3OI	CH2OI	CH1OI	CH0OI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTINIT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OI	Channel 7 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
6 CH6OI	Channel 6 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
5 CH5OI	Channel 5 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
4 CH4OI	Channel 4 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
3 CH3OI	Channel 3 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs.

Table continues on the next page...

FTMx_OUTINIT field descriptions (continued)

Field	Description
	0 The initialization value is 0. 1 The initialization value is 1.
2 CH2OI	Channel 2 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
1 CH1OI	Channel 1 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
0 CH0OI	Channel 0 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.

15.4.13 Output Mask (FTMx_OUTMASK)

This register provides a mask for each FTM channel. The mask of a channel determines if its output responds, that is, it is masked or not, when a match occurs. This feature is used for BLDC control where the PWM signal is presented to an electric motor at specific times to provide electronic commutation.

Any write to the OUTMASK register, stores the value in its write buffer. The register is updated with the value of its write buffer according to PWM synchronization.

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7OM	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM	CH0OM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTMASK field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OM	Channel 7 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
6 CH6OM	Channel 6 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
5 CH5OM	Channel 5 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
4 CH4OM	Channel 4 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
3 CH3OM	Channel 3 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
2 CH2OM	Channel 2 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
1 CH1OM	Channel 1 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
0 CH0OM	Channel 0 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.

15.4.14 Function For Linked Channels (FTMx_COMBINE)

This register contains the control bits used to configure the synchronization, deadtime insertion, Dual Edge Capture mode, Complementary, and Combine mode for each pair of channels (n) and (n+1), where n equals 0, 2, 4, and 6.

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	SYNCEN3	DTEN3	DECAP3	DECAPEN3	COMP3	COMBINE3	0	0	SYNCEN2	DTEN2	DECAP2	DECAPEN2	COMP2	COMBINE2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SYNCEN1	DTEN1	DECAP1	DECAPEN1	COMP1	COMBINE1	0	0	SYNCEN0	DTEN0	DECAP0	DECAPEN0	COMP0	COMBINE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_COMBINE field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 SYNCEN3	Synchronization Enable For n = 6 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
28 DTEN3	Deadtime Enable For n = 6 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
27 DECAP3	Dual Edge Capture Mode Captures For n = 6 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
26 DECAPEN3	Dual Edge Capture Mode Enable For n = 6 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 15-3. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
25 COMP3	Complement Of Channel (n) for n = 6 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
24 COMBINE3	Combine Channels For n = 6 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 SYNCEN2	Synchronization Enable For n = 4 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
20 DTEN2	Deadtime Enable For n = 4 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
19 DECAP2	Dual Edge Capture Mode Captures For n = 4 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
18 DECAPEN2	Dual Edge Capture Mode Enable For n = 4 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 15-3. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
17 COMP2	Complement Of Channel (n) For n = 4 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
16 COMBINE2	Combine Channels For n = 4 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SYNCEN1	Synchronization Enable For n = 2 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
12 DTEN1	Deadtime Enable For n = 2 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
11 DECAP1	Dual Edge Capture Mode Captures For n = 2 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if Dual Edge Capture – One-Shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
10 DECAPEN1	Dual Edge Capture Mode Enable For n = 2 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 15-3. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
9 COMP1	Complement Of Channel (n) For n = 2 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
8 COMBINE1	Combine Channels For n = 2 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SYNCEN0	Synchronization Enable For n = 0 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
4 DTEN0	Deadtime Enable For n = 0 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
3 DECAP0	Dual Edge Capture Mode Captures For n = 0 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
2 DECAPEN0	Dual Edge Capture Mode Enable For n = 0 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 15-3. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
1 COMP0	Complement Of Channel (n) For n = 0 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
0 COMBINE0	Combine Channels For n = 0 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.

15.4.15 Deadtime Insertion Control (FTMx_DEADTIME)

This register selects the deadtime prescaler factor and deadtime value. All FTM channels use this clock prescaler and this deadtime value for the deadtime insertion.

Address: Base address + 68h offset

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

FTMx_DEADTIME field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTMx_DEADTIME field descriptions (continued)

Field	Description
7–6 DTPS	Deadtime Prescaler Value Selects the division factor of the system clock. This prescaled clock is used by the deadtime counter. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0x Divide the system clock by 1. 10 Divide the system clock by 4. 11 Divide the system clock by 16.
DTVAL	Deadtime Value Selects the deadtime insertion value for the deadtime counter. The deadtime counter is clocked by a scaled version of the system clock. See the description of DTPS. Deadtime insert value = (DTPS × DTVAL). DTVAL selects the number of deadtime counts inserted as follows: When DTVAL is 0, no counts are inserted. When DTVAL is 1, 1 count is inserted. When DTVAL is 2, 2 counts are inserted. This pattern continues up to a possible 63 counts. This field is write protected. It can be written only when MODE[WPDIS] = 1.

15.4.16 FTM External Trigger (FTMx_EXTTRIG)

This register:

- Indicates when a channel trigger was generated
- Enables the generation of a trigger when the FTM counter is equal to its initial value
- Selects which channels are used in the generation of the channel triggers

Several channels can be selected to generate multiple triggers in one PWM period. See Channel trigger output and Initialization trigger.

Channels 6 and 7 are not used to generate channel triggers.

Address: Base address + 6Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TRIGF	INITTRIGEN	CH1TRIG	CH0TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 TRIGF	Channel Trigger Flag Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.

Table continues on the next page...

FTMx_EXTTRIG field descriptions (continued)

Field	Description
5 CH1TRIG	Channel 1 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
2 CH4TRIG	Channel 4 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
1 CH3TRIG	Channel 3 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
0 CH2TRIG	Channel 2 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

15.4.17 Channels Polarity (FTMx_POL)

This register defines the output polarity of the FTM channels.

NOTE

The safe value of a channel is the value of its POL bit.

Address: Base address + 70h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_POL field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 POL7	Channel 7 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
6 POL6	Channel 6 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
5 POL5	Channel 5 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
4 POL4	Channel 4 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
3 POL3	Channel 3 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
2 POL2	Channel 2 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.

Table continues on the next page...

FTMx_POL field descriptions (continued)

Field	Description
1 POL1	Channel 1 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
0 POL0	Channel 0 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.

15.4.18 Fault Mode Status (FTMx_FMS)

This register contains the write protection enable bit.

Address: Base address + 74h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									WPEN	0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FMS field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 WPEN	Write Protection Enable The WPEN bit is the negation of the WPDIS bit. WPEN is set when 1 is written to it. WPEN is cleared when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPEN has no effect. 0 Write protection is disabled. Write protected bits can be written. 1 Write protection is enabled. Write protected bits cannot be written.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

15.4.19 Input Capture Filter Control (FTMx_FILTER)

This register selects the filter value for the inputs of channels.

Channels 4, 5, 6 and 7 do not have an input filter.

NOTE

Writing to the FILTER register has immediate effect and must be done only when the channels 0, 1, 2, and 3 are not in input modes. Failure to do this could result in a missing valid signal.

Address: Base address + 78h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FILTER field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–12 CH3FVAL	Channel 3 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
11–8 CH2FVAL	Channel 2 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
7–4 CH1FVAL	Channel 1 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
CH0FVAL	Channel 0 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

15.4.20 Configuration (FTMx_CONF)

This register selects the number of times that the FTM counter overflow should occur before the TOF bit to be set, the FTM behavior in Debug modes, the use of an external global time base, and the global time base signal generation.

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					GTBEOUT	GTBEEN	0	BDMMODE		0	NUMTOF				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CONF field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 GTBEOUT	Global Time Base Output Enables the global time base signal generation to other FTMs. 0 A global time base signal generation is disabled. 1 A global time base signal generation is enabled.
9 GTBEEN	Global Time Base Enable Configures the FTM to use an external global time base signal that is generated by another FTM. 0 Use of an external global time base is disabled. 1 Use of an external global time base is enabled.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 BDMMODE	Debug Mode Selects the FTM behavior in Debug mode. See Debug mode.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
NUMTOF	TOF Frequency

Table continues on the next page...

FTMx_CONF field descriptions (continued)

Field	Description
	Selects the ratio between the number of counter overflows to the number of times the TOF bit is set. NUMTOF = 0: The TOF bit is set for each counter overflow. NUMTOF = 1: The TOF bit is set for the first counter overflow but not for the next overflow. NUMTOF = 2: The TOF bit is set for the first counter overflow but not for the next 2 overflows. NUMTOF = 3: The TOF bit is set for the first counter overflow but not for the next 3 overflows. This pattern continues up to a maximum of 31.

15.4.21 Synchronization Configuration (FTMx_SYNCONF)

This register selects the PWM synchronization configuration, SWOCTRL, INVCTRL and CNTIN registers synchronization, if FTM clears the TRIGj bit, where j = 0, 1, 2, when the hardware trigger j is detected.

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											HWSOC	HWINVC	HWOM	HWWRBUF	HWRSTCNT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SWSOC	SWINVC	SWOM	SWWRBUF	SWRSTCNT	SYNCMODE	0	SWOC	INVC	0	CNTINC	0	HWTRIGMODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SYNCONF field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 HWSOC	Software output control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the SWOCTRL register synchronization. 1 A hardware trigger activates the SWOCTRL register synchronization.
19 HWINVC	Inverting control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the INVCTRL register synchronization. 1 A hardware trigger activates the INVCTRL register synchronization.

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
18 HWOM	Output mask synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the OUTMASK register synchronization. 1 A hardware trigger activates the OUTMASK register synchronization.
17 HWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 A hardware trigger activates MOD, CNTIN, and CV registers synchronization.
16 HWRSTCNT	FTM counter synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the FTM counter synchronization. 1 A hardware trigger activates the FTM counter synchronization.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 SWSOC	Software output control synchronization is activated by the software trigger. 0 The software trigger does not activate the SWOCTRL register synchronization. 1 The software trigger activates the SWOCTRL register synchronization.
11 SWINVC	Inverting control synchronization is activated by the software trigger. 0 The software trigger does not activate the INVCTRL register synchronization. 1 The software trigger activates the INVCTRL register synchronization.
10 SWOM	Output mask synchronization is activated by the software trigger. 0 The software trigger does not activate the OUTMASK register synchronization. 1 The software trigger activates the OUTMASK register synchronization.
9 SWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by the software trigger. 0 The software trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 The software trigger activates MOD, CNTIN, and CV registers synchronization.
8 SWRSTCNT	FTM counter synchronization is activated by the software trigger. 0 The software trigger does not activate the FTM counter synchronization. 1 The software trigger activates the FTM counter synchronization.
7 SYNCMODE	Synchronization Mode Selects the PWM Synchronization mode. 0 Legacy PWM synchronization is selected. 1 Enhanced PWM synchronization is selected.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SWOC	SWOCTRL Register Synchronization 0 SWOCTRL register is updated with its buffer value at all rising edges of system clock. 1 SWOCTRL register is updated with its buffer value by the PWM synchronization.
4 INVC	INVCTRL Register Synchronization

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
	0 INVCTRL register is updated with its buffer value at all rising edges of system clock. 1 INVCTRL register is updated with its buffer value by the PWM synchronization.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CNTINC	CNTIN Register Synchronization 0 CNTIN register is updated with its buffer value at all rising edges of system clock. 1 CNTIN register is updated with its buffer value by the PWM synchronization.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 HWTRIGMODE	Hardware Trigger Mode 0 FTM clears the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2. 1 FTM does not clear the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2.

15.4.22 FTM Inverting Control (FTMx_INVCTRL)

This register controls when the channel (n) output becomes the channel (n+1) output, and channel (n+1) output becomes the channel (n) output. Each INVmEN bit enables the inverting operation for the corresponding pair channels m.

This register has a write buffer. The INVmEN bit is updated by the INVCTRL register synchronization.

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												INV3EN	INV2EN	INV1EN	INV0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_INVCTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTMx_INVCTRL field descriptions (continued)

Field	Description
3 INV3EN	Pair Channels 3 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
2 INV2EN	Pair Channels 2 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
1 INV1EN	Pair Channels 1 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
0 INV0EN	Pair Channels 0 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

15.4.23 FTM Software Output Control (FTMx_SWOCTRL)

This register enables software control of channel (n) output and defines the value forced to the channel (n) output:

- The CHnOC bits enable the control of the corresponding channel (n) output by software.
- The CHnOCV bits select the value that is forced at the corresponding channel (n) output.

This register has a write buffer. The fields are updated by the SWOCTRL register synchronization.

Address: Base address + 94h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7OCV	CH6OCV	CH5OCV	CH4OCV	CH3OCV	CH2OCV	CH1OCV	CH0OCV	CH7OC	CH6OC	CH5OC	CH4OC	CH3OC	CH2OC	CH1OC	CH0OC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SWOCTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 CH7OCV	Channel 7 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
14 CH6OCV	Channel 6 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
13 CH5OCV	Channel 5 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
12 CH4OCV	Channel 4 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
11 CH3OCV	Channel 3 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
10 CH2OCV	Channel 2 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
9 CH1OCV	Channel 1 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
8 CH0OCV	Channel 0 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
7 CH7OC	Channel 7 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
6 CH6OC	Channel 6 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
5 CH5OC	Channel 5 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
4 CH4OC	Channel 4 Software Output Control Enable

Table continues on the next page...

FTMx_SWOCTRL field descriptions (continued)

Field	Description
	0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
3 CH3OC	Channel 3 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
2 CH2OC	Channel 2 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
1 CH1OC	Channel 1 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
0 CH0OC	Channel 0 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.

15.4.24 FTM PWM Load (FTMx_PWMLOAD)

Enables the loading of the MOD, CNTIN, C(n)V, and C(n+1)V registers with the values of their write buffers when the FTM counter changes from the MOD register value to its next value or when a channel (j) match occurs. A match occurs for the channel (j) when FTM counter = C(j)V.

Address: Base address + 98h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						LDOK	0	CH7SEL	CH6SEL	CH5SEL	CH4SEL	CH3SEL	CH2SEL	CH1SEL	CH0SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_PWMLOAD field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 LDOK	Load Enable Enables the loading of the MOD, CNTIN, and CV registers with the values of their write buffers. 0 Loading updated values is disabled. 1 Loading updated values is enabled.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7SEL	Channel 7 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
6 CH6SEL	Channel 6 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
5 CH5SEL	Channel 5 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
4 CH4SEL	Channel 4 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
3 CH3SEL	Channel 3 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
2 CH2SEL	Channel 2 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
1 CH1SEL	Channel 1 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
0 CH0SEL	Channel 0 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.

15.5 Functional description

The notation used in this document to represent the counters and the generation of the signals is shown in the following figure.

Figure 15-2. Notation used

15.5.1 Clock source

The FTM has only one clock domain: the system clock.

15.5.1.1 Counter clock source

The CLKS[1:0] bits in the SC register selects clock sources for the FTM counter or disables the FTM counter. After any chip reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The fixed frequency clock is an alternative clock source for the FTM counter that allows the selection of a clock other than the system clock or an external clock. This clock input is defined by chip integration; see the chip-specific FTM information for further details. Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed 1/2 of the system clock frequency.

The external clock passes through a synchronizer clocked by the system clock to assure that counter transitions are properly aligned to system clock transitions.Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must not exceed 1/4 of the system clock frequency.

15.5.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter. The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and FTM counter.

Figure 15-3. Example of the prescaler counter

15.5.3 Counter

The FTM has a 16-bit counter that is used by the channels either for input or output modes. The FTM counter clock is the selected clock divided by the prescaler.

The FTM counter has these modes of operation:

- Up counting
- Up-down counting

15.5.3.1 Up counting

Up counting is selected when:

- CPWMS = 0

CNTIN defines the starting value of the count and MOD defines the final value of the count, see the following figure. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with the value of CNTIN.

The FTM period when using up counting is (MOD – CNTIN + 0x0001) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to CNTIN.

Figure 15-4. Example of FTM up and signed counting

Table 15-5. FTM counting based on CNTIN value

When	Then
CNTIN = 0x0000	The FTM counting is equivalent to TPM up counting, that is, up and unsigned counting. See the following figure.
CNTIN[15] = 1	The initial value of the FTM counter is a negative number in two's complement, so the FTM counting is up and signed.
CNTIN[15] = 0 and CNTIN ≠ 0x0000	The initial value of the FTM counter is a positive number, so the FTM counting is up and unsigned.

Figure 15-5. Example of FTM up counting with CNTIN = 0x0000

Note

- FTM operation is only valid when the value of the CNTIN register is less than the value of the MOD register, either in the unsigned counting or signed counting. It is the responsibility of the software to ensure that the values in the CNTIN and MOD registers meet this requirement. Any values of CNTIN and MOD that do not satisfy this criteria can result in unpredictable behavior.
- MOD = CNTIN is a redundant condition. In this case, the FTM counter is always equal to MOD and the TOF bit is set in each rising edge of the FTM counter clock.
- When MOD = 0x0000, CNTIN = 0x0000, for example after reset, and FTMEN = 1, the FTM counter remains stopped at 0x0000 until a non-zero value is written into the MOD or CNTIN registers.
- Setting CNTIN to be greater than the value of MOD is not recommended as this unusual setting may make the FTM operation difficult to comprehend. However, there is no restriction on this configuration, and an example is shown in the following figure.

Figure 15-6. Example of up counting when the value of CNTIN is greater than the value of MOD

15.5.3.2 Up-down counting

Up-down counting is selected when:

- CPWMS = 1

CNTIN defines the starting value of the count and MOD defines the final value of the count. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to the value of CNTIN and the up-down counting restarts.

The FTM period when using up-down counting is 2 × (MOD – CNTIN) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to (MOD – 1).

If (CNTIN = 0x0000), the FTM counting is equivalent to TPM up-down counting, that is, up-down and unsigned counting. See the following figure.

Figure 15-7. Example of up-down counting when CNTIN = 0x0000

Note

When CNTIN is different from zero in the up-down counting, a valid CPWM signal is generated:

- if CnV > CNTIN, or
- if CnV = 0 or if CnV[15] = 1. In this case, 0% CPWM is generated.

15.5.3.3 Free running counter

If (FTMEN = 0) and (MOD = 0x0000 or MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000. See the following figure.

Figure 15-8. Example when the FTM counter is free running

The FTM counter is also a free running counter when:

- FTMEN = 1
- CPWMS = 0
- CNTIN = 0x0000, and
- MOD = 0xFFFF

15.5.3.4 Counter reset

Any one of the following cases resets the FTM counter to the value in the CNTIN register and the channels output to its initial value, except for channels in Output Compare mode.

- Any write to CNT.
- FTM counter synchronization.

15.5.3.5 When the TOF bit is set

The NUMTOF[4:0] bits define the number of times that the FTM counter overflow should occur before the TOF bit to be set. If NUMTOF[4:0] = 0x00, then the TOF bit is set at each FTM counter overflow.

Initialize the FTM counter, by writing to CNT, after writing to the NUMTOF[4:0] bits to avoid confusion about when the first counter overflow will occur.

Figure 15-9. Periodic TOF when NUMTOF = 0x02

Figure 15-10. Periodic TOF when NUMTOF = 0x00

15.5.4 Input Capture mode

The Input Capture mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0
- MSnB:MSnA = 0:0, and
- ELSnB:ELSnA ≠ 0:0

When a selected edge occurs on the channel input, the current value of the FTM counter is captured into the CnV register, at the same time the CHnF bit is set and the channel interrupt is generated if enabled by CHnIE = 1. See the following figure.

When a channel is configured for input capture, the FTMxCHn pin is an edge-sensitive input. ELSnB:ELSnA control bits determine which edge, falling or rising, triggers input-capture event. Note that the maximum frequency for the channel input signal to be detected correctly is system clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register is ignored in Input Capture mode.

While in Debug mode, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of Debug, is captured into the CnV register and the CHnF bit is set.

Figure 15-11. Input Capture mode

If the channel input does not have a filter enabled, then the input signal is always delayed 3 rising edges of the system clock, that is, two rising edges to the synchronizer plus one more rising edge to the edge detector. In other words, the CHnF bit is set on the third rising edge of the system clock after a valid edge occurs on the channel input.

15.5.4.1 Filter for Input Capture mode

The filter function is only available on channels 0, 1, 2, and 3.

First, the input signal is synchronized by the system clock. Following synchronization, the input signal enters the filter block. See the following figure.

Figure 15-12. Channel input filter

When there is a state change in the input signal, the counter is reset and starts counting up. As long as the new state is stable on the input, the counter continues to increment. When the counter is equal to CHnFVAL[3:0], the state change of the input signal is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the input signal before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by CHnFVAL[3:0] (× 4 system clocks) is regarded as a glitch and is not passed on to the edge detector. A timing diagram of the input filter is shown in the following figure.

The filter function is disabled when CHnFVAL[3:0] bits are zero. In this case, the input signal is delayed 3 rising edges of the system clock. If (CHnFVAL[3:0] ≠ 0000), then the input signal is delayed by the minimum pulse width (CHnFVAL[3:0] × 4 system clocks) plus a further 4 rising edges of the system clock: two rising edges to the synchronizer, one rising edge to the filter output, plus one more to the edge detector. In other words, CHnF is set (4 + 4 × CHnFVAL[3:0]) system clock periods after a valid edge occurs on the channel input.

The clock for the counter in the channel input filter is the system clock divided by 4.

* Note: Filter output is delayed one system clock of filter counter logic output.

Figure 15-13. Channel input filter example

The figure below shows an example of input capture with filter enabled and the delay added by each part of the input capture logic. Note that the input signal is delayed only by the synchronizer and edge dector logic if the filter is disabled.

Figure 15-14. Input capture example

15.5.5 Output Compare mode

The Output Compare mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB:MSnA = 0:1

In Output Compare mode, the FTM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared, or toggled.

When a channel is initially configured to Toggle mode, the previous value of the channel output is held until the first output compare event occurs.

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV).

Figure 15-15. Example of the Output Compare mode when the match toggles the channel output

Figure 15-16. Example of the Output Compare mode when the match clears the channel output

Figure 15-17. Example of the Output Compare mode when the match sets the channel output

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not modified and controlled by FTM.

15.5.6 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB = 1

The EPWM period is determined by (MOD − CNTIN + 0x0001) and the pulse width (duty cycle) is determined by (CnV − CNTIN).

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an FTM.

Figure 15-18. EPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced low at the channel (n) match (FTM counter = CnV). See the following figure.

Figure 15-19. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced high at the channel (n) match (FTM counter = CnV). See the following figure.

Figure 15-20. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match.

If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set.Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

Note

When CNTIN is different from zero the following EPWM signals can be generated:

- 0% EPWM signal if CnV = CNTIN,
- EPWM signal between 0% and 100% if CNTIN < CnV <= MOD,
- 100% EPWM signal when CNTIN > CnV or CnV > MOD.

15.5.7 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- DECAPEN = 0
- COMBINE = 0, and
- CPWMS = 1

The CPWM pulse width (duty cycle) is determined by 2 × (CnV − CNTIN) and the period is determined by 2 × (MOD − CNTIN). See the following figure. MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

In the CPWM mode, the FTM counter counts up until it reaches MOD and then counts down until it reaches CNTIN.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = CnV) when the FTM counting is down (at the begin of the pulse width) and when the FTM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are aligned with the value of CNTIN.

The other channel modes are not compatible with the up-down counter (CPWMS = 1). Therefore, all FTM channels must be used in CPWM mode when (CPWMS = 1).

Figure 15-21. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the FTM counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up. See the following figure.

Figure 15-22. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up. See the following figure.

Figure 15-23. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) or CnV is a negative value, that is (CnV[15] = 1), then the channel (n) output is a 0% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match.

If CnV is a positive value, that is (CnV[15] = 0), (CnV ≥ MOD), and (MOD ≠ 0x0000), then the channel (n) output is a 100% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match. This implies that the usable range of periods set by MOD is 0x0001 through 0x7FFE, 0x7FFF if you do not need to generate a 100% duty cycle CPWM signal. This is not a significant limitation because the resulting period is much longer than required for normal applications.

The CPWM mode must not be used when the FTM counter is a free running counter.

15.5.8 Combine mode

The Combine mode is selected when:

- DECAPEN = 0

- COMBINE = 1, and
- CPWMS = 0

In Combine mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output.

In the Combine mode, the PWM period is determined by (MOD − CNTIN + 0x0001) and the PWM pulse width (duty cycle) is determined by (|C(n+1)V − C(n)V|).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = C(n)V). The CH(n+1)F bit is set and the channel (n +1) interrupt is generated, if CH(n+1)IE = 1, at the channel (n+1) match (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced high at the channel (n) match (FTM counter = C(n)V). See the following figure.

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced high at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced low at the channel (n) match (FTM counter = C(n)V). See the following figure.

In Combine mode, the ELS(n+1)B and ELS(n+1)A bits are not used in the generation of the channels (n) and (n+1) output. However, if (ELSnB:ELSnA = 0:0) then the channel (n) output is not controlled by FTM, and if (ELS(n+1)B:ELS(n+1)A = 0:0) then the channel (n+1) output is not controlled by FTM.

Figure 15-24. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

Figure 15-25. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V < C(n+1)V)

Figure 15-26. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n+1)V = MOD)

Figure 15-27. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD)

Figure 15-28. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n)V is Almost Equal to CNTIN) and (C(n+1)V = MOD)

Figure 15-29. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD) and (C(n+1)V is Almost Equal to MOD)

Figure 15-30. Channel (n) output if C(n)V and C(n+1)V are not between CNTIN and MOD

Figure 15-31. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V = C(n+1)V)

Figure 15-32. Channel (n) output if (C(n)V = C(n+1)V = CNTIN)

Figure 15-33. Channel (n) output if (C(n)V = C(n+1)V = MOD)

Figure 15-34. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V > C(n+1)V)

Figure 15-35. Channel (n) output if (C(n)V < CNTIN) and (CNTIN < C(n+1)V < MOD)

Figure 15-36. Channel (n) output if (C(n+1)V < CNTIN) and (CNTIN < C(n)V < MOD)

Figure 15-37. Channel (n) output if (C(n)V > MOD) and (CNTIN < C(n+1)V < MOD)

Figure 15-38. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V < MOD)

Figure 15-39. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V = MOD)

15.5.8.1 Asymmetrical PWM

In Combine mode, the control of the PWM signal first edge, when the channel (n) match occurs, that is, FTM counter = C(n)V, is independent of the control of the PWM signal second edge, when the channel (n+1) match occurs, that is, FTM counter = C(n+1)V. So, Combine mode allows the generation of asymmetrical PWM signals.

15.5.9 Complementary mode

The Complementary mode is selected when:

- DECAPEN = 0
- COMP = 1

In Complementary mode, the channel (n+1) output is the inverse of the channel (n) output.

So, the channel (n+1) output is the same as the channel (n) output when:

- DECAPEN = 0
- COMP = 0

Figure 15-40. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = 1:0)

Figure 15-41. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = X:1)

NOTE

The complementary mode is not available in Output Compare mode.

15.5.10 Registers updated from write buffers

15.5.10.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 15-6. CNTIN register update

When	Then CNTIN register is updated
CLKS[1:0] = 0:0	When CNTIN register is written, independent of FTMEN bit.
• FTMEN = 0, or • CNTINC = 0	At the next system clock after CNTIN was written.
• FTMEN = 1, • SYNCMODE = 1, and • CNTINC = 1	By the CNTIN register synchronization.

15.5.10.2 MOD register update

The following table describes when MOD register is updated:

Table 15-7. MOD register update

When	Then MOD register is updated
CLKS[1:0] = 0:0	When MOD register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the CPWMS bit, that is: • If the selected mode is not CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	By the MOD register synchronization.

15.5.10.3 CnV register update

The following table describes when CnV register is updated:

Table 15-8. CnV register update

When	Then CnV register is updated
CLKS[1:0] = 0:0	When CnV register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the selected mode, that is: • If the selected mode is Output Compare, then CnV register is updated on the next FTM counter change, end of the prescaler counting, after CnV register was written. • If the selected mode is EPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	According to the selected mode, that is: • If the selected mode is output compare then CnV register is updated according to the SYNCEN bit. If (SYNCEN = 0) then CnV register is updated after CnV register was written at the next change of the FTM counter, the end of the prescaler counting. If (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization. • If the selected mode is not output compare and (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization.

15.5.11 PWM synchronization

The PWM synchronization provides an opportunity to update the MOD, CNTIN, CnV, OUTMASK, INVCTRL and SWOCTRL registers with their buffered value and force the FTM counter to the CNTIN register value.

Note

The legacy PWM synchronization (SYNCMODE = 0) is a subset of the enhanced PWM synchronization (SYNCMODE = 1). Thus, only the enhanced PWM synchronization must be used.

15.5.11.1 Hardware trigger

Three hardware trigger signal inputs of the FTM module are enabled when TRIGn = 1, where n = 0, 1 or 2 corresponding to each one of the input signals, respectively. The hardware trigger input n is synchronized by the system clock. The PWM synchronization with hardware trigger is initiated when a rising edge is detected at the enabled hardware trigger inputs.

If (HWTRIGMODE = 0) then the TRIGn bit is cleared when 0 is written to it or when the trigger n event is detected.

In this case, if two or more hardware triggers are enabled (for example, TRIG0 and TRIG1 = 1) and only trigger 1 event occurs, then only TRIG1 bit is cleared. If a trigger n event occurs together with a write setting TRIGn bit, then the synchronization is initiated, but TRIGn bit remains set due to the write operation.

Note
All hardware trigger inputs have the same behavior.

Figure 15-42. Hardware trigger event with HWTRIGMODE = 0

If HWTRIGMODE = 1, then the TRIGn bit is only cleared when 0 is written to it.

NOTE

The HWTRIGMODE bit must be 1 only with enhanced PWM synchronization (SYNCMODE = 1).

15.5.11.2 Software trigger

A software trigger event occurs when 1 is written to the SYNC[SWSYNC] bit. The SWSYNC bit is cleared when 0 is written to it or when the PWM synchronization, initiated by the software event, is completed.

If another software trigger event occurs (by writing another 1 to the SWSYNC bit) at the same time the PWM synchronization initiated by the previous software trigger event is ending, a new PWM synchronization is started and the SWSYNC bit remains equal to 1.

If SYNCMODE = 0 then the SWSYNC bit is also cleared by FTM according to PWMSYNC and REINIT bits. In this case if (PWMSYNC = 1) or (PWMSYNC = 0 and REINIT = 0) then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see Boundary cycle and loading points and the following figure. If (PWMSYNC = 0) and (REINIT = 1) then SWSYNC bit is cleared when the software trigger event occurs.

If SYNCMODE = 1 then the SWSYNC bit is also cleared by FTM according to the SWRSTCNT bit. If SWRSTCNT = 0 then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see the following figure. If SWRSTCNT = 1 then SWSYNC bit is cleared when the software trigger event occurs.

Figure 15-43. Software trigger event

15.5.11.3 Boundary cycle and loading points

The boundary cycle definition is important for the loading points for the registers MOD, CNTIN, and C(n)V.

In Up counting mode, the boundary cycle is defined as when the counter wraps to its initial value (CNTIN). If in Up-down counting mode, then the boundary cycle is defined as when the counter turns from down to up counting and when from up to down counting.

The following figure shows the boundary cycles and the loading points for the registers. In the Up Counting mode, the loading points are enabled if one of CNTMIN or CTMAX bits are 1. In the Up-Down Counting mode, the loading points are selected by CNTMIN and CNTMAX bits, as indicated in the figure. These loading points are safe places for register updates thus allowing a smooth transitions in PWM waveform generation.

For both counting modes, if neither CNTMIN nor CNTMAX are 1, then the boundary cycles are not used as loading points for registers updates. See the register synchronization descriptions in the following sections for details.

Figure 15-44. Boundary cycles and loading points

15.5.11.4 MOD register synchronization

The MOD register synchronization updates the MOD register with its buffer value. This synchronization is enabled if (FTMEN = 1).

The MOD register synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, it is expected that the MOD register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the MOD register synchronization depends on SWWRBUF, SWRSTCNT, HWWRBUF, and HWRSTCNT bits according to this flowchart:

Figure 15-45. MOD register synchronization flowchart

In the case of legacy PWM synchronization, the MOD register synchronization depends on PWMSYNC and REINIT bits according to the following description.

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 0), then this synchronization is made on the next selected loading point after an enabled trigger event takes place. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected

loading point. If the trigger event was a hardware trigger, then the trigger enable bit (TRIGn) is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

Figure 15-46. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 0), and software trigger was used

Figure 15-47. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 0), and a hardware trigger was used

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 1), then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

Figure 15-48. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 1), and software trigger was used

Figure 15-49. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 1), and a hardware trigger was used

If (SYNCMODE = 0) and (PWMSYNC = 1), then this synchronization is made on the next selected loading point after the software trigger event takes place. The SWSYNC bit is cleared on the next selected loading point:

Figure 15-50. MOD synchronization with (SYNCMODE = 0) and (PWMSYNC = 1)

15.5.11.5 CNTIN register synchronization

The CNTIN register synchronization updates the CNTIN register with its buffer value.

This synchronization is enabled if (FTMEN = 1), (SYNCMODE = 1), and (CNTINC = 1). The CNTIN register synchronization can be done only by the enhanced PWM synchronization (SYNCMODE = 1). The synchronization mechanism is the same as the MOD register synchronization done by the enhanced PWM synchronization; see MOD register synchronization.

15.5.11.6 C(n)V and C(n+1)V register synchronization

The C(n)V and C(n+1)V registers synchronization updates the C(n)V and C(n+1)V registers with their buffer values.

This synchronization is enabled if (FTMEN = 1) and (SYNCEN = 1). The synchronization mechanism is the same as the MOD register synchronization. However, it is expected that the C(n)V and C(n+1)V registers be synchronized only by the enhanced PWM synchronization (SYNCMODE = 1).

15.5.11.7 OUTMASK register synchronization

The OUTMASK register synchronization updates the OUTMASK register with its buffer value.

The OUTMASK register can be updated at each rising edge of system clock (SYNCHOM = 0), by the enhanced PWM synchronization (SYNCHOM = 1 and SYNCMODE = 1) or by the legacy PWM synchronization (SYNCHOM = 1 and SYNCMODE = 0). However, it is expected that the OUTMASK register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the OUTMASK register synchronization depends on SWOM and HWOM bits. See the following flowchart:

Figure 15-51. OUTMASK register synchronization flowchart

In the case of legacy PWM synchronization, the OUTMASK register synchronization depends on PWMSYNC bit according to the following description.

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 0), then this synchronization is done on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

Figure 15-52. OUTMASK synchronization with (SYNCMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0) and software trigger was used

Figure 15-53. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 1), then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger. An example with a hardware trigger follows.

Figure 15-54. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 1), and a hardware trigger was used

15.5.11.8 INVCTRL register synchronization

The INVCTRL register synchronization updates the INVCTRL register with its buffer value.

The INVCTRL register can be updated at each rising edge of system clock (INVC = 0) or by the enhanced PWM synchronization (INVC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the INVCTRL register synchronization depends on SWINVC and HWINVC bits.

Figure 15-55. INVCTRL register synchronization flowchart

15.5.11.9 SWOCTRL register synchronization

The SWOCTRL register synchronization updates the SWOCTRL register with its buffer value.

The SWOCTRL register can be updated at each rising edge of system clock (SWOC = 0) or by the enhanced PWM synchronization (SWOC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the SWOCTRL register synchronization depends on SWSOC and HWSOC bits.

Figure 15-56. SWOCTRL register synchronization flowchart

15.5.11.10 FTM counter synchronization

The FTM counter synchronization is a mechanism that allows the FTM to restart the PWM generation at a certain point in the PWM period. The channels outputs are forced to their initial value, except for channels in Output Compare mode, and the FTM counter is forced to its initial counting value defined by CNTIN register.

The following figure shows the FTM counter synchronization. Note that after the synchronization event occurs, the channel (n) is set to its initial value and the channel (n +1) is not set to its initial value due to a specific timing of this figure in which the deadtime insertion prevents this channel output from transitioning to 1. If no deadtime insertion is selected, then the channel (n+1) transitions to logical value 1 immediately after the synchronization event occurs.

Figure 15-57. FTM counter synchronization

The FTM counter synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, the FTM counter must be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the FTM counter synchronization depends on SWRSTCNT and HWRSTCNT bits according to the following flowchart.

Figure 15-58. FTM counter synchronization flowchart

In the case of legacy PWM synchronization, the FTM counter synchronization depends on REINIT and PWMSYNC bits according to the following description.

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 0) then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

Figure 15-59. FTM counter synchronization with (SYNCMODE = 0), (REINIT = 1), (PWMSYNC = 0), and software trigger was used

Figure 15-60. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 1) then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger.

Figure 15-61. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 1), and a hardware trigger was used

15.5.12 Inverting

The invert functionality swaps the signals between channel (n) and channel (n+1) outputs. The inverting operation is selected when:

- DECAPEN = 0
- COMP = 1, and
- INVm = 1 (where m represents a channel pair)

The INVm bit in INVCTRL register is updated with its buffer value according to INVCTRL register synchronization

In High-True (ELSnB:ELSnA = 1:0) Combine mode, the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN), forced high at the channel (n) match and forced low at the channel (n+1) match. If the inverting is selected, the channel (n) output behavior is changed to force high at the beginning of the PWM period, force low at the channel (n) match and force high at the channel (n+1) match. See the following figure.

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 15-62. Channels (n) and (n+1) outputs after the inverting in High-True (ELSnB:ELSnA = 1:0) Combine mode

Note that the ELSnB:ELSnA bits value should be considered because they define the active state of the channels outputs. In Low-True (ELSnB:ELSnA = X:1) Combine mode, the channel (n) output is forced high at the beginning of the period, forced low at the channel (n) match and forced high at the channel (n+1) match. When inverting is selected, the channels (n) and (n+1) present waveforms as shown in the following figure.

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 15-63. Channels (n) and (n+1) outputs after the inverting in Low-True (ELSnB:ELSnA = X:1) Combine mode

Note

The inverting feature is not available in Output Compare mode.

15.5.13 Software output control

The software output control forces the channel output according to software defined values at a specific time in the PWM generation.

The software output control is selected when:

- DECAPEN = 0, and
- CHnOC = 1

The CHnOC bit enables the software output control for a specific channel output and the CHnOCV selects the value that is forced to this channel output.

Both CHnOC and CHnOCV bits in SWOCTRL register are buffered and updated with their buffer value according to SWOCTRL register synchronization.

The following figure shows the channels (n) and (n+1) outputs signals when the software output control is used. In this case the channels (n) and (n+1) are set to Combine and Complementary mode.

NOTE
CH(n)OCV = 1 and CH(n+1)OCV = 0.

Figure 15-64. Example of software output control in Combine and Complementary mode

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is zero.

Table 15-9. Software ouput control behavior when (COMP = 0)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to one

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is one.

Table 15-10. Software ouput control behavior when (COMP = 1)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to zero

Note

- The CH(n)OC and CH(n+1)OC bits should be equal.
- The COMP bit must not be modified when software output control is enabled, that is, CH(n)OC = 1 and/or CH(n +1)OC = 1.
- Software output control has the same behavior with disabled or enabled FTM counter (see the CLKS field description in the Status and Control register).

15.5.14 Deadtime insertion

The deadtime insertion is enabled when (DTEN = 1) and (DTVAL[5:0] is non- zero).

DEADTIME register defines the deadtime delay that can be used for all FTM channels. The DTPS[1:0] bits define the prescaler for the system clock and the DTVAL[5:0] bits define the deadtime modulo, that is, the number of the deadtime prescaler clocks.

The deadtime delay insertion ensures that no two complementary signals (channels (n) and (n+1)) drive the active state at the same time.

If POL(n) = 0, POL(n+1) = 0, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the low value until the end of the deadtime delay when the channel (n) output is set. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the low value until the end of the deadtime delay when the channel (n+1) output is set. See the following figures.

If POL(n) = 1, POL(n+1) = 1, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the high value until the end of the deadtime delay when the channel (n) output is cleared. Similarly,

when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the high value until the end of the deadtime delay when the channel (n +1) output is cleared.

Figure 15-65. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0

Figure 15-66. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 0

NOTE

- The deadtime feature must be used only in Complementary mode.
- The deadtime feature is not available in Output Compare mode.

15.5.14.1 Deadtime insertion corner cases

If (PS[2:0] is cleared), (DTPS[1:0] = 0:0 or DTPS[1:0] = 0:1):

- and the deadtime delay is greater than or equal to the channel (n) duty cycle ((C(n +1)V – C(n)V) × system clock), then the channel (n) output is always the inactive value (POL(n) bit value).
- and the deadtime delay is greater than or equal to the channel (n+1) duty cycle ((MOD – CNTIN + 1 – (C(n+1)V – C(n)V)) × system clock), then the channel (n+1) output is always the inactive value (POL(n+1) bit value).

Although, in most cases the deadtime delay is not comparable to channels (n) and (n+1) duty cycle, the following figures show examples where the deadtime delay is comparable to the duty cycle.

Figure 15-67. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channel (n+1) duty cycle

Figure 15-68. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channels (n) and (n+1) duty cycle

15.5.15 Output mask

The output mask can be used to force channels output to their inactive state through software. For example: to control a BLDC motor.

Any write to the OUTMASK register updates its write buffer. The OUTMASK register is updated with its buffer value by PWM synchronization; see OUTMASK register synchronization.

If CHnOM = 1, then the channel (n) output is forced to its inactive state (POLn bit value). If CHnOM = 0, then the channel (n) output is unaffected by the output mask. See the following figure.

Figure 15-69. Output mask with POLn = 0

The following table shows the output mask result before the polarity control.

Table 15-11. Output mask result for channel (n) before the polarity control

CHnOM	Output Mask Input	Output Mask Result
0	inactive state	inactive state
	active state	active state
1	inactive state	inactive state
	active state	

15.5.16 Polarity control

The POLn bit selects the channel (n) output polarity:

- If POLn = 0, the channel (n) output polarity is high, so the logical one is the active state and the logical zero is the inactive state.
- If POLn = 1, the channel (n) output polarity is low, so the logical zero is the active state and the logical one is the inactive state.

15.5.17 Initialization

The initialization forces the CHnOI bit value to the channel (n) output when a one is written to the INIT bit.

The initialization depends on COMP and DTEN bits. The following table shows the values that channels (n) and (n+1) are forced by initialization when the COMP and DTEN bits are zero.

Table 15-12. Initialization behavior when (COMP = 0 and DTEN = 0)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	0	is forced to zero	is forced to zero
0	1	is forced to zero	is forced to one
1	0	is forced to one	is forced to zero
1	1	is forced to one	is forced to one

The following table shows the values that channels (n) and (n+1) are forced by initialization when (COMP = 1) or (DTEN = 1).

Table 15-13. Initialization behavior when (COMP = 1 or DTEN = 1)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	X	is forced to zero	is forced to one
1	X	is forced to one	is forced to zero

Note

The initialization feature must be used only with disabled FTM counter. See the description of the CLKS field in the Status and Control register.

15.5.18 Features priority

The following figure shows the priority of the features used at the generation of channels (n) and (n+1) outputs signals.

NOTE
The channels (n) and (n+1) are in output compare, EPWM, CPWM or combine modes.

Figure 15-70. Priority of the features used at the generation of channels (n) and (n+1) outputs signals

Note

The Initialization feature must not be used with Inverting and Software output control features.

15.5.19 Channel trigger output

If CH(j)TRIG bit of the FTM External Trigger (FTM_EXTTRIG) register is set, where j = 0, 1, 2, 3, 4, or 5, then the FTM generates a trigger when the channel (j) match occurs (FTM counter = C(j)V).

The channel trigger output provides a trigger signal which has one FTM clock period width and is used for on-chip modules.

The FTM is able to generate multiple triggers in one PWM period. Because each trigger is generated for a specific channel, several channels are required to implement this functionality. This behavior is described in the following figure.

NOTE

(a) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(b) CH0TRIG = 1, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(c) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1
(d) CH0TRIG = 1, CH1TRIG = 1, CH2TRIG = 1, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1

Figure 15-71. Channel match trigger

15.5.20 Initialization trigger

If INITTRIGEN = 1, then the FTM generates a trigger when the FTM counter is updated with the CNTIN register value in the following cases.

- The FTM counter is automatically updated with the CNTIN register value by the selected counting mode.
- When there is a write to CNT register.
- When there is the FTM counter synchronization.
- If (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits.

The following figures show these cases.

Figure 15-72. Initialization trigger is generated when the FTM counting achieves the CNTIN register value

Figure 15-73. Initialization trigger is generated when there is a write to CNT register

Figure 15-74. Initialization trigger is generated when there is the FTM counter synchronization

Figure 15-75. Initialization trigger is generated if (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits

The initialization trigger output provides a trigger signal that is used for on-chip modules.

15.5.21 Capture Test mode

The Capture Test mode allows to test the CnV registers, the FTM counter and the interconnection logic between the FTM counter and CnV registers.

In this test mode, all channels must be configured for Input Capture mode and FTM counter must be configured to the Up counting.

When the Capture Test mode is enabled (CAPTEST = 1), the FTM counter is frozen and any write to CNT register updates directly the FTM counter; see the following figure. After it was written, all CnV registers are updated with the written value to CNT register and CHnF bits are set. Therefore, the FTM counter is updated with its next value according to its configuration. Its next value depends on CNTIN, MOD, and the written value to FTM counter.

The next reads of CnV registers return the written value to the FTM counter and the next reads of CNT register return FTM counter next value.

NOTE
- FTM counter configuration: (FTMEN = 1), (CAPTEST = 1), (CPWMS = 0), (CNTIN = 0x0000), and (MOD = 0xFFFF)
- FTM channel n configuration: input capture mode - (DECAPEN = 0), (COMBINE = 0), and (MSnB:MSnA = 0:0)

Figure 15-76. Capture Test mode

15.5.22 Dual Edge Capture mode

The Dual Edge Capture mode is selected if DECAPEN = 1. This mode allows to measure a pulse width or period of the signal on the input of channel (n) of a channel pair. The channel (n) filter can be active in this mode when n is 0 or 2.

* Filtering function for dual edge capture mode is only available in the channels 0 and 2

Figure 15-77. Dual Edge Capture mode block diagram

The MS(n)A bit defines if the Dual Edge Capture mode is one-shot or continuous.

The ELS(n)B:ELS(n)A bits select the edge that is captured by channel (n), and ELS(n+1)B:ELS(n+1)A bits select the edge that is captured by channel (n+1). If both ELS(n)B:ELS(n)A and ELS(n+1)B:ELS(n+1)A bits select the same edge, then it is the period measurement. If these bits select different edges, then it is a pulse width measurement.

In the Dual Edge Capture mode, only channel (n) input is used and channel (n+1) input is ignored.

If the selected edge by channel (n) bits is detected at channel (n) input, then CH(n)F bit is set and the channel (n) interrupt is generated (if CH(n)IE = 1). If the selected edge by channel (n+1) bits is detected at channel (n) input and (CH(n)F = 1), then CH(n+1)F bit is set and the channel (n+1) interrupt is generated (if CH(n+1)IE = 1).

The C(n)V register stores the value of FTM counter when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the value of FTM counter when the selected edge by channel (n+1) is detected at channel (n) input.

In this mode, a coherency mechanism ensures coherent data when the C(n)V and C(n+1)V registers are read. The only requirement is that C(n)V must be read before C(n+1)V.

Note

- The CH(n)F, CH(n)IE, MS(n)A, ELS(n)B, and ELS(n)A bits are channel (n) bits.
- The CH(n+1)F, CH(n+1)IE, MS(n+1)A, ELS(n+1)B, and ELS(n+1)A bits are channel (n+1) bits.
- The Dual Edge Capture mode must be used with ELS(n)B:ELS(n)A = 0:1 or 1:0, ELS(n+1)B:ELS(n+1)A = 0:1 or 1:0 and the FTM counter in Free running counter.

15.5.22.1 One-Shot Capture mode

The One-Shot Capture mode is selected when (DECAPEN = 1), and (MS(n)A = 0). In this capture mode, only one pair of edges at the channel (n) input is captured. The ELS(n)B:ELS(n)A bits select the first edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the second edge to be captured.

The edge captures are enabled while DECAP bit is set. For each new measurement in One-Shot Capture mode, first the CH(n)F and CH(n+1) bits must be cleared, and then the DECAP bit must be set.

In this mode, the DECAP bit is automatically cleared by FTM when the edge selected by channel (n+1) is captured. Therefore, while DECAP bit is set, the one-shot capture is in process. When this bit is cleared, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

Similarly, when the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

15.5.22.2 Continuous Capture mode

The Continuous Capture mode is selected when (DECAPEN = 1), and (MS(n)A = 1). In this capture mode, the edges at the channel (n) input are captured continuously. The ELS(n)B:ELS(n)A bits select the initial edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the final edge to be captured.

The edge captures are enabled while DECAP bit is set. For the initial use, first the CH(n)F and CH(n+1)F bits must be cleared, and then DECAP bit must be set to start the continuous measurements.

When the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers. The latest captured values are always available in these registers even after the DECAP bit is cleared.

In this mode, it is possible to clear only the CH(n+1)F bit. Therefore, when the CH(n+1)F bit is set again, the latest captured values are available in C(n)V and C(n+1)V registers.

For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the CH(n)F and CH(n+1)F bits to start new measurements.

15.5.22.3 Pulse width measurement

If the channel (n) is configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1) and the channel (n+1) to capture falling edges (ELS(n+1)B:ELS(n+1)A = 1:0), then the positive polarity pulse width is measured. If the channel (n) is configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0) and the channel (n+1) to capture rising edges (ELS(n +1)B:ELS(n+1)A = 0:1), then the negative polarity pulse width is measured.

The pulse width measurement can be made in One-Shot Capture mode or Continuous Capture mode.

The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The CH(n)F bit is set when the first edge of this pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 15-78. Dual Edge Capture – One-Shot mode for positive polarity pulse width measurement

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first edge of the positive polarity

pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 15-79. Dual Edge Capture – Continuous mode for positive polarity pulse width measurement

15.5.22.4 Period measurement

If the channels (n) and (n+1) are configured to capture consecutive edges of the same polarity, then the period of the channel (n) input signal is measured. If both channels (n) and (n+1) are configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1 and ELS(n+1)B:ELS(n+1)A = 0:1), then the period between two consecutive rising edges is measured. If both channels (n) and (n+1) are configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0 and ELS(n+1)B:ELS(n+1)A = 1:0), then the period between two consecutive falling edges is measured.

The period measurement can be made in One-Shot Capture mode or Continuous Capture mode.

The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next period. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two selected edges were captured and the C(n)V and C(n+1)V registers are ready for reading.

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 0, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 3: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 15-80. Dual Edge Capture – One-Shot mode to measure of the period between two consecutive rising edges

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the period were captured and the C(n)V and C(n+1)V registers are ready for reading.

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 15-81. Dual Edge Capture – Continuous mode to measure of the period between two consecutive rising edges

15.5.22.5 Read coherency mechanism

The Dual Edge Capture mode implements a read coherency mechanism between the FTM counter value captured in C(n)V and C(n+1)V registers. The read coherency mechanism is illustrated in the following figure. In this example, the channels (n) and (n

+1) are in Dual Edge Capture – Continuous mode for positive polarity pulse width measurement. Thus, the channel (n) is configured to capture the FTM counter value when there is a rising edge at channel (n) input signal, and channel (n+1) to capture the FTM counter value when there is a falling edge at channel (n) input signal.

When a rising edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n) capture buffer. The channel (n) capture buffer value is transferred to C(n)V register when a falling edge occurs in the channel (n) input signal. C(n)V register has the FTM counter value when the previous rising edge occurred, and the channel (n) capture buffer has the FTM counter value when the last rising edge occurred.

When a falling edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n+1) capture buffer. The channel (n+1) capture buffer value is transferred to C(n+1)V register when the C(n)V register is read.

In the following figure, the read of C(n)V returns the FTM counter value when the event 1 occurred and the read of C(n+1)V returns the FTM counter value when the event 2 occurred.

Figure 15-82. Dual Edge Capture mode read coherency mechanism

C(n)V register must be read prior to C(n+1)V register in dual edge capture one-shot and continuous modes for the read coherency mechanism works properly.

15.5.23 Debug mode

When the chip is in Debug mode, the BDMMODE[1:0] bits select the behavior of the FTM counter, the CH(n)F bit, the channels output, and the writes to the MOD, CNTIN, and C(n)V registers according to the following table.

Table 15-14. FTM behavior when the chip Is in Debug mode

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
00	Stopped	can be set	Functional mode	Writes to these registers bypass the registers buffers
01	Stopped	is not set	The channels outputs are forced to their safe value according to POLn bit	Writes to these registers bypass the registers buffers
10	Stopped	is not set	The channels outputs are frozen when the chip enters in Debug mode	Writes to these registers bypass the registers buffers
11	Functional mode	can be set	Functional mode	Functional mode

Note that if BDMMODE[1:0] = 2’b00 then the channels outputs remain at the value when the chip enters in Debug mode, because the FTM counter is stopped. However, the following situations modify the channels outputs in this Debug mode.

- Write any value to CNT register; see Counter reset. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for those channels set to Output Compare mode.
- FTM counter is reset by PWM Synchronization mode; see FTM counter synchronization. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for channels in Output Compare mode.
- In the channels outputs initialization, the channel (n) output is forced to the CH(n)OI bit value when the value 1 is written to INIT bit. See Initialization.

Note

If CLKS[1:0] = 2'b00 in BDM, a non-zero value is written to CLKS in BDM, and CnV = CNTIN when the BDM is disabled, then the CHnF bit is set (since if the channel is a 0% EPWM signal) when the BDM is disabled.

15.5.24 Intermediate load

The PWMLOAD register allows to update the MOD, CNTIN, and C(n)V registers with the content of the register buffer at a defined load point. In this case, it is not required to use the PWM synchronization.

There are multiple possible loading points for intermediate load:

Table 15-15. When possible loading points are enabled

Loading point	Enabled
When the FTM counter wraps from MOD value to CNTIN value	Always
At the channel (j) match (FTM counter = C(j)V)	When CHjSEL = 1

The following figure shows some examples of enabled loading points.

NOTE

(a) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(b) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(c) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 1, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(d) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0
(e) LDOK = 1, CH0SEL = 1, CH1SEL = 0, CH2SEL = 1, CH3SEL = 0, CH4SEL = 1, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0
(f) LDOK = 1, CH0SEL = 1, CH1SEL = 1, CH2SEL = 1, CH3SEL = 1, CH4SEL = 1, CH5SEL = 1, CH6SEL = 1, CH7SEL = 1

Figure 15-83. Loading points for intermediate load

After enabling the loading points, the LDOK bit must be set for the load to occur. In this case, the load occurs at the next enabled loading point according to the following conditions:

Table 15-16. Conditions for loads occurring at the next enabled loading point

When a new value was written	Then
To the MOD register	The MOD register is updated with its write buffer value.

Table continues on the next page...

Table 15-16. Conditions for loads occurring at the next enabled loading point (continued)

When a new value was written	Then
To the CNTIN register and CNTINC = 1	The CNTIN register is updated with its write buffer value.
To the C(n)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n)V register is updated with its write buffer value.
To the C(n+1)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n+1)V register is updated with its write buffer value.

NOTE

- If ELSjB and ELSjA bits are different from zero, then the channel (j) output signal is generated according to the configured output mode. If ELSjB and ELSjA bits are zero, then the generated signal is not available on channel (j) output.
- If CHjIE = 1, then the channel (j) interrupt is generated when the channel (j) match occurs.
- At the intermediate load neither the channels outputs nor the FTM counter are changed. Software must set the intermediate load at a safe point in time.

15.5.25 Global time base (GTB)

The global time base (GTB) is a FTM function that allows the synchronization of multiple FTM modules on a chip. The following figure shows an example of the GTB feature used to synchronize two FTM modules. In this case, the FTM A and B channels can behave as if just one FTM module was used, that is, a global time base.

Figure 15-84. Global time base (GTB) block diagram

The GTB functionality is implemented by the GTBEEN and GTBEOUT bits in the CONF register, the input signal *gtb_in*, and the output signal *gtb_out*. The GTBEEN bit enables gtb_in to control the FTM counter enable signal:

- If GTBEEN = 0, each one of FTM modules works independently according to their configured mode.
- If GTBEEN = 1, the FTM counter update is enabled only when gtb_in is 1.

In the configuration described in the preceding figure, FTM modules A and B have their FTM counters enabled if at least one of the gtb_out signals from one of the FTM modules is 1. There are several possible configurations for the interconnection of the gtb_in and gtb_out signals, represented by the example glue logic shown in the figure. Note that these configurations are chip-dependent and implemented outside of the FTM modules. See the chip-specific FTM information for the chip's specific implementation.

NOTE

- In order to use the GTB signals to synchronize the FTM counter of different FTM modules, the configuration of each FTM module should guarantee that its FTM counter starts counting as soon as the gtb_in signal is 1.
- The GTB feature does not provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during FTM operation. The GTB feature only allows the FTM counters to *start* their operation synchronously.

15.5.25.1 Enabling the global time base (GTB)

To enable the GTB feature, follow these steps for each participating FTM module:

1. Stop the FTM counter: Write 00b to SC[CLKS].
2. Program the FTM to the intended configuration. The FTM counter mode needs to be consistent across all participating modules.
3. Write 1 to CONF[GTBEEN] and write 0 to CONF[GTBEOUT] at the same time.
4. Select the intended FTM counter clock source in SC[CLKS]. The clock source needs to be consistent across all participating modules.
5. Reset the FTM counter: Write any value to the CNT register.

To initiate the GTB feature in the configuration described in the preceding figure, write 1 to CONF[GTBEOUT] in the FTM module used as the time base.

15.6 Reset overview

The FTM is reset whenever any chip reset occurs.

When the FTM exits from reset:

- the FTM counter and the prescaler counter are zero and are stopped (CLKS[1:0] = 00b);
- the timer overflow interrupt is zero, see Timer Overflow Interrupt;
- the channels interrupts are zero, see Channel (n) Interrupt;
- the channels are in input capture mode, see Input Capture mode;
- the channels outputs are zero;
- the channels pins are not controlled by FTM (ELS(n)B:ELS(n)A = 0:0) (See the table in the description of CnSC register).

The following figure shows the FTM behavior after the reset. At the reset (item 1), the FTM counter is disabled (see the description of the CLKS field in the Status and Control register), its value is updated to zero and the pins are not controlled by FTM (See the table in the description of CnSC register).

After the reset, the FTM should be configurated (item 2). It is necessary to define the FTM counter mode, the FTM counting limits (MOD and CNTIN registers value), the channels mode and CnV registers value according to the channels mode.

Thus, it is recommended to write any value to CNT register (item 3). This write updates the FTM counter with the CNTIN register value and the channels output with its initial value (except for channels in output compare mode) (Counter reset).

The next step is to select the FTM counter clock by the CLKS[1:0] bits (item 4). It is important to highlight that the pins are only controlled by FTM when CLKS[1:0] bits are different from zero (See the table in the description of CnSC register).

NOTES:
– CNTIN = 0x0010
– Channel (n) is in low-true combine mode with CNTIN < C(n)V < C(n+1)V < MOD
– C(n)V = 0x0015

Figure 15-85. FTM behavior after reset when the channel (n) is in Combine mode

The following figure shows an example when the channel (n) is in Output Compare mode and the channel (n) output is toggled when there is a match. In the Output Compare mode, the channel output is not updated to its initial value when there is a write to CNT register (item 3). In this case, use the software output control (Software output control) or the initialization (Initialization) to update the channel output to the selected value (item 4).

NOTES:
– CNTIN = 0x0010
– Channel (n) is in output compare and the channel (n) output is toggled when there is a match
– C(n)V = 0x0014

Figure 15-86. FTM behavior after reset when the channel (n) is in Output Compare mode

15.7 FTM Interrupts

15.7.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

15.7.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

15.8 Initialization Procedure

The following initialization procedure is recommended to configure the FlexTimer operation. This procedure can also be used to do a new configuration of the FlexTimer operation.

- Define the POL bits.
- Mask the channels outputs using SYNCHOM = 0. Two clocks after the write to OUTMASK, the channels output are in the safe value.
- (Re)Configuration FTM counter and channels to generation of periodic signals - Disable the clock. If the selected mode is Quadrature Decoder, then disable this mode. Examples of the (re)configuration:
    - Write to MOD.
    - Write to CNTIN.
    - Select OC, EPWM, CPWM, Combine, Complement modes for all channels that will be used
    - Select the high-true and low-true channels modes.
    - Write to CnV for all channels that will be used .
    - (Re)Configure deadtime.
    - Do not use the SWOC without SW synchronization (see item 6).
    - Do not use the Inverting without SW synchronization (see item 6).
    - Do not use the Initialization.
    - Do not change the polarity control.
    - Do not configure the HW synchronization
- Write any value to CNT. The FTM Counter is reset and the channels output are updated according to new configuration.
- Enable the clock. Write to CLKS[1:0] bits a value different from zero. If in the Quadrature Decoder mode, enable this mode.
- Configure the SW synchronization for SWOC (if it is necessary), Inverting (if it is necessary) and Output Mask (always)

- Select synchronization for Output Mask Write to SYNC (SWSYNC = 0, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)
- Write to SYNCONF.
  - HW Synchronization can not be enabled (HWSOC = 0, HWINVC = 0, HWOM = 0, HWWRBUF = 0, HWRSTCNT = 0, HWTRIGMODE = 0).
  - SW Synchronization for SWOC (if it is necessary): SWSOC = [0/1] and SWOC = [0/1].
  - SW Synchronization for Inverting (if it is necessary): SWINVC = [0/1] and INVC = [0/1].
  - SW Synchronization for SWOM (always): SWOM = 1. No enable the SW Synchronization for write buffers (because the writes to registers with write buffer are done using CLKS[1:0] = 2’b00): SWWRBUF = 0 and CNTINC = 0 .
  - SW Synchronization for counter reset (always): SWRSTCNT = 1.
  - Enhanced synchronization (always): SYNCMODE = 1
- If the SWOC is used (SWSOC = 1 and SWOC = 1), then write to SWOCTRL register.
- If the Inverting is used (SWINVC = 1 and INVC = 1), then write to INVCTRL register.
- Write to OUTMASK to enable the masked channels.

- Generate the Software Trigger Write to SYNC (SWSYNC = 1, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)

Chapter 16
General Purpose I/O (GPIO)

16.1 The GPIO module as implemented on the chip

This section provides details about how the GPIO module is integrated into this chip.

Table 16-1. GPIO module as implemented on chip

Module name	Module base address	Supported	Number of ports
GPIO1	0x230_0000	28:0	29
GPIO2	0x231_0000	30:0	31
GPIO3	0x232_0000	30:6	25
GPIO4	0x233_0000	31:0	32

16.2 GPIO overview

This chapter describes the general-purpose I/O (GPIO) module, including signal descriptions, register settings and interrupt capabilities.

This figure shows the block diagram for the GPIO module.

Figure 16-1. GPIO module block diagram

In general, the GPIO module supports up to 32 general-purpose I/O ports. Each port can be configured as an input or as an output. However, some implementations may restrict specific ports to input-only, output-only, or reserved (unimplemented). See "The GPIO module as implemented on the chip" section for more information. If a port is configured as an input, it can optionally generate an interrupt upon detection of a change. If a port is configured as an output, it can be individually configured as an open-drain or a fully active output.

16.3 GPIO features summary

The GPIO unit includes the following features:

- Supports 32 general-purpose input/output ports
- All signals are high-impedance during reset
- Open-drain capability on all ports
- All ports can optionally generate an interrupt upon changing their state
- Ports may be multiplexed with other functional signals

16.4 GPIO signal descriptions

This table provides detailed descriptions of the external GPIO signals. Note that depending on the signal multiplexing, some signals may not be available.

Table 16-2. GPIO external signals-detailed signal descriptions

Signal	I/O	Description	
GPIO*n*[0:31]	I/O	General Purpose I/O. Each signal can be individually set to act as input or output, according to application needs. Some implementations may restrict specific ports to input-only, output-only, or reserved (unimplemented). See "The GPIO module as implemented on the chip" section for more information.	
		State Meaning	Asserted/Negated-Defined per application.
		Timing	Assertion/Negation-Inputs can be asserted completely asynchronously. Outputs are asynchronous to any externally visible clock.

16.5 GPIO register descriptions

The programmable register map for the GPIO module occupies 28 bytes of memory-mapped space. The full register address is comprised of the base address (specified in CCSR address space) plus the module base address, plus the specific register's offset within the module. The table below shows the memory map for the GPIO module.

All GPIO registers are 32 bits wide located on 32-bit address boundaries. Note that reading undefined portions of the memory map returns all zeros and writing has no effect.

16.5.1 GPIO memory map

GPIO1 base address: 230_0000h

GPIO2 base address: 231_0000h

GPIO3 base address: 232_0000h

GPIO4 base address: 233_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	GPIO direction register (GPDIR)	32	RW	0000_0000h
4h	GPIO open drain register (GPODR)	32	RW	0000_0000h
8h	GPIO data register (GPDAT)	32	RW	0000_0000h
Ch	GPIO interrupt event register (GPIER)	32	W1C	Table 16-2
10h	GPIO interrupt mask register (GPIMR)	32	RW	0000_0000h
14h	GPIO interrupt control register (GPICR)	32	RW	0000_0000h
18h	GPIO Input Buffer Enable Register (GPIBE)	32	RW	0000_0000h

16.5.2 GPIO direction register (GPDIR)

16.5.2.1 Offset

Register	Offset
GPDIR	0h

16.5.2.2 Function

The GPIO direction register (GPDIR) defines the direction of the individual ports.

16.5.2.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	DRn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	DRn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.2.4 Fields

Field	Function
0-31 DRn	Direction. Indicates whether a pin is used as an input or an output. 00000000000000000000000000000000b - The corresponding pin is an input. 00000000000000000000000000000001b - The corresponding pin is an output.

16.5.3 GPIO open drain register (GPODR)

16.5.3.1 Offset

Register	Offset
GPODR	4h

16.5.3.2 Function

The GPIO open drain register (GPODR) defines the way individual ports drive their output.

16.5.3.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	ODn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ODn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.3.4 Fields

Field	Function
0-31 ODn	Open drain configuration. Indicates whether a signal is actively driven as an output or is an open-drain driver. This register has no effect on signals programmed as inputs in GPDIR. 00000000000000000000000000000000b - The corresponding signal is actively driven as an output. 00000000000000000000000000000001b - The corresponding signal is an open-drain driver. As an output, the signal is driven active-low, otherwise it is not driven (high impedance).

16.5.4 GPIO data register (GPDAT)

16.5.4.1 Offset

Register	Offset
GPDAT	8h

16.5.4.2 Function

The GPIO data register (GPDAT) carries the data in/out for the individual ports.

16.5.4.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	Dn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	Dn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.4.4 Fields

Field	Function
0-31 Dn	Data. Writes to this register latches the data which is presented on the external pins provided the corresponding GPDIR bit is configured as an output. When GPDIR is in output mode, GPDAT read operation returns data at pin. When GPDIR is in input mode, GPDAT read operation returns state of the port.

16.5.5 GPIO interrupt event register (GPIER)

16.5.5.1 Offset

Register	Offset
GPIER	Ch

16.5.5.2 Function

The GPIO interrupt event register (GPIER) carries information of the events that caused an interrupt. Each bit in GPIER, corresponds to an interrupt source. GPIER bits are cleared by writing ones. However, writing zero has no effect.

NOTE

Some implementations may ignore the interrupt mask as configured in GPIMR. In these implementations, a GPIER bit can be set even though the associated interrupt is masked. See The "GPIO module as implemented on the chip" section for more information.

16.5.5.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	EVn															
W	W1C															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	EVn															
W	W1C															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

16.5.5.4 Fields

Field	Function
0-31 EVn	Interrupt events. Indicates whether an interrupt event occurred on the corresponding GPIO signal. 00000000000000000000000000000000b - No interrupt event occurred on the corresponding GPIO signal. 00000000000000000000000000000001b - An interrupt event occurred on the corresponding GPIO signal.

16.5.6 GPIO interrupt mask register (GPIMR)

16.5.6.1 Offset

Register	Offset
GPIMR	10h

16.5.6.2 Function

The GPIO interrupt mask register (GPIMR) defines the interrupt masking for the individual ports. When a masked interrupt request occurs, the corresponding GPIER bit is set, regardless of the GPIMR state. When one or more non-masked interrupt events occur, the GPIO module issues an interrupt to the interrupt controller.

16.5.6.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	IMn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	IMn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.6.4 Fields

Field	Function
0-31 IMn	Interrupt mask. Indicates whether an interrupt event is masked or not masked for the corresponding GPIO signal. 00000000000000000000000000000000b - The input interrupt signal is masked (disabled). 00000000000000000000000000000001b - The input interrupt signal is not masked (enabled).

16.5.7 GPIO interrupt control register (GPICR)

16.5.7.1 Offset

Register	Offset
GPICR	14h

16.5.7.2 Function

The GPIO interrupt control register (GPICR) determines whether the corresponding port line asserts an interrupt request upon either a high-to-low change or any change on the state of the signal.

16.5.7.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	EDn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	EDn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.7.4 Fields

Field	Function
0-31 EDn	Edge detection mode. The corresponding port line asserts an interrupt request according to the following: 00000000000000000000000000000000b - Any change on the state of the port generates an interrupt request. 00000000000000000000000000000001b - High-to-low change on the port generates an interrupt request.

16.5.8 GPIO Input Buffer Enable Register (GPIBE)

16.5.8.1 Offset

Register	Offset
GPIBE	18h

16.5.8.2 Function

The GPIO Input Buffer Enable register is used to control the input enable of each individual GPIO port. When an individual GPIO port's direction is set to input (GPIO_GPDIR[DRn=0]), the associated input enable must be set (GPIOxGPIE[IEn]=1) to propagate the port value to the GPIO Data Register. When a port's input enable is disabled, the port value does not propagate to the GPIO Data Register. Unused GPIO

ports may be non-connected on the board if their “direction” and “input enable” are configured for “input” (GPIO_GPDIR[DRn=0]) and “disabled” (GPIBE[IEn=0]) respectively, which is the default configuration (based on GPIO input signal GPIBE_RESET_VAL[31:0] tied to 0x0000_0000 which sets the reset value of this register).

16.5.8.3 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	IBEn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	IBEn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16.5.8.4 Fields

Field	Function
0-31 IBEn	Input Enable. Indicates whether a GPIO port’s input is enabled. 00000000000000000000000000000000b - The corresponding port input is disabled. 00000000000000000000000000000001b - The corresponding port input is enabled.

Chapter 17
Integrated Flash Controller (IFC)

17.1 IFC overview

The integrated flash controller (IFC) is used to interface with external asynchronous/ synchronous NAND flash, asynchronous NOR flash, SRAM, generic ASIC memory and EPROM.

It has eight chip-selects, to which a maximum of eight flash devices can be attached, although only one of these can be accessed at any given time.

The IFC handles pin multiplexing to the internal system bus based on the selected controller (NAND, NOR, GPCM, or generic ASIC). To save pins at the chip level, multiplexing of address pins can be done on the data bus using an external latch (through control signal ALE). The BCH error-correction algorithm is used to correct the error bits while reading from a NAND device.

This figure shows the block diagram of IFC.

Figure 17-1. IFC block diagram

17.1.1 IFC features summary

The IFC supports both general and controller-specific features.

The general features of the IFC include the following:

- Flash controller with eight chip-selects
- Supports error and debug registers
- Functional muxing of pins between NAND, NOR, and GPCM
- Supports memory banks of sizes up to 4 GB (for NOR and GPCM)
- Write-protection capability (for NAND and NOR)
- Provision of software reset
- External transceiver enable/disable control on a per-bank basis

17.1.1.1 NAND flash controller features

The IFC NAND flash controller features are:

- x8 NAND flash interface
- Support for ONFI-2.2 asynchronous interface (8-/16-bit)/NVDDR interface and mandatory commands
- BCH code for 4-bit and 8-bit error correction per sector of 512 bytes (using GF-2^{13} Galois field) and 24 and 40-bit ECC per sector of 1 KB (using GF- 2^{14} Galois field):
  - For a 512-byte page and 16-byte spare region, 4-bit error correction is supported
  - For a 2 KB page and 64-byte spare region, 4-bit error correction is supported
  - For a 4 KB page width:
    - For 128-byte spare region, 4-bit error correction is supported
    - For 210-, 218-, and 224-byte spare regions, 4- or 8-bit error correction is supported
    - Spare size of minimum 176 bytes (8 for BBI + 4x42 for ECC bytes), 4-, 8-, or 24-bit BCH
    - Spare size of minimum 288 bytes (8 for BBI + 4x70 for ECC bytes), 4-, 8-, 24-, or 40-bit BCH
  - For an 8 KB page width:
    - At least 136(8 + 128) bytes of spare region is required for bad block information and ECC bytes, 4-bit BCH
    - 8 KB page, at least 264(8 + 256) bytes of spare region required for bad block information and ECC bytes, 4-bit or 8-bit BCH
    - Spare size of minimum 344 bytes (8 for BBI + 8x42 for ECC bytes), 4-, 8-, or 24-bit BCH
    - Spare size of minimum 568 bytes (8 for BBI + 8x70 for ECC bytes), 4-, 8-, 24-, or 40-bit BCH
  - 16k byte page:
    - At least 264 (8+32x8) bytes of spare region required for bad block information and ecc bytes, 4-bit BCH

    - At least 520 (8+32x16) bytes of spare region required for bad block information and ecc bytes, 4bit or 8 bit BCH
    - Spare size of minimum of 680 bytes (8 for BBI + 16x42 for ECC bytes), 4, 8 or 24 bit BCH
    - Spare size of minimum of 1128 bytes (8 for BBI + 16x70 for ECC bytes), 4, 8, 24 or 40 bit BCH
  - For all ECC modes (4/8/24/40), parity bytes are stored at offset 08h in a spare region
- ECC generation/checking is optional
- Flexible timing control to allow interfacing with proprietary NAND devices
- Supports SLC and MLC flash devices with configurable page sizes of up to 16 KB
- Supports advance NAND commands such as cache, copy-back, and multi-plane programming
- Supports four RDY_B/BSY_B signals
- Programmable command and data transfer sequences of up to 15 steps
- Configurable block-size
- Interrupt for error handling and flash command completion event
- Internal SRAM of 18 KB
  - Memory-mapped
  - Can be configured as boot RAM
  - Acts as data buffer for normal operation
- Boot chip-select (CS0) available after system reset, with a boot block size of 16 KB for execute-in-place boot loading from NAND flash
- Supports flash devices of a magnitude of terabytes
  - By using 32 bit row address, 4 Giga pages of NAND flash can be accessed

17.1.1.2 NOR flash features

The IFC features a NOR flash controller.

- Compatible with asynchronous/synchronous NOR flash
- Provides memory-mapped interfacing to NOR
- Supports an address data multiplexed (ADM) NOR device
- Flexible timing control allows interfacing with proprietary NOR devices (WE_B controlled writes only)
- Boot chip-select (CS0) is available at system reset
- Data bus width of 16-bits
- Synchronous NOR features:
  - Programmable burst size control (continuous and split mode)
  - Support for programmable initial access latency
  - Data is qualified with NRRB_B input

17.1.1.3 GPCM and GASIC features

The IFC's general-purpose, chip-select Controller (GPCM) supports operation in either normal or generic ASIC modes.

Normal GPCM features include the following:

- Support for x8-/16-bit devices
- Compatible with general-purpose addressable device, such as SRAM and ROM
- External clock is generated with programmable division ratio (2, 3, 4, ... up to 16)
- Output enable signal (OE_B)
- Write enable signal(s) (WE_B)
- Even/odd parity data bus support
- External access termination signal (IFCTA_B)
- Programmable burst support

Generic ASIC features include the following:

- Support for x8-/16-bit devices
- Address and data are shared on I/O bus
- The following address and data sequences are supported on the I/O bus:
  - 16-bit I/O: AADD
  - 8-bit I/O: AAAADDDD
- Configurable even/odd parity on address/data bus
- Parity-error detection

17.1.2 IFC modes of operation

The IFC contains one NAND controller, one NOR flash controller, and one GPCM/generic-ASIC controller.

It can be programmed such that all eight memory banks can work with NAND, NOR, and GPCM/generic-ASIC interfaces as required. However, only one memory bank can be active at any given time.

17.2 External signal descriptions

This section provides both general and detailed information on the chip's external signals.

This table provides an overview of the chip's external signals, and the following table then provides a much more detailed description of the signals.

Table 17-1. Signal properties

Chip signal name	IFC signal name	Alternate Functions	Mode	Description / Function	Number of signals	I/O
IFC_AD[15:0]	AD[15:0]	AD[15:0]	NAND FCM	I/O Bus for Asynchronous NAND	16	I/O
		AD[7:0]	NVDDR	I/O Bus for NVDDR sync NAND		
		AD[15:0]	NOR FCM	Bidirectional Data Bus for NOR. During AVD assertion, this bus will carry address bits. Address LSB or MSB will be governed by CSOR[ADM_SHFT_MODE]		
		AD[15:0]	GPCM	Bidirectional data bus for GPCM. For external address latching, this bus is used to carry address MSBs.		
		AD[15:0]	GASIC	Bidirectional shared address/ data bus for GASIC.		
IFC_A[11:0]	ADDR[11:0]	-	-	Dedicated address bus used by NOR and GPCM mode	12	O
IFC_PAR[1:0]	IFC_PAR[1:0]	IFC_PAR[1:0]	GPCM	Parity data	2	I/O
		IFC_PAR[1:0]	GASIC	Parity address and data		
IFC_NDDDR_CLK	IFC_NDDDR_CLK	IFC_NDDDR_CLK	NV-DDR	External IFC DDR Clock (NVDDR mode)	1	O
		NDWE_B	Async NAND	NAND Write Enable	1	O
IFC_CS[7:0]_B	CE/CS[7:0]_B	-	-	Chip Selects	8	O
NDWE_B	Async	NAND Write Enable	-			
IFC_BCTL	BCTL	-		Data buffer control	1	O

Table continues on the next page...

Table 17-1. Signal properties (continued)

Chip signal name	IFC signal name	Alternate Functions	Mode	Description / Function	Number of signals	I/O
IFC_TE	TE	-	-	External Transceiver Enable/Disable Control	1	I/O
IFC_NDDQS	NDDQS	-	NVDDR	Data Qualify Strobe to/from NAND memory (in NVDDR)	1	I/O
IFC_AVD	ALE/AVD	NDALE	NAND FCM	NAND Address Latch Enable	1	O
		NRALE	NOR FCM	External Latch Enable control for unmuxed (parallel) NOR		
		GPALE	GPCM	GPCM External Address Latch Enable		
IFC_CLE	CLE	NDCLE	NAND FCM	NAND Command Latch Enable	1	O
		GPWE1_B	GPCM	GPCM Write Byte Select 1		
		NRAVD_B	NOR FCM	Active low Address Valid Signal for synchronous burst NOR		
IFC_OE_B	OE_B	NROE_B	NOR FCM	NOR Output Enable	1	O
		GPOE_B	GPCM	GPCM Output Enable		
		RW_L_B	GASIC	GASIC Read/ Write Indication		
		NDRE_B	Async/NVDDR	NAND Read Enable		
IFC_WP[3:0]_B	WP[3:0]_B	NDWP[3:0]_B	NAND FCM	Per chip select based NAND Write Protect	4	O
IFC_RB[3:0]_B	RB[3:0]_B	NAND FCM	NAND Ready/ Busy Input	4	I	
		NRRB[3:0]_B	NOR FCM			NOR Ready Busy Input
		IFCTA[3:0]_B	GPCM			GPCM External Termina tion

Table continues on the next page...

Table 17-1. Signal properties (continued)

Chip signal name	IFC signal name	Alternate Functions	Mode	Description / Function	Number of signals	I/O
		RDY_L[3:0]_B	GASIC			GASIC Ready Indicatio n
IFC_PERR_B	PERR_L_B	PERR_L_B	GASIC	Parity error from Generic ASIC	1	I
		NAND_ENI_B	NAND	NAND volume assignment status for ONFI devices supporting volume addressing		
IFC_CLK[1:0]	IFC_CLK[1:0]	-	-	External IFC clock	2	O

This table describes the external signals in detail.

Table 17-2. Detailed signal description

Signal	I/O	Description	
AD[0:*n*]	I/O	Multiplexed address/data bus.	
		State meaning	Asserted/Negated - During the assertion of ALE (external latch enable control) or AVD_B (for sync NOR), AD[*n*:0] are driven with the address for the access to follow. External logic should propagate the address on AD[*n*:0] . After it is negated, AD[*n*:0] are either driven by write data or are switched to an input to sample read data driven by an external device. Following the last data transfer of a write access, AD[*n*:0] are released to a high-impedance state. For exact functionality, behavior, and timings, refer to detailed waveform and timing diagrams within the specific interface.
ADDR[*n*:*m*]	O	Non-multiplexed address bus.	
		State meaning	Asserted/Negated - It is a non-multiplexed address bus which carries address MSBs or LSBs depending on CSOR[ADM_SHFT_MODE].
ALE	O	Address latch enable for NAND. External latch for unmuxed (parallel) NOR/ GPCM (control signal for an external address latch, which allows address and data to be multiplexed on the device pins)	
		State meaning	Asserted - External Latch Enable is asserted when AD bus carries Address bits. The number of cycles for which it is asserted is governed by the FTIM0_CSn[EADC] field. Negated - Address gets latched on external latch on the falling edge of ALE and address hold value is governed by FTIM0_CSn[EAHC] field.
AVD_B	O	Active low address valid signal for the address latch in NOR devices.	
		State meaning	Timing is same as AVD signal except opposite polarity.
CS[0:*n*]_B	O	Chip-Select; mutually exclusive chip-selects are available.	

Table continues on the next page...

Table 17-2. Detailed signal description (continued)

Signal	I/O	Description	
		State meaning	Asserted/Negated - Used to enable each specific memory device connected to the IFC. CS_B[0] corresponds to the chip-select for memory bank 0, which has the memory type and attributes defined by CSPR0 and FTIM0_CS0.
WE_B	O	NOR write enable, GPCM write byte select 0, GASIC start of frame (SOF_L_B).	
		State meaning	Asserted/Negated - Used to control the data write cycles on NOR flash. For GPCM it validates its corresponding byte number on the data bus. For GASIC it indicates start of a transaction frame.
NDDQS	I/O	Bi-directional Data Qualify Strobe for NAND in NV-DDR mode	
		State Meaning	Bi-directional signal used to capture write data in the NAND flash and read data in IFC. Data is captured/driven on both of its edges (rising and falling). For program operations, the DQS is centred with respect to program data when driven by IFC. For read operations, the DQS is edge-aligned with respect to read data when driven by NAND memory.
CLE	O	NAND Command latch enable, GPCM write byte select 1 which is WE[1]_B. Active low NRAVD is used by synchronous NOR device (Unmuxed as well as muxed)	
		State meaning	Asserted/Negated - It enables the command cycle on NAND flash. For GPCM it validates 1st byte(AD [15:8]) on data bus. Active Llow address valid signal for synchronous NOR Devices (Unmuxed as well as Muxed). Address value is latched in NOR device on the rising edge of IFC_CLK when NRAVD is asserted (low)
OE_B	O	NAND read enable, NOR output enable, GPCM output enable, GASIC Read/Write Enable and is valid only when SOF_L_B is asserted.	
		State meaning	Asserted/Negated - It enables data read cycles on NOR and GPCM devices. For GASIC, it indicates whether the transaction is read or write. High value on this indicates read operation while low value indicates write.
WP[*n*:0]_B	O	Per chip select based non-muxed NAND Write Protect Output. Note that if chip has multiplexed single WP_B pin to different chip selects connecting different NAND devices, then same value must be programmed in corresponding CSPRn[WP] bit field sharing same WP_B pin. It is required as NAND device expects TWW wait time for any toggling in WP_B pin.	
		State meaning	Asserted/Negated - It protects NAND flash from accidental erasure or program when asserted low. For GPCM it validates the corresponding byte number on data bus.
RB[*n*:0]_B	I	NAND ready busy for all the chip-selects, NOR Read Busy, GPCM External Termination of Access, GASIC Ready Indication. Valid after the completion of address phase till it asserts.	
		State meaning	Asserted/Negated - It indicates the ready busy status of the flash operation. It is used to stall the controller operation when the flash is indicating busy. For GPCM this input is used for termination of current access. For GASIC it indicates that the current GASIC transaction is accepted by device and the host can start the next transaction.
BCTL	O	Data buffer control. Buffer control is disabled by FTIM0_CS0[BCTLD] field.	
		State meaning	Asserted/Negated - The BCTL pin normally functions as a write (High)/read (Low) direction control for the external data buffer used in the AD lines (bidirectional bus). Default value of BCTL is high (write direction).

Table continues on the next page...

Table 17-2. Detailed signal description (continued)

Signal	I/O	Description	
TE	I/O	External transceiver enable/disable. This pin acts as an input during reset and should be weakly pulled up/down so as to disable the external transceiver during that time. After reset, the IFC drives TE with the value configured in CSPRn[TE] as per the selected chip select. When none of the CS pins are selected, TE is driven High-Z from the IFC (the external transceiver will be disabled by the appropriate external pull device).	
		State meaning	Asserted/Negated - The TE pin functions as an enable/disable for an external transceiver connected to the AD lines. In asserted state, it enables the external transceiver and when negated, disables the transceiver.
PAR[*n*:0]	I/O	GPCM data bus parity, GASIC address bus parity.	
		State meaning	Asserted/Negated - For GPCM during Write a parity bit is generated for each data byte sent over the bus. Unused byte lanes have undefined parity. For GASIC a parity bit is sent for each address and data byte sent over the AD bus.
IFC_NDDDR_CLK/NDWE_B	O	IFC NAND DDR External Clock for NV-DDR mode NAND Write Enable for ASync	
		State Meaning	NV-DDR mode: External divided clock derived from IP clock for the NAND Flash. ASync Used to control write cycles for NAND Flash
CLK[*n*:0]	O	IFC external clock.	
		State meaning	This is an external clock derived from the IFC module input clock. Division ratio is programmable (using CCR register). This clock is active only when there is any access on GPCM or synchronous NOR.

17.2.1 Internal connectivity of WP/RB signal

The IFC supports one write protect and one ready/busy signal per chip select. The available write protect and ready/busy signals are mapped to the eight chip selects that the IFC supports.

The chip supports one dedicated write protect signal (IFC_WP0_B), three multiplexed write protect pins (IFC_WP[1:3]_B), one dedicated ready/busy signals (IFC_RB0_B) and three multiplexed ready/busy signals IFC_RB[1:3]_B .

NOTE

The internal connectivity of WP/RB signal depends on the IFC mode selected by RCW[IFC_MODE].

17.2.1.1 Internal connectivity of WP signal

IFC has support of 1 write protect signal per chip select.

For the 22-bit address mode, the first three write protect signals WP_B[0:2] are directly sourced by the three write protect signals from the IFC block which correspond to chip selects 0-2, respectively. The fourth write protect signal (WP_B[3]) is sourced by the logical OR of the remaining five write protect signals from the IFC block which correspond to chip selects 3-7. This allows for some flexibility in controlling write protection to individual or groups of NAND flash devices.

Figure 17-2. Internal connectivity of WP signal for 22-bit address mode

NOTE

For 25-bit and 28-bit address mode, only WP0 is available.
WP0 is the logical OR of WP[0:7].

17.2.1.2 Internal connectivity of RB signal

IFC has support of 1 ready/busy per chip select. The first 2 ready/busy signals (RB_B[0:1]) are directly routed to the ready/busy inputs of the IFC block which correspond to chip selects 0-1. RB_B[2:4] is configurable. When RCW[449:448] (IFC_GRP_A_BASE) is 2'b10, RB_B[2:4] are directly routed to ready/busy inputs of the IFC block which corresponds to chip selects 2-4. For all other value of RCW[449:448], ready/busy signals of chip selects 2-4 are connected to RB_B[0]. RB_B[4] is also routed to the ready/busy inputs of the IFC block which correspond to chip selects 5-7.

Figure 17-3. Internal connectivity of RB signal-bit address mode

17.3 IFC register descriptions

The IFC is allocated 8 KB of memory-mapped space. The memory map is divided into two parts (4 KB each):

- Common registers shared by NAND, NOR, and GPCM FCM
- Specific registers defined for the NAND FCM, NOR FCM (IFC global), and GPCM FCM (IFC run time) exclusively

17.3.1 IFC memory map

IFC base address: 224_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	IFC Revision Control register (REV)	32	RO	Table 17-2
Ch	Extended Chip Select Property Registers (CSPR0_EXT)	32	RW	0000_0000h
10h	Chip-select Property register n (CSPR0)	32	RW	0000_0000h
18h	Extended Chip Select Property Registers (CSPR1_EXT)	32	RW	0000_0000h
1Ch	Chip-select Property register n (CSPR1)	32	RW	0000_0000h
24h	Extended Chip Select Property Registers (CSPR2_EXT)	32	RW	0000_0000h
28h	Chip-select Property register n (CSPR2)	32	RW	0000_0000h
30h	Extended Chip Select Property Registers (CSPR3_EXT)	32	RW	0000_0000h
34h	Chip-select Property register n (CSPR3)	32	RW	0000_0000h
3Ch	Extended Chip Select Property Registers (CSPR4_EXT)	32	RW	0000_0000h
40h	Chip-select Property register n (CSPR4)	32	RW	0000_0000h
48h	Extended Chip Select Property Registers (CSPR5_EXT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4Ch	Chip-select Property register n (CSPR5)	32	RW	0000_0000h
54h	Extended Chip Select Property Registers (CSPR6_EXT)	32	RW	0000_0000h
58h	Chip-select Property register n (CSPR6)	32	RW	0000_0000h
60h	Extended Chip Select Property Registers (CSPR7_EXT)	32	RW	0000_0000h
64h	Chip-select Property register n (CSPR7)	32	RW	0000_0000h
A0h	Address Mask register (AMASK0)	32	RW	0000_0000h
ACh	Address Mask register (AMASK1)	32	RW	0000_0000h
B8h	Address Mask register (AMASK2)	32	RW	0000_0000h
C4h	Address Mask register (AMASK3)	32	RW	0000_0000h
D0h	Address Mask register (AMASK4)	32	RW	0000_0000h
DCh	Address Mask register (AMASK5)	32	RW	0000_0000h
E8h	Address Mask register (AMASK6)	32	RW	0000_0000h
F4h	Address Mask register (AMASK7)	32	RW	0000_0000h
130h	Chip-Select Option register - GPCM (CSOR0_GPCM)	32	RW	0000_000Ch
130h	Chip-Select Option register - NAND Flash Mode (CSOR0_NAND)	32	RW	0000_000Ch
130h	Chip-Select Option register - NOR Flash Mode (CSOR0_NOR)	32	RW	0000_000Ch
134h	Extended Chip-Select Option register - NAND Flash Mode (CSOR0_EXT)	32	RW	0010_0000h
13Ch	Chip-Select Option register - GPCM (CSOR1_GPCM)	32	RW	0000_000Ch
13Ch	Chip-Select Option register - NAND Flash Mode (CSOR1_NAND)	32	RW	0000_000Ch
13Ch	Chip-Select Option register - NOR Flash Mode (CSOR1_NOR)	32	RW	0000_000Ch
140h	Extended Chip-Select Option register - NAND Flash Mode (CSOR1_EXT)	32	RW	0010_0000h
148h	Chip-Select Option register - GPCM (CSOR2_GPCM)	32	RW	0000_000Ch
148h	Chip-Select Option register - NAND Flash Mode (CSOR2_NAND)	32	RW	0000_000Ch
148h	Chip-Select Option register - NOR Flash Mode (CSOR2_NOR)	32	RW	0000_000Ch
14Ch	Extended Chip-Select Option register - NAND Flash Mode (CSOR2_EXT)	32	RW	0010_0000h
154h	Chip-Select Option register - GPCM (CSOR3_GPCM)	32	RW	0000_000Ch
154h	Chip-Select Option register - NAND Flash Mode (CSOR3_NAND)	32	RW	0000_000Ch
154h	Chip-Select Option register - NOR Flash Mode (CSOR3_NOR)	32	RW	0000_000Ch
158h	Extended Chip-Select Option register - NAND Flash Mode (CSOR3_EXT)	32	RW	0010_0000h
160h	Chip-Select Option register - GPCM (CSOR4_GPCM)	32	RW	0000_000Ch
160h	Chip-Select Option register - NAND Flash Mode (CSOR4_NAND)	32	RW	0000_000Ch
160h	Chip-Select Option register - NOR Flash Mode (CSOR4_NOR)	32	RW	0000_000Ch
164h	Extended Chip-Select Option register - NAND Flash Mode (CSOR4_EXT)	32	RW	0010_0000h
16Ch	Chip-Select Option register - GPCM (CSOR5_GPCM)	32	RW	0000_000Ch
16Ch	Chip-Select Option register - NAND Flash Mode (CSOR5_NAND)	32	RW	0000_000Ch
16Ch	Chip-Select Option register - NOR Flash Mode (CSOR5_NOR)	32	RW	0000_000Ch

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
170h	Extended Chip-Select Option register - NAND Flash Mode (CSOR5_EXT)	32	RW	0010_0000h
178h	Chip-Select Option register - GPCM (CSOR6_GPCM)	32	RW	0000_000Ch
178h	Chip-Select Option register - NAND Flash Mode (CSOR6_NAND)	32	RW	0000_000Ch
178h	Chip-Select Option register - NOR Flash Mode (CSOR6_NOR)	32	RW	0000_000Ch
17Ch	Extended Chip-Select Option register - NAND Flash Mode (CSOR6_EXT)	32	RW	0010_0000h
184h	Chip-Select Option register - GPCM (CSOR7_GPCM)	32	RW	0000_000Ch
184h	Chip-Select Option register - NAND Flash Mode (CSOR7_NAND)	32	RW	0000_000Ch
184h	Chip-Select Option register - NOR Flash Mode (CSOR7_NOR)	32	RW	0000_000Ch
188h	Extended Chip-Select Option register - NAND Flash Mode (CSOR7_EXT)	32	RW	0010_0000h
1C0h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS0_GPCM)	32	RW	0000_0000h
1C0h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS0_NAND)	32	RW	0000_0000h
1C0h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS0_NAND_ASYNC_MODE)	32	RW	0000_0000h
1C0h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS0_NOR)	32	RW	0000_0000h
1C4h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS0_GPCM)	32	RW	0000_0000h
1C4h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS0_NAND)	32	RW	0000_0000h
1C4h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS0_NAND_ASYNC_MODE)	32	RW	0000_0000h
1C4h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS0_NOR)	32	RW	0000_0000h
1C8h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS0_GPCM)	32	RW	0000_0000h
1C8h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS0_NAND)	32	RW	0000_0000h
1C8h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS0_NAND_ASYNC_MODE)	32	RW	0000_0000h
1C8h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS0_NOR)	32	RW	0000_0000h
1CCh	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS0_GPCM)	32	RW	0000_0000h
1CCh	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS0_NAND)	32	RW	0000_0000h
1CCh	Flash Timing register 3 for CSn - NOR Flash Mode (FTIM3_CS_NOR)	32	RW	0000_0000h
1CCh	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM03_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1F0h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS1_GPCM)	32	RW	0000_0000h
1F0h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS1_NAND)	32	RW	0000_0000h
1F0h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS1_NAND_ASYNC_MODE)	32	RW	0000_0000h
1F0h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS1_NOR)	32	RW	0000_0000h
1F4h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS1_GPCM)	32	RW	0000_0000h
1F4h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS1_NAND)	32	RW	0000_0000h
1F4h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS1_NAND_ASYNC_MODE)	32	RW	0000_0000h
1F4h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS1_NOR)	32	RW	0000_0000h
1F8h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS1_GPCM)	32	RW	0000_0000h
1F8h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS1_NAND)	32	RW	0000_0000h
1F8h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS1_NAND_ASYNC_MODE)	32	RW	0000_0000h
1F8h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS1_NOR)	32	RW	0000_0000h
1FCh	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS1_GPCM)	32	RW	0000_0000h
1FCh	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS1_NAND)	32	RW	0000_0000h
1FCh	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM13_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
220h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS2_GPCM)	32	RW	0000_0000h
220h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS2_NAND)	32	RW	0000_0000h
220h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS2_NAND_ASYNC_MODE)	32	RW	0000_0000h
220h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS2_NOR)	32	RW	0000_0000h
224h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS2_GPCM)	32	RW	0000_0000h
224h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS2_NAND)	32	RW	0000_0000h
224h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS2_NAND_ASYNC_MODE)	32	RW	0000_0000h
224h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS2_NOR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
228h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS2_GPCM)	32	RW	0000_0000h
228h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS2_NAND)	32	RW	0000_0000h
228h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS2_NAND_ASYNC_MODE)	32	RW	0000_0000h
228h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS2_NOR)	32	RW	0000_0000h
22Ch	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS2_GPCM)	32	RW	0000_0000h
22Ch	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS2_NAND)	32	RW	0000_0000h
22Ch	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM23_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
250h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS3_GPCM)	32	RW	0000_0000h
250h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS3_NAND)	32	RW	0000_0000h
250h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS3_NAND_ASYNC_MODE)	32	RW	0000_0000h
250h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS3_NOR)	32	RW	0000_0000h
254h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS3_GPCM)	32	RW	0000_0000h
254h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS3_NAND)	32	RW	0000_0000h
254h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS3_NAND_ASYNC_MODE)	32	RW	0000_0000h
254h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS3_NOR)	32	RW	0000_0000h
258h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS3_GPCM)	32	RW	0000_0000h
258h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS3_NAND)	32	RW	0000_0000h
258h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS3_NAND_ASYNC_MODE)	32	RW	0000_0000h
258h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS3_NOR)	32	RW	0000_0000h
25Ch	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS3_GPCM)	32	RW	0000_0000h
25Ch	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS3_NAND)	32	RW	0000_0000h
25Ch	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM33_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
280h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS4_GPCM)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
280h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS4_NAND)	32	RW	0000_0000h
280h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS4_NAND_ASYNC_MODE)	32	RW	0000_0000h
280h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS4_NOR)	32	RW	0000_0000h
284h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS4_GPCM)	32	RW	0000_0000h
284h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS4_NAND)	32	RW	0000_0000h
284h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS4_NAND_ASYNC_MODE)	32	RW	0000_0000h
284h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS4_NOR)	32	RW	0000_0000h
288h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS4_GPCM)	32	RW	0000_0000h
288h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS4_NAND)	32	RW	0000_0000h
288h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS4_NAND_ASYNC_MODE)	32	RW	0000_0000h
288h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS4_NOR)	32	RW	0000_0000h
28Ch	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS4_GPCM)	32	RW	0000_0000h
28Ch	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS4_NAND)	32	RW	0000_0000h
28Ch	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM43_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
2B0h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS5_GPCM)	32	RW	0000_0000h
2B0h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS5_NAND)	32	RW	0000_0000h
2B0h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS5_NAND_ASYNC_MODE)	32	RW	0000_0000h
2B0h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS5_NOR)	32	RW	0000_0000h
2B4h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS5_GPCM)	32	RW	0000_0000h
2B4h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS5_NAND)	32	RW	0000_0000h
2B4h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS5_NAND_ASYNC_MODE)	32	RW	0000_0000h
2B4h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS5_NOR)	32	RW	0000_0000h
2B8h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS5_GPCM)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2B8h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS5_NAND)	32	RW	0000_0000h
2B8h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS5_NAND_ASYNC_MODE)	32	RW	0000_0000h
2B8h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS5_NOR)	32	RW	0000_0000h
2BCh	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS5_GPCM)	32	RW	0000_0000h
2BCh	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS5_NAND)	32	RW	0000_0000h
2BCh	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM53_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
2E0h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS6_GPCM)	32	RW	0000_0000h
2E0h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS6_NAND)	32	RW	0000_0000h
2E0h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS6_NAND_ASYNC_MODE)	32	RW	0000_0000h
2E0h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS6_NOR)	32	RW	0000_0000h
2E4h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS6_GPCM)	32	RW	0000_0000h
2E4h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS6_NAND)	32	RW	0000_0000h
2E4h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS6_NAND_ASYNC_MODE)	32	RW	0000_0000h
2E4h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS6_NOR)	32	RW	0000_0000h
2E8h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS6_GPCM)	32	RW	0000_0000h
2E8h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS6_NAND)	32	RW	0000_0000h
2E8h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS6_NAND_ASYNC_MODE)	32	RW	0000_0000h
2E8h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS6_NOR)	32	RW	0000_0000h
2ECh	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS6_GPCM)	32	RW	0000_0000h
2ECh	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS6_NAND)	32	RW	0000_0000h
2ECh	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM63_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
310h	Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS7_GPCM)	32	RW	0000_0000h
310h	Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS7_NAND)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
310h	Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS7_NAND_ASYNC_MODE)	32	RW	0000_0000h
310h	Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS7_NOR)	32	RW	0000_0000h
314h	Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS7_GPCM)	32	RW	0000_0000h
314h	Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS7_NAND)	32	RW	0000_0000h
314h	Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS7_NAND_ASYNC_MODE)	32	RW	0000_0000h
314h	Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS7_NOR)	32	RW	0000_0000h
318h	Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS7_GPCM)	32	RW	0000_0000h
318h	Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS7_NAND)	32	RW	0000_0000h
318h	Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS7_NAND_ASYNC_MODE)	32	RW	0000_0000h
318h	Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS7_NOR)	32	RW	0000_0000h
31Ch	Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS7_GPCM)	32	RW	0000_0000h
31Ch	Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS7_NAND)	32	RW	0000_0000h
31Ch	Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM73_CS_NAND_ASYNC_MODE)	32	RW	0000_0000h
400h	Ready Busy Status for each Chip Select (RB_STAT)	32	RO	Table 17-4
404h	Ready Busy Map register (RB_MAP)	32	RW	Table 17-4
408h	Write Protect Map register (WP_MAP)	32	RW	Table 17-4
40Ch	General Control register (GCR)	32	RW	0000_0000h
418h	Common Event and Error Status register (CM_EVTER_STAT)	32	W1C	0000_0000h
424h	Common Event and Error Enable register (CM_EVTER_EN)	32	RW	8000_0000h
430h	Common Event and Error Interrupt Enable register (CM_EVTER_INTR_EN)	32	RW	0000_0000h
43Ch	Common Transfer Error Attributes register 0 (CM_ERATTR0)	32	RO	0000_0000h
440h	Common Transfer Error Attributes register 1 (CM_ERATTR1)	32	RO	0000_0000h
44Ch	Clock Control register (CCR)	32	RW	0300_8000h
450h	Clock Status register (CSR)	32	RO	Table 17-4
454h	DDR Clock Control register Low (DDR_CCR_LOW)	32	RW	0080_0000h
1_0000h	NAND Configuration register (NCFGR)	32	RW	0000_0000h
1_0014h	NAND Flash Command register 0 (NAND_FCR0)	32	RW	0000_0000h
1_0018h	NAND Flash Command register 1 (NAND_FCR1)	32	RW	0000_0000h
1_003Ch	Flash Row Address register n (ROW0)	32	RW	0000_0000h
1_0044h	Flash COL Address register n (COL0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1_0044h	Flash COL Address register for 2 KB Large-Page Device (COL0_2KB)	32	RW	0000_0000h
1_0044h	Flash COL Address register for 4 KB Large-Page Device (COL0_4KB)	32	RW	0000_0000h
1_004Ch	Flash Row Address register n (ROW1)	32	RW	0000_0000h
1_0054h	Flash COL Address register n (COL1)	32	RW	0000_0000h
1_0054h	Flash COL Address register for 2 KB Large-Page Device (COL1_2KB)	32	RW	0000_0000h
1_0054h	Flash COL Address register for 4 KB Large-Page Device (COL1_4KB)	32	RW	0000_0000h
1_005Ch	Flash Row Address register n (ROW2)	32	RW	0000_0000h
1_0064h	Flash COL Address register n (COL2)	32	RW	0000_0000h
1_0064h	Flash COL Address register for 2 KB Large-Page Device (COL2_2KB)	32	RW	0000_0000h
1_0064h	Flash COL Address register for 4 KB Large-Page Device (COL2_4KB)	32	RW	0000_0000h
1_006Ch	Flash Row Address register n (ROW3)	32	RW	0000_0000h
1_0074h	Flash COL Address register n (COL3)	32	RW	0000_0000h
1_0074h	Flash COL Address register for 2 KB Large-Page Device (COL3_2KB)	32	RW	0000_0000h
1_0074h	Flash COL Address register for 4 KB Large-Page Device (COL3_4KB)	32	RW	0000_0000h
1_0108h	Flash Byte Count register for NAND Flash (NAND_BC)	32	RW	0000_0000h
1_0110h	NAND Flash Instruction register 0 (NAND_FIR0)	32	RW	0000_0000h
1_0114h	NAND Flash Instruction register 1 (NAND_FIR1)	32	RW	0000_0000h
1_0118h	NAND Flash Instruction register 2 (NAND_FIR2)	32	RW	0000_0000h
1_015Ch	NAND Chip-Select register (NAND_CSEL)	32	RW	0000_0000h
1_0164h	NAND Operation Sequence Start (NANDSEQ_STRT)	32	RW	0000_0000h
1_016Ch	NAND Event and Error Status register (NAND_EVTER_STAT)	32	W1C	0000_0000h
1_0174h	NAND Page Read Completion Event Status register (PGRDCMPL_EVT_STAT)	32	W1C	0000_0000h
1_0180h	NAND Event and Error Enable register (NAND_EVTER_EN)	32	RW	AE00_0000h
1_018Ch	NAND Event and Error Interrupt Enable register (NAND_EVTER_INTR_EN)	32	RW	0000_0000h
1_0198h	NAND Transfer Error Attributes register 0 (NAND_ERATTR0)	32	RO	0000_0000h
1_019Ch	NAND Transfer Error Attributes register 1 (NAND_ERATTR1)	32	RO	0000_0000h
1_01E0h	NAND Flash Status register (NAND_FSR)	32	RO	0000_0000h
1_01E8h	ECC Status and Result of Flash Operation register 0 (ECCSTAT0)	32	RO	0000_0000h
1_01ECh	ECC Status and Result of Flash Operation register 1 (ECCSTAT1)	32	RO	0000_0000h
1_01F0h	ECC Status and Result of Flash Operation Register2 (ECCSTAT2)	32	RO	0000_0000h
1_01F4h	ECC Status and Result of Flash Operation Register3 (ECCSTAT3)	32	RO	0000_0000h
1_01F8h	ECC Status and Result of Flash Operation Register4 (ECCSTAT4)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1_01FCh	ECC Status and Result of Flash Operation Register5 (ECCSTAT5)	32	RO	0000_0000h
1_0200h	ECC Status and Result of Flash Operation Register6 (ECCSTAT6)	32	RO	0000_0000h
1_0204h	ECC Status and Result of Flash Operation Register7 (ECCSTAT7)	32	RO	0000_0000h
1_0278h	NAND Control register (NANDCR)	32	RW	1E00_0000h
1_0284h	NAND Autoboot Trigger register (NAND_AUTOBOOT_TRGR)	32	WORZ	0000_0000h
1_028Ch	NAND Flash Memory Data register (NAND_MDR)	32	RO	0000_0000h
1_0300h	Nand DLL Low Config 0 Register (NAND_DLL_LOW_CFG0)	32	RW	8007_0000h
1_0304h	Nand DLL Low Config 1 Register (NAND_DLL_LOW_CFG1)	32	RW	8000_0000h
1_030Ch	NAND DLL Low Status Register (NAND_DLL_LOW_STAT)	32	RO	0000_0000h
1_0400h	NOR Event and Error Status register (NOR_EVTER_STAT)	32	W1C	0000_0000h
1_040Ch	NOR Event and Error Enable register (NOR_EVTER_EN)	32	RW	8500_0000h
1_0418h	NOR Event and Error Interrupt enable register (NOR_EVTER_INTR_EN)	32	RW	0000_0000h
1_0424h	NOR Transfer Error Attributes register 0 (NOR_ERATTR0)	32	RO	0000_0000h
1_0428h	NOR Transfer Error Attribute register 1 (NOR_ERATTR1)	32	RO	0000_0000h
1_042Ch	NOR Transfer Error Attribute register 2 (NOR_ERATTR2)	32	RO	0000_0000h
1_0440h	NOR Control register (NORCR)	32	RW	000F_0000h
1_0800h	GPCM Event and Error Status register (GPCM_EVTER_STAT)	32	W1C	0000_0000h
1_080Ch	GPCM Event and Error Enable register (GPCM_EVTER_EN)	32	RW	0540_0000h
1_0818h	GPCM Event and Error Interrupt enable register (GPCM_EVTER_INTR_EN)	32	RW	0000_0000h
1_0824h	GPCM Transfer Error Attributes register 0 (GPCM_ERATTR0)	32	RO	0000_0000h
1_0828h	GPCM Transfer Error Attributes register 1 (GPCM_ERATTR1)	32	RO	0000_0000h
1_082Ch	GPCM Transfer Error Attributes register 2 (GPCM_ERATTR2)	32	RO	0000_0000h
1_0830h	GPCM Status register (GPCM_STAT)	32	RO	0000_0000h

17.3.2 IFC Revision Control register (REV)

17.3.2.1 Offset

Register	Offset
REV	0h

17.3.2.2 Function

This register represents the revision number of the IFC.

17.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				REV_MAJ				Reserved				REV_MIN			
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.2.4 Fields

Field	Function
31-28 —	Reserved
27-24 REV_MAJ	Major Revision. It represents the major revision of IFC
23-20 —	Reserved
19-16 REV_MIN	Minor Revision. It represents the minor revision of IFC.
15-0 —	Reserved

17.3.3 Extended Chip Select Property Registers (CSPR0_EXT - CSPR7_EXT)

17.3.3.1 Offset

For a = 0 to 7:

Register	Offset
CSPRa_EXT	Ch + (a × Ch)

17.3.3.2 Function

The extended chip select property register (CSPRn_EXT) contains the extended base address, that is, the most significant bits (msb) of of the base address.

17.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								BA_EXT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.3.4 Fields

Field	Function
31-8 —	Reserved
7-0 BA_EXT	Extended Base Address: This field contains the msbs of the base address. Complete base address can be represented by concatenating CSPRn_EXT[BA_EXT] and CSPR*n*[BA] fields. Each chip selects's base register value is compared to the address on the address bus to determine if the master is accessing a memory bank controlled by the IFC. Address decoding is performed by using the address mask bit in AMASK*n*[AM] field more details of decoding is given in AMASK register description.

17.3.4 Chip-select Property register n (CSPR0 - CSPR7)

17.3.4.1 Offset

For a = 0 to 7:

Register	Offset
CSPRa	10h + (a × Ch)

17.3.4.2 Function

The chip-select property register (CSPR*n*) contains the base address and memory attributes for each bank.

NOTE

CSPR0: Only chip-select 0 is used for booting. PS, MSEL[1], and TE are obtained from the configuration word when boot_load/rcw_load occurs. V will be set for CSPR0 when boot_load/rcw_load occurs with valid port size.

17.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							PS		WP	Reserved	TE	Reserved	MSEL		V
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.4.4 Fields

Field	Function
31-16 BA	Base Address: Each base register value is compared to the address on the address bus to determine if the master is accessing a memory bank controlled by the IFC. Used with the address mask bit.
15-9 —	Reserved
8-7 PS	Port Size-Specifies the port size of this memory region. For CSPR0, PS is configured through reset configuration word as loaded during power on reset. For all other banks the value is reset to 00 (port size not defined). 00b - Reserved 01b - 8 bit 10b - 16 bit 11b - Reserved
6 WP	Write Protect: **NOTE:** 1. This bit is valid only for NAND and NOR; for GPCM this bit should not be set. **NOTE:** 2. If CS0 is used for booting from NAND or NOR, CSPR0[WP] will be set (write protected). Software must clear this bit after completing the boot operation to permit write operations on CS0 0b - Read and write accesses are allowed. 1b - Only read accesses are allowed. The memory controller does not assert chip-select on write cycles to this memory bank for NOR flash. (Refer Write protect for more details)
5 —	Reserved
4 TE	External Transceiver Enable. It specifies the value that will be driven on TE pin when a particular CSn is selected. 0b - Logic 0 will be driven on TE pin 1b - Logic 1 will be driven on TE pin
3 —	Reserved
2-1 MSEL	Machine Select 00b - NOR flash 01b - NAND flash 10b - GPCM 11b - Reserved
0 V	Valid: Indicates that the contents of CSPRn are valid. 0b - Bank is invalid 1b - Bank is valid

17.3.5 Address Mask register (AMASK0 - AMASK7)

17.3.5.1 Offset

For a = 0 to 7:

Register	Offset
AMASKa	A0h + (a × Ch)

17.3.5.2 Function

The address mask (AMASKn) registers contains the 16-bit address mask field for all four memory banks. The selected 16-bit address mask field masks corresponding CSPR*n*[BA] fields. The 16 lsbs of the 32-bit internal address do not participate in bank address matching in selecting a bank for access. Masking address bits independently allows external devices of different size address ranges to be used. Address mask bits can be set or cleared in any order in the field, allowing a resource to reside in more than one area of the address map.

The IFC only has 32 address pins at the external memory interface side. Since the system side address bus width is 40 bits, only the lower 32 bits will be given out. The upper 8 bits are used only for address decoding to identify the bank (chip select) on which access will be performed. There is no masking of the upper 8 bits of the system side address so it will be compared with CSPRn_EXT[BA_EXT] field as it is. Therefore, if the user programs the same value in the CSPRn_EXT[BA_EXT] field, the entire address range of the IFC will be 4 GB. If different values are programmed in the CSPRn_EXT[BA_EXT] field, the address range of the IFC will then be (number of chip selects) x 4 GB, that is, 32 GB.

The following logic is used for address decoding:

24-bit base address is formed as, BASE_ADDRn[23:0] = {CSPRn_EXT[BA_EXT], CSPRn[BA]}, n=0-7 (chip select)

Select CSn (chip select n) if,

{BASE_ADDRn[7:0], (BASE_ADDRn[23:8] & AMn[15:0])} ==
{SYSTEM_ADDR[39:32], (SYSTEM_ADDR[31:16] & AMn[15:0])}, where

{CSPRn_EXT[BA_EXT], (CSPRn[BA] & AMASKn[AM]) }==
{SYSTEM_ADDR[39:32], (SYSTEM_ADDR[31:16] & AMASKn[AM])}, where

- SYSTEM_ADDR is a 40-bit incoming address from the system side
- Index 39-32 represents 8 address msbs

- SYSTEM_ADDR[31:16] represent the next lower 16 address bits and the remaining 16 lsbs are not used in the bank selection
- In the logic mentioned above, the 8 msbs are not masked and are used as is for comparison.

NOTE

Chip select 0 (CS0) is used for boot purposes; if the boot source is NOR, then it must be executed in place. During booting, the IFC registers (including BA_EXT, BA, and AMASK registers) are modified. If there is race condition between the update of BA_EXT, BA, and AMASK registers, a chip select error may occur. To avoid this error and to map every transaction in CS0 during boot, the upper 8 bits of the system address and the base address, will be masked before comparison. Masking of these fields is turned off only after the first write transaction modifies the AMASK0 register.

After reset, until a first write transaction occurs to update AMASK0 register, all the transactions coming from the system side will always be mapped to CS0. After receiving the first write transaction to update AMASK0, the chip select decoding logic will work as per the above mentioned equation.

The table below shows memory bank size from 64 KB to 4 GB.

Table 17-3. Memory Bank Sizes in Relation to Address Mask

AM	Memory Bank Size
0000_0000_0000_0000	4 GBytes
1000_0000_0000_0000	2 Gbytes
1100_0000_0000_0000	1 Gbytes
1110_0000_0000_0000	512 MB
1111_0000_0000_0000	256 MB
1111_1000_0000_0000	128 MB
1111_1100_0000_0000	64 MB
1111_1110_0000_0000	32 MB
1111_1111_0000_0000	16 MB
1111_1111_1000_0000	8 MB
1111_1111_1100_0000	4 MB
1111_1111_1110_0000	2 MB
1111_1111_1111_0000	1 MB
1111_1111_1111_1000	512 KB
1111_1111_1111_1100	256 KB
1111_1111_1111_1110	128 KB
1111_1111_1111_1111	64 KB

17.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.5.4 Fields

Field	Function
31-16 AM	16-bit Address mask corresponding to memory bank
15-0 —	Reserved

17.3.6 Chip-Select Option register - GPCM (CSOR0_GPCM - CSOR7_GPCM)

17.3.6.1 Offset

For a = 0 to 7:

Register	Offset
CSORa_GPCM	130h + (a × Ch)

17.3.6.2 Function

The CSCOR*n* can work in following two modes:

- Normal GPCM
- Generic ASIC

The following figure shows the CSOR*n* fields when CSPR*n*[MSEL] selects the GPCM mode.

17.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	GPMODE	PAR	PAR_EN	Reserved	GPTO				Reserved			ABRT_RSP_EN	RGETA	WGETA	ADM_SHFT	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ADM_SHFT			Reserved	BURST_LEN			GAPERRD		Reserved		TRHZ			Reserved	BCTLD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

17.3.6.4 Fields

Field	Function
31 GPMODE	GPCM Mode of operation: 0b - Normal GPCM operation 1b - Generic ASIC mode operation
30 PAR	Parity mode. 0b - Odd Parity 1b - Even Parity
29 PAR_EN	Parity checking enable/disable: 0b - Parity checking disabled over the received data 1b - Parity checking enabled over the received data
28 —	Reserved
27-24 GPTO	GPCM Timeout Count: For normal GPCM, the timeout occur only when read data/write data transaction is external transaction acknowledgement based and IFCTA signal has not come. In Generic ASIC mode timeout occurs when RDY_L has not come for the number of IP clks cycles defined by this register field.

Table continues on the next page...

Field	Function
	0000b - 256 cycles of IFC module input clocks 0001b - 512 cycles of IFC module input clocks 0010b - 1024 cycles of IFC module input clocks 0011b - 2048 cycles of IFC module input clocks 0100b - 4096 cycles of IFC module input clocks 0101b - 8192 cycles of IFC module input clocks 0110b - 16,384 cycles of IFC module input clocks 0111b - 32,768 cycles of IFC module input clocks 1000b - 65,536 cycles of IFC module input clocks 1001b - 131,072 cycles of IFC module input clocks 1010b - 262,144 cycles of IFC module input clocks 1011b - 524,288 cycles of IFC module input clocks 1100b - 1,048,576 cycles of IFC module input clocks 1101b - 2,097,152 cycles of IFC module input clocks 1110b - 4,194,304 cycles of IFC module input clocks 1111b - 8,388,608 cycles of IFC module input clocks
23-21 —	Reserved
20 ABRT_RSP_EN	Abort Error Response Enable. An error response will be sent for read transaction if the transaction is aborted by IFCTA_B when REGTA is programmed in abort mode. This bit is valid only for normal GPCM mode. 0b - No error response will be sent. 1b - Error response will be sent for abort.
19 RGETA	GPCM external access termination mode for read access. **NOTE:** This bit is valid only for normal GPCM Mode. 0b - Abort mode. IFCTA_B signal acts as abort signal.GPCM read access is terminated internally by the controller at the expiry of TRAD counter unless aborted externally (If IFCTA_B asserts before the expiry of TRAD counter access will be terminated). Error will be reported in GPCM_EVTER_STAT[ABER] register. 1b - Acknowledgement mode. IFCTA_B acts as acknowledgement/data qualifier signal.GPCM read access is acknowledged by external pin IFCTA_B and only it can complete the read access. If it is not asserted within CSOR[GPTO] time, GPCM_EVTER_STAT[TOER] will be set.
18 WGETA	GPCM external access termination mode for write access: **NOTE:** This bit is valid only for normal GPCM Mode. 0b - Abort mode. GPCM write access is terminated if IFCTA_B asserts before expiry of TWP counter in case of single beat transaction. If IFCTA_B is asserted during burst write the current transaction is aborted and only after deassertion of IFCTA_B it will be resumed. Details are given in Normal GPCM program operation, Figure 17-49. Note that in abort mode no error is reported for write transaction. 1b - Acknowledgement mode: GPCM write access is acknowledged/qualified by external pin IFCTA_B assertion. If it is not asserted within CSOR[GPTO] time, GPCM_EVTER_STAT[TOER] will be set
17-13 ADM_SHFT	Address data multiplexing shift: Left shift the flash address by this value and assign it to address data multiplexed bus (AD[0:15]) ifc_data. By this method, the address msbs will be assigned to address data muxed bus that can be latched by asserting the AVD/ALE signal. **NOTE:** This shifting is only valid in normal GPCM mode. Others Reserved 00000b - No shift 00001b - Left shift addr[0:31] by 1 and assign it to AD[0:15] 00010b - Left shift ifc_addr by 2

Table continues on the next page...

Field	Function
	00011b - Left shift ifc_addr by 3 00100b - Left shift ifc_addr by 4 00101b - Left shift ifc_addr by 5 00110b - Left shift ifc_addr by 6 00111b - Left shift ifc_addr by 7 01000b - Left shift ifc_addr by 8 01001b - Left shift ifc_addr by 9 01010b - Left shift ifc_addr by 10 01011b - Left shift ifc_addr by 11 01100b - Left shift ifc_addr by 12 01101b - Left shift ifc_addr by 13 01110b - Left shift ifc_addr by 14 01111b - Left shift ifc_addr by 15 10000b - Left shift ifc_addr by 16 10001b - Left shift ifc_addr by 17 10010b - Left shift ifc_addr by 18 10011b - Left shift ifc_addr by 19 10100b - Left shift ifc_addr by 20
12 —	Reserved
11-9 BURST_LEN	GPCM burst length. It defines the maximum number of beats that will be sent/received in one burst cycle. This is programmed in terms of port-size transfer. For example, if the port size is 2 bytes and burst_length is 2, then the total of 8 bytes will be transferred in one burst cycle (that is, 4 beats of data transfers and each data transfer of 2 bytes). This is valid only in normal GPCM mode. 000b - Non-burst mode 001b - 2 010b - 4 011b - 8 100b - 16 101b - 32 110b - 64 111b - 128
8-7 GAPERRD	Generic ASIC parity error indication delay. Represents the delay (in terms of number of IFC_CLK cycles) between the indication of parity error for address and write data with respect to start of frame (SOF_L). 00b - 1 IFC_CLK delayed 01b - 2 IFC_CLK delayed 10b - 3 IFC_CLK delayed 11b - 4 IFC_CLK delayed
6-5 —	Reserved
4-2 TRHZ	Time for read enable high to output high impedance (Z)- Number of clocks required for memory to go in high Z after read enable deassertion. If IFC is accessing for read operation, then after last byte read wait for these many clock cycles so that there is no contention on external bus. Others Reserved 000b - Wait for 20 IFC module input clocks 001b - Wait for 40 IFC module input clocks 010b - Wait for 60 IFC module input clocks 011b - Wait for 80 IFC module input clocks 100b - Wait for 100 IFC module input clocks

Table continues on the next page...

Field	Function
1 —	Reserved
0 BCTLD	Buffer control disable. This bit signifies presence or absence of external buffer. If buffer is absent then this bit should be set to 1. 0b - BCTL is actively driven by IFC based on direction of access. 1b - BCTL is set to its default value (high) irrespective of direction of access.

17.3.7 Chip-Select Option register - NAND Flash Mode (CSOR0_NAND - CSOR7_NAND)

17.3.7.1 Offset

For a = 0 to 7:

Register	Offset
CSORa_NAND	130h + (a × Ch)

17.3.7.2 Function

The following figure shows the CSOR*n* fields when CSPR*n*[MSEL] selects the NAND flash mode.

NOTE

- CSOR0 is the only chip-select used for booting.
- ECC_DEC_EN, ECC_MODE[0:1], and PGS are obtained from the configuration word when boot_load/rcw_load occurs.

17.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ECC_ENC_EN	Reserved	ECC_MODE		Reserved	ECC_DEC_EN	Reserved	RAL		Reserved	PGS			Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPRZ			Reserved		PB			Reserved	NAND_MODE		TRHZ			Reserved	BCTLD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

17.3.7.4 Fields

Field	Function
31 ECC_ENC_EN	ECC Encoder Enable/Disable bit: **NOTE:** The encoder should be enabled only when performing full page operations. In case of partial page operations, the encoder should be disabled. 0b - ECC encoding disabled 1b - ECC Encoding enabled
30 —	Reserved
29-28 ECC_MODE	ECC Mode of operation 00b - 4 bit correction per 512 byte data sector 01b - 8 bit correction per 512 byte data sector 10b - 24 bit correction per 1 KB sector 11b - 40 bit correction per 1 KB sector
27 —	Reserved
26 ECC_DEC_EN	ECC Decoding Enable/Disable bit **NOTE:** The decoder should be enabled only when performing full page operations. In case of partial page operations, the decoder should be disabled 0b - ECC decoding disabled 1b - ECC decoding enabled
25 —	Reserved
24-23	Row Address Length: Number of address bytes issued during page address operation

Table continues on the next page...

Field	Function
RAL	**NOTE:** Bytes for column address are determined by page size 00b - 1 byte 01b - 2 bytes 10b - 3 bytes 11b - 4 bytes
22 —	Reserved
21-19 PGS	Page Size Others Reserved 000b - 512 Bytes 001b - 2 KB 010b - 4 KB 011b - 8 KB 100b - 16 KB
18-16 —	Reserved
15-13 SPRZ	Spare size **NOTE:** Depending on the ECC mode and page size, a fixed value of the spare region is selected during boot. For more information, see Booting methods. Others Reserved 000b - 16 Bytes 001b - 64 Bytes 010b - 128 Bytes 011b - 210 Bytes 100b - 218 Bytes 101b - 224 Bytes 110b - Spare size information will be used from corresponding CSORn_EXT register.
12-11 —	Reserved
10-8 PB	Pages per Block Others Reserved 000b - 32 Pages 001b - 64 Pages 010b - 128 Pages 011b - 256 Pages 100b - 512 Pages
7 —	Reserved
6-5 NAND_MODE	NAND mode of operation Others Reserved 00b - Asynchronous mode 01b - NVDDR 10b - Reserved 11b - Reserved
4-2 TRHZ	Time for read enable high to output high impedance (Z). Number of clocks required for memory to go in high-Z after read enable deassertion.

Table continues on the next page...

Field	Function
	This field is used during last read data access. If the IFC is accessing the NAND flash for a read operation, then after the last byte is read the NAND FSM must wait for TRHZ clock cycles so that there is no contention on the external buffer. Other settings are Reserved. 000b - Wait for 20 IP Clocks 001b - Wait for 40 IP Clocks 010b - Wait for 60 IP Clocks 011b - Wait for 80 IP Clocks 100b - Wait for 100 IP Clocks
1 —	Reserved
0 BCTLD	Buffer control disable. This bit signifies presence or absence of external buffer. If buffer is absent then this bit should be set to 1. 0b - BCTL is actively driven by IFC based on direction of access. 1b - BCTL is set to its default value (high) irrespective of direction of access.

17.3.8 Chip-Select Option register - NOR Flash Mode (CSOR0_NOR - CSOR7_NOR)

17.3.8.1 Offset

For a = 0 to 7:

Register	Offset
CSORa_NOR	130h + (a × Ch)

17.3.8.2 Function

The following figure shows the CSOR*n* fields when CSPR*n*[MSEL] selects the NOR flash mode.

NOTE

- CSOR0 is the only chip-select used for booting.
- ADM_SHFT is obtained from the configuration word when boot_load/rcw_load occurs.

17.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R / W	ADM_SHFT_MODE	Reserved		PGRD_EN	Reserved			AVD_TGL_PGM_EN	SYNC_RD_TYPE	Reserved					ADM_SHFT	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	ADM_SHFT			Reserved	WORD_LINE_LEN			SYNC_RD_EN	Reserved	NOR_MODE		TRHZ			Reserved	BCTLD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

17.3.8.4 Fields

Field	Function
31 ADM_SHFT_MODE	Address shift mode 0b - Address msbs will be assigned to AD bus and ADDR bus carries the lsb 1b - AD bus will carry lsbs and ADDR bus carries the msb
30-29 —	Reserved
28 PGRD_EN	Page read enable from NOR device **NOTE:** Software should issue transaction based on device page size and it is valid only for parallel NOR devices and should not be set for ADM NOR devices. 0b - A multi-beat read transaction received from system bus will be split into per-beat accesses (based on NOR port size). 1b - A multi-beat read transaction received from system bus will be performed as a single-page read operation (burst type) on NOR flash.
27-25 —	Reserved
24 AVD_TGL_PGM_EN	AVD toggle enable during burst program 0b - Assert AVD only for the first address phase (performed on the flash interface) of a burst write operation received from the system interface 1b - Assert AVD during every subsequent address phase (performed on the flash interface) including the first phase of a burst write operation received from the system interface

Table continues on the next page...

Field	Function
23 SYNC_RD_TYPE	Synchronous NOR b read enable.The field is applicable for both unmuxed (parallel) and muxed (ADM) NOR devices (if they have burst read capability). If this bit is set then PGRD_EN field is ignored as it is applicable only to Async page read capable unmuxed NOR devices. Also note that NOR device program operation is always asynchronous. In this mode IFC will ignore data whenever RB_B is low during data phase. Details are given in section NOR flash control machine. 0b - Sync burst read disable 1b - Sync burst read enable
22-18 —	Reserved
17-13 ADM_SHFT	Address data multiplexing shift. It controls the way internal 32 bit address is placed on the external bus, during NOR/GPCM address phase. Address shifting will be done by 1 bit. Patterns not shown are reserved. During reset, in case the RCW source is from NOR then, if cfg_rcw_src[2:1]=00 (00 22b addressability) then ADM_SHFT = 6 if cfg_rcw_src[2:1]=01 (00 25b addressability) then ADM_SHFT = 9 if cfg_rcw_src[2:1]=10 (00 28b addressability) then ADM_SHFT = 12 00000b - No shift 00001b - shift ifc_addr by 1 00010b - shift ifc_addr by 2 00011b - shift ifc_addr by 3 00100b - shift ifc_addr by 4 00101b - shift ifc_addr by 5 00110b - shift ifc_addr by 6 00111b - shift ifc_addr by 7 01000b - shift ifc_addr by 8 01001b - shift ifc_addr by 9 01010b - shift ifc_addr by 10 01011b - shift ifc_addr by 11 01100b - shift ifc_addr by 12 01101b - shift ifc_addr by 13 01110b - shift ifc_addr by 14 01111b - shift ifc_addr by 15 10000b - shift ifc_addr by 16 10001b - shift ifc_addr by 17 10010b - shift ifc_addr by 18 10011b - shift ifc_addr by 19 10100b - shift ifc_addr by 20
12 —	Reserved
11-9 WORD_LINE_LEN	Synchronous NOR word line length. Its defines the word line length of synchronous NOR during burst read (for unmuxed as well as muxed devices). It defines maximum no. of beats that will be received in one burst cycle. This is programmed in terms of port-size transfer. For example if port size is 2 bytes and burst_length is 2 then total 8 bytes will be transferred in one burst cycle i.e. 4 beats (words) of data transfers and each data transfer of 2 bytes. This field is applicable only for synchronous NOR burst read (CSOR[SYNC_NOR_ON] =1) mode 000b - Non-burst mode 001b - 2 010b - 4

Table continues on the next page...

Field	Function
	011b - 8 100b - 16 101b - 32 110b - 64 111b - 128
8 SYNC_RD_EN	Synchronous NOR burst read enable. This field is applicable for both unmuxed (parallel) as well as muxed (ADM) NOR devices (if they have burst read capability). If this bit is set then PGRD_EN field is ignored as it is applicable only to Async page read capable unmuxed NOR devices. Also note that NOR device program operation is always asynchronous. In this mode IFC will ignore data whenever RB_B is low during data phase. Details are given in section NOR flash control machine. 0b - Sync burst read disable 1b - Sync burst read enable
7 —	Reserved
6-5 NOR_MODE	Type of the NOR device. Others Reserved 00b - These NOR devices have dedicated address and data pins. In this mode external latch is used to latch address msbs (with ALE assertion) with ADM_SHFT_MODE=0. If NOR device is synchronous burst read capable then user need to set SYNC_RD_EN bit along with proper value programmed in WORD_LINE_LEN. In this case IFC will also assert AVD_B apart from ALE. 01b - These NOR devices contains multiplexed address and data pins which can be used to carry address as well as data. In this mode external latch is not required and ADM_SHFT_MODE needs to be programmed to 1. If ADM NOR device supports synchronous burst read then used need to set SYNC_RD_EN bit along with proper value programmed in WORD_LINE_LEN. In this case IFC will assert AVD_B to latch address values carried by AD bus during address phase.
4-2 TRHZ	Time for read enable high to output high impedance (Z). Number of clocks required for memory to go in high Z after read enable deassertion. If IFC is accessing NOR flash for read operation, then after last byte read NOR FSM must wait for these many clock cycles so that there is no contention on external buffer. Others Reserved 000b - Wait for 20 IP Clocks 001b - Wait for 40 IP Clocks 010b - Wait for 60 IP Clocks 011b - Wait for 80 IP Clocks 100b - Wait for 100 IP Clocks
1 —	Reserved
0 BCTLD	Buffer control disable. This bit signifies presence or absence of external buffer. If buffer is absent then this bit should be set to 1. 0b - BCTL is actively driven by IFC based on direction of access. 1b - BCTL is set to its default value (high) irrespective of direction of access.

17.3.9 Extended Chip-Select Option register - NAND Flash Mode (CSOR0_EXT - CSOR7_EXT)

17.3.9.1 Offset

For a = 0 to 7:

Register	Offset
CSORa_EXT	134h + (a × Ch)

17.3.9.2 Function

The figure below shows the CSOR*n*_EXT fields when CSPR*n*[MSEL] selects the NAND flash mode.

NOTE

CSOR*n*_EXT register is reserved if CSPR*n*[MSEL] is in NOR/GPCM mode.

17.3.9.3 Diagram

17.3.9.4 Fields

Field	Function
31-21 —	Reserved
20 AUTO_TIM_PARAMS_SEL	Used to select the automatically calculated timing parameters instead of the user programmed timing parameters. **NOTE:** Automatic parameter calculation option is not applicable for Async Mode. User has to compute and program FTIM registers for Async Mode. 0b - User programmed timing parameters (FTIM0-FTIM3) selected 1b - Automatically calculated timing parameters selected
19 —	Reserved
18-16 MODE_FREQ	Indicates the timing mode with which the source synchronous device is programmed. This field is qualified with NAND_MODE (CSOR[25:26]) and defined as follows: NAND_MODE = 01 (NVDDR mode) 000b - 20 MHz 001b - 33 MHz 010b - 50 MHz 011b - 66 MHz 100b - 83 MHz 101b - 100 MHz 110b - Reserved 111b - Reserved
15-11 —	Reserved
10-0 SPARE_BYTES_CSn	No. of bytes in spare region. This filed is applicable only when corresponding CSOR*n*[SPRZ] is 3'h110. 00000000000b - 0 Spare region bytes 00000000001b - 1 Spare region byte. 11111111110b - 2046 spare region bytes

17.3.10 Flash Timing register 0 for CSn - Normal GPCM Mode (FTIM0_CS0_GPCM - FTIM0_CS7_GPCM)

17.3.10.1 Offset

For a = 0 to 7:

Register	Offset
FTIM0_CSa_GPCM	1C0h + (a × 30h)

17.3.10.2 Function

The following figure shows the FTIM0_CS*n* fields when CSPR*n*[MSEL] selects the GPCM mode.

17.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TACSE				Reserved						TEADC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TEAHC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.10.4 Fields

Field	Function
31-28 TACSE	Address phase (After address hold cycle) end to chip enable assertion time 0000b - Reserved 0001b - 1 IFC module input clock 0010b - 2 IFC module input clocks 1111b - 15 IFC module input clocks
27-22 —	Reserved
21-16 TEADC	External latch address delay cycles. External address latch enable (ALE) assertion time (pulse width of external latch enable signal) 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
15-6 —	Reserved
5-0 TEAHC	External latch address hold cycles: Address hold cycle relative to external latch enable signal 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks

17.3.11 Flash Timing register 0 for Chip Select n - NAND flash asyncNVDDR2and toggleNVDDR mode (FTIM0_CS0_NAND - FTIM0_CS7_NAND)

17.3.11.1 Offset

For a = 0 to 7:

Register	Offset
FTIM0_CSa_NAND	1C0h + (a × 30h)

17.3.11.2 Function

The flash timing registers define the flash interface timings. These register fields are defined differently depending on the machine type (NOR/NAND/GPCM) selected for that bank by CSPR*n*[MSEL] field. Timing registers are provided separately for each chip-select. Timing parameter are in terms of IFC module input clock cycles. More details about the register field is given in section Programming model for flash interface timing .

NOTE

- Timing values of FTIM0, FTIM1, FTIM2, and FTIM3 registers for chip-select 0 in NAND/NOR flash mode will be loaded at boot_load/rcw_load by the parameters passed.
- If CSOR_EXTn[AUTO_TIM_PARAMS_SEL] is set, then the values returned on reading the FTIM0, FTIM1, FTIM2, FTIM3 registers will be the ones calcuated automatically by IFC (based on CSOR_EXTn[MODE_FREQ] and the Clock divsion ratio) and not the ones programmed in the FTIM registers. This is valid only when CSORn[NAND_MODE] = 2b01, 2b10, 2b11.
- The TRP value loaded during RCW/boot load is taken as (TRAD +2) input clocks.

- For normal GPCM mode (write transaction) all the three timing parameters (that is, TEAHC, TACSE, and TCS) should not be programmed zero together.
- For normal GPCM mode (read transaction) all the three timing parameters (that is, TEAHC, TACSE, and TACO) should not be programmed zero together.

During NAND and NOR boot, default values of of FTIM*n*CS0 must be as shown in below table.

Table 17-4. Default values of FTIM*n*CS0 during boot

FTIMnCS0	Value during NAND RCW and boot	Value during NOR RCW	Value during NOR boot
FTIM0_CS0	181C_0C10	2002_0202	4005_0405
FTIM1_CS0	5454_141E	1B00_0E0E	5000_2828 7200_3412
FTIM2_CS0	0600_8028	0410_501C	4005_0405
FTIM3_CS0	2C00_0000	0000_0000	0000_0000

The following register shows the FTIM0_CSn fields when CSPRn[MSEL] selects the NAND Flash Mode and CSORn[NAND_MODE] selects NAND Flash NV-DDR Mode.

17.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		TCS						Reserved		TCAD					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.11.4 Fields

Field	Function
31-30	Reserved

Table continues on the next page...

Field	Function
—	
29-24 TCS	Chip Enable (CE) setup time. 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
23-22 —	Reserved
21-16 TCAD	Command, Address, Data delay (Command-command, command-address, address-address, address-command, command/address- start of data) 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
15-0 —	Reserved

17.3.12 Flash Timing register 0 for Chip Select n - NAND flash Asynchronous Mode (FTIM0_CS0_NAND_ASYNC_MODE - FTIM0_CS7_NAND_ASYNC_MODE)

17.3.12.1 Offset

For a = 0 to 7:

Register	Offset
FTIM0_CSa_NAND_ASYNC_MODE	1C0h + (a × 30h)

17.3.12.2 Function

The following register shows the FTIM0_CSn fields when CSPRn[MSEL] selects the NAND Flash Mode and CSORn[NAND_MODE] selects NAND Flash Asynchronous Mode.

17.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	TCCST						Reserved	TWP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TWCHT						Reserved		TWH					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.12.4 Fields

Field	Function
31 —	Reserved
30-25 TCCST	This parameter is defined in the following two ways: Command latch enable (CLE) assertion time with respect to CS assertion time. CLE/ALE assertion time after previous CLE/ALE phase (post TWH time). 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
24 —	Reserved
23-16 TWP	Write enable (WE) pulse width 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
15-14 —	Reserved
13-8 TWCHT	WE to command hold time 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
7-6	Reserved

Table continues on the next page...

Field	Function
—	
5-0 TWH	WE high hold time 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks

17.3.13 Flash Timing register 0 for CSn - NOR Flash Mode (FTIM0_CS0_NOR - FTIM0_CS7_NOR)

17.3.13.1 Offset

For a = 0 to 7:

Register	Offset
FTIM0_CSa_NOR	1C0h + (a × 30h)

17.3.13.2 Function

The following figure shows the FTIM0_CS*n* fields when CSPR*n*[MSEL] selects the NOR flash mode.

17.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TACSE				Reserved						TEADC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TAVDS						Reserved		TEAHC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.13.4 Fields

Field	Function
31-28 TACSE	Address phase (after external latch enable deassertion) end to chip enable assertion time (Applicable only for Async Read CSORn[SYNC_RD_EN]=0) 0000b - Reserved 0001b - 1 IFC module input clock 0010b - 2 IFC module input clocks 1111b - 15 IFC module input clocks
27-22 —	Reserved
21-16 TEADC	External latch address delay cycles. Pulse width of external latch enable signal ALE (applicable for both sync and async read of unmuxed NOR) AVD assertion time in async read (CSORn[SYNC_RD_EN]=0) for muxed (ADM) NOR device (CSORn[NOR_MODE] =1) 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
15-14 —	Reserved
13-8 TAVDS	Delay between CS assertion to AVD/ALE assertion 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
7-6 —	Reserved
5-0 TEAHC	Latch address hold cycles. Address hold relative to deassertion of external latch enable signal ALE (applicable for both sync and async read of unmuxed NOR). Address hold relative to AVD_B deassertion (applicable for async read (CSORn[SYNC_RD_EN]=0) for muxed (ADM) NOR device (CSORn[NOR_MODE] =1) 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks

17.3.14 Flash Timing register 1 for CSn - Normal GPCM Mode (FTIM1_CS0_GPCM - FTIM1_CS7_GPCM)

17.3.14.1 Offset

For a = 0 to 7:

Register	Offset
FTIM1_CSa_GPCM	1C4h + (a × 30h)

17.3.14.2 Function

The following figure shows the FTIM1_CS*n* fields when CSPR*n*[MSEL] and CSOP*n*[GPMODE] selects the normal GPCM mode.

17.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TACO								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TRAD						Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.14.4 Fields

Field	Function
31-24 TACO	CS assertion to output enable (OE) assertion setup time 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clock
23-14 —	Reserved
13-8 TRAD	GPCM read access delay: It represents the output enable assertion time. See Normal GPCM read operation .

Table continues on the next page...

Field	Function
	000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
7-0 —	Reserved

17.3.15 Flash Timing register 1 for Chip-Select n - NAND Flash NVDDR Mode (FTIM1_CS0_NAND - FTIM1_CS7_NAND)

17.3.15.1 Offset

For a = 0 to 7:

Register	Offset
FTIM1_CSa_NAND	1C4h + (a × 30h)

17.3.15.2 Function

The following figure shows the FTIM1_CS*n* fields when CSPR*n*[MSEL] selects the NAND flash mode.

17.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TADLE								TWB							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TR R						TWRCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.15.4 Fields

Field	Function
31-24 TADLE	Effective address to data loading time 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
23-16 TWB	Clock Rising Edge to SR[6] (R/B) low
15-14 —	Reserved
13-8 TRR	Ready busy high to read enable (RE) low time 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
7-0 TWRCK	W/R low to data output cycle

17.3.16 Flash Timing register 1 for Chip Select n - Asynchronous Mode (FTIM1_CS0_NAND_ASYNC_MODE - FTIM1_CS7_NAND_ASYNC_MODE)

17.3.16.1 Offset

For a = 0 to 7:

Register	Offset
FTIM1_CSa_NAND_ASYNC_MODE	1C4h + (a × 30h)

17.3.16.2 Function

The following figure shows the FTIM1_CS*n* fields when CSPR*n*[MSEL] selects the NAND flash asynchronus mode.

17.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TADLE								TWBE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TR R						TRP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.16.4 Fields

Field	Function
31-24 TADLE	Effective address to data loading time 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
23-16 TWBE	WE high (after TWH time) to ready busy (RB_B) low time 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11101111b - 239 IFC module input clocks 11110000-11111111b - Reserved
15-14 —	Reserved
13-8 TRR	Ready busy high to read enable (RE) low time 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
7-0 TRP	RE pulse width 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks

17.3.17 Flash Timing register 1 for CSn - NOR Flash Mode (FTIM1_CS0_NOR - FTIM1_CS7_NOR)

17.3.17.1 Offset

For a = 0 to 7:

Register	Offset
FTIM1_CSa_NOR	1C4h + (a × 30h)

17.3.17.2 Function

The following figure shows the FTIM1_CS*n* fields when CSPR*n*[MSEL] selects the NOR flash mode.

17.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TACO								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TRAD_NOR								TSEQRAD_NOR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.17.4 Fields

Field	Function
31-24 TACO	Unmuxed (Parallel) NOR (CSORn[NOR_MODE=0): • Async Read (CSORn[SYNC_RD_EN=0): In this mode this field represents CS assertion to Output Enable (OE) assertion setup time • Sync Read (CSORn[SYNC_RD_EN=1): In this mode this field represents address hold to Output Enable (OE) assertion setup time Muxed (ADM) NOR (CSORn[NOR_MODE] = 1):

Table continues on the next page...

Field	Function
	• Async Read (CSORn[SYNC_RD_EN=0): In this mode this field is not applicable • Sync Read (CSORn[SYNC_RD_EN=1): In this mode this field represents address hold to Output Enable (OE) assertion setup time 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
23-16 —	Reserved
15-8 TRAD_NOR	NOR flash read access delay: It represents the read enable to data access time plus total round trip board delay from the external NOR flash memory during read operation. Its value can be calculated as *Read data access time considering data gets sampled in the mid of data eye + 2* Board Delay + 2* . 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
7-0 TSEQRAD_NOR	NOR flash sequential read access delay. It represents the address to data access time plus total round trip delay from the external NOR flash memory during sequential read operation. Its value can be calculated as *[NOR Page Address Delay (max) for sequential read + 2*Board Delay + Device Setup Time]*. 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks

17.3.18 Flash Timing register 2 for CSn - Normal GPCM Mode (FTIM2_CS0_GPCM - FTIM2_CS7_GPCM)

17.3.18.1 Offset

For a = 0 to 7:

Register	Offset
FTIM2_CSa_GPCM	1C8h + (a × 30h)

17.3.18.2 Function

The following figure shows the FTIM2_CS*n* fields when CSPR*n*[MSEL] selects the normal GPCM mode.

17.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				TCS				Reserved		TCH				Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TWP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.18.4 Fields

Field	Function
31-28 —	Reserved
27-24 TCS	Chip-select assertion to WE assertion setup time 0000b - Reserved 0001b - 1 IFC module input clock 0010b - 2 IFC module input clocks 1111b - 15 IFC module input clocks
23-22 —	Reserved
21-18 TCH	Chip-select hold time with respect to WE deassertion 0000b - Reserved 0001b - 1 IFC module input clock 0010b - 2 IFC module input clocks 1111b - 15 IFC module input clocks
17-8 —	Reserved
7-0 TWP	Write enable pulse width 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clock

17.3.19 Flash Timing register 2 for Chip Select n - NAND Flash NVDDR Mode (FTIM2_CS0_NAND - FTIM2_CS7_NAND)

17.3.19.1 Offset

For a = 0 to 7:

Register	Offset
FTIM2_CSa_NAND	1C8h + (a × 30h)

17.3.19.2 Function

The following figure shows the FTIM2_CS*n* fields when CSPR*n*[MSEL] selects the NAND flash mode.

17.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCKWR								TWHR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TRHW								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.19.4 Fields

Field	Function
31-24 TCKWR	Data output end to W/R high.
23-16 TWHR	Command, Address or Data input cycle to Data output clycle.
15-8	Data Output cycle to Command, Address or Data Input cycle

Table continues on the next page...

Field	Function
TRHW	
7-0 —	Reserved

17.3.20 Flash Timing register 2 for Chip Select n - NAND Flash Asynchronous Mode (FTIM2_CS0_NAND_ASYNC_MODE - FTIM2_CS7_NAND_ASYNC_MODE)

17.3.20.1 Offset

For a = 0 to 7:

Register	Offset
FTIM2_CSa_NAND_ASYNC_MODE	1C8h + (a × 30h)

17.3.20.2 Function

The following figure shows the FTIM2_CS*n* fields when CSPR*n*[MSEL] selects the NAND flash asynchronous mode.

17.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TRAD								Reserved				TREH
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TREH					Reserved			TWHRE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.20.4 Fields

Field	Function
31-29 —	Reserved
28-21 TRAD	Flash read access delay: It represents the data sampling time of read data. It should be programmed such that sampling is done at the center of the received data eye. For calculation of data eye width and appropriate data sampling time refer to NAND asynchronous mode calculating read data window width . 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
20-17 —	Reserved
16-11 TREH	RE_B High Time 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
10-8 —	Reserved
7-0 TWHRE	WE_B High to RE_B Low Effective 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks

17.3.21 Flash Timing register 2 for CSn - NOR Flash Mode (FTIM2_CS0_NOR - FTIM2_CS7_NOR)

17.3.21.1 Offset

For a = 0 to 7:

Register	Offset
FTIM2_CSa_NOR	1C8h + (a × 30h)

17.3.21.2 Function

The following figure shows the FTIM2_CS*n* fields when CSPR*n*[MSEL] selects the NOR flash mode.

17.3.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				TCS				Reserved		TCH				Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWPH						Reserved		TWP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.21.4 Fields

Field	Function
31-28 —	Reserved
27-24 TCS	Chip-select assertion to WE assertion setup time 0000b - Reserved 0001b - 1 IFC module input clock 0010b - 2 IFC module input clocks 1111b - 15 IFC module input clocks
23-22 —	Reserved
21-18 TCH	Chip-select hold time with respect to WE deassertion 0000b - Reserved 0001b - 1 IFC module input clock 0010b - 2 IFC module input clocks 1111b - 15 IFC module input clocks
17-16 —	Reserved
15-10	Write enable pulse high time

Table continues on the next page...

Field	Function
TWPH	000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
9-8 —	Reserved
7-0 TWP	Write enable pulse width 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks

17.3.22 Flash Timing register 3 for CSn - Normal GPCM Mode (FTIM3_CS0_GPCM - FTIM3_CS7_GPCM)

17.3.22.1 Offset

For a = 0 to 7:

Register	Offset
FTIM3_CSa_GPCM	1CCh + (a × 30h)

17.3.22.2 Function

The following figure shows the FTIM3_CS*n* fields when CSPR*n*[MSEL] selects the GPCM flash mode.

17.3.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TAAD						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.22.4 Fields

Field	Function
31-26 TAAD	GPCM address access delay. This timing parameter is required for read cycles when GPCM is working in burst mode, that is, burst length is programmed to a non-zero value. The first data beat of a burst is sampled after TRAD time of OE assertion; subsequent beat data are sampled after TAAD time of the previous sample pulse. The same timing is used for sending the new address of the beat. The second address is sent after TRAD time of OE assertion and subsequent address beats are sent after TAAD time of the last beat. This is programmed in terms of IFC module input clock but should always be a mupltiple of IFC_CLK. This is valid only for normal GPCM mode. 000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clocks
25-0 —	Reserved

17.3.23 Flash Timing register 3 for Chip Select n - NAND Flash Mode (FTIM3_CS0_NAND - FTIM3_CS7_NAND)

17.3.23.1 Offset

For a = 0 to 7:

Register	Offset
FTIM3_CSa_NAND	1CCh + (a × 30h)

17.3.23.2 Function

The following figure shows the FTIM3_CS*n* fields when CSPR*n*[MSEL] selects the NAND flash NVDDRmode.

17.3.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TWW								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.23.4 Fields

Field	Function
31-24 TWW	Write Protect WP_B transition time. It represents the idle cycles inserted by the IFC controller whenever corresponding WP_B pin toggles. Refer Write protect for more details. 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
23-0 —	Reserved

17.3.24 Flash Timing register 3 for CSn - NOR Flash Mode (FTIM3_CS_NOR)

17.3.24.1 Offset

Register	Offset
FTIM3_CS_NOR	1CCh

17.3.24.2 Function

The following figure shows the FTIM3_CS*n* fields when CSPR*n*[MSEL] selects the NOR flash mode. All fields in this register are applicable only for sync read mode (CSORn[SYNC_RD_EN=1).

NOTE

Offset for IFC_FTIM3_CSn_NOR are 0x0_01CC, 0x0_01FC, 0x0_022C, 0x0_025C, 0x0_028C, 0x0_02BC, 0x0_02EC, and 0x0_031C

17.3.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		TCHVL						Reserved		TAVDP					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		TVHAX						Reserved		TIACC					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.24.4 Fields

Field	Function
31-30 —	Reserved
29-24	Rising edge of IFC_CLK to CS assertion time.

Table continues on the next page...

Field	Function
TCHVL	000000b - Reserved 000001b - 1 IFC module input clock 000010b - 2 IFC module input clocks 111111b - 63 IFC module input clock
23-22 —	Reserved
21-16 TAVDP	AVD_B assertion time (pulse width of AVD_B)
15-14 —	Reserved
13-8 TVHAX	Address hold time from AVD_B deassertion
7-6 —	Reserved
5-0 TIACC	Initial data access latency : It represents the time from rising edge of IFC_CLK where address gets sampled (AVD is low) to rising edge of IFC_CLK when first data word is valid. This field is represented in terms of number of IFC_CLK cycles.

17.3.25 Flash Timing register 3 for Chip Select n - NAND Flash Asynchronous Mode (IFC_FTIM03_CS_NAND_ASYNC_MODE - IFC_FTIM73_CS_NAND_ASYNC_MODE)

17.3.25.1 Offset

For a = 0 to 7:

Register	Offset
IFC_FTIMa3_CS_NAND_ASYNC_MODE	1CCh + (a × 30h)

17.3.25.2 Function

The following figure shows the FTIM3_CS*n* fields when CSPR*n*[MSEL] selects the NAND flash asynchronous mode.

17.3.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TWW								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.25.4 Fields

Field	Function
31-24 TWW	Write Protect WP_B transition time. It represents the idle cycles inserted by the IFC controller whenever corresponding WP_B pin toggles. Refer Write protect for more details. 00000000b - Reserved 00000001b - 1 IFC module input clock 00000010b - 2 IFC module input clocks 11111111b - 255 IFC module input clocks
23-0 —	Reserved

17.3.26 Ready Busy Status for each Chip Select (RB_STAT)

17.3.26.1 Offset

Register	Offset
RB_STAT	400h

17.3.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RB0	RB1	RB2	RB3	RB4	RB5	RB6	RB7	Reserved							
W																
Reset	u	u	u	u	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.26.3 Fields

Field	Function
31 RB0	Ready Busy input from CS0. This register field represents the status of RB_B signal of CS0 sampled by IFC at every clock. 0b - Device connected at CS0 is driving RB_B signal as Busy 1b - Device connected at CS0 is driving RB_B signal as Ready.
30 RB1	Ready Busy input from CS1 This register field represents the status of RB_B signal of CS1 sampled by IFC at every clock. 0b - Device connected at CS1 is driving RB_B signal as Busy 1b - Device connected at CS1 is driving RB_B signal as Ready.
29 RB2	Ready Busy input from CS2 This register field represents the status of RB_B signal of CS2 sampled by IFC at every clock. 0b - Device connected at CS2 is driving RB_B signal as Busy 1b - Device connected at CS2 is driving RB_B signal as Ready.
28 RB3	Ready Busy input from CS3 This register field represents the status of RB_B signal of CS3 sampled by IFC at every clock. 0b - Device connected at CS3 is driving RB_B signal as Busy 1b - Device connected at CS3 is driving RB_B signal as Ready.
27 RB4	Ready Busy input from CS4 This register field represents the status of RB_B signal of CS4 sampled by IFC at every clock. 0b - At least one of the device connected at CS4-CS7 is busy 1b - All devices connected to CS4-CS7 are ready
26 RB5	Ready Busy input from CS5 This register field represents the status of RB_B signal of CS5 sampled by IFC at every clock. 0b - At least one of the device connected at CS4-CS7 is busy 1b - All devices connected to CS4-CS7 are ready
25 RB6	Ready Busy input from CS6 This register field represents the status of RB_B signal of CS6 sampled by IFC at every clock.

Table continues on the next page...

Field	Function
	0b - At least one of the device connected at CS4-CS7 is busy 1b - All devices connected to CS4-CS7 are ready
24 RB7	Ready Busy input from CS7 This register field represents the status of RB_B signal of CS7 sampled by IFC at every clock. 0b - At least one of the device connected at CS4-CS7 is busy 1b - All devices connected to CS4-CS7 are ready
23-0 —	Reserved

17.3.27 Ready Busy Map register (RB_MAP)

17.3.27.1 Offset

Register	Offset
RB_MAP	404h

17.3.27.2 Function

This register represents the mapping of Ready/Busy_B pins to each of the chip select. Single Ready/Busy_B pin can be mapped to multiple chip selects.

17.3.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	CS0_RB			Reserved	CS1_RB			Reserved	CS2_RB			Reserved	CS3_RB		
Reset	u	u	u	u	0	0	0	1	0	0	1	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	CS4_RB			Reserved	CS5_RB			Reserved	CS6_RB			Reserved	CS7_RB		
Reset	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1

17.3.27.4 Fields

Field	Function
31 —	Reserved
30-28 CS0_RB	Ready/Busy_B mapping to chip-select0. This field defines which of the Ready/Busy_B pin is associated to chip-select0. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7
27 —	Reserved
26-24 CS1_RB	Ready/Busy_B mapping to chip-select1. This field defines which of the Ready/Busy_B pin is associated to chip-select1. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7
23 —	Reserved
22-20 CS2_RB	Ready/Busy_B mapping to chip-select2. This field defines which of the Ready/Busy_B pin is associated to chip-select2. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7
19 —	Reserved
18-16 CS3_RB	Ready/Busy_B mapping to chip-select3. This field defines which of the Ready/Busy_B pin is associated to chip-select3. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6

Table continues on the next page...

Field	Function
	111b - Ready/Busy-7
15 —	Reserved
14-12 CS4_RB	Ready/Busy_B mapping to chip-select4. This field defines which of the Ready/Busy_B pin is associated to chip-select4. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7
11 —	Reserved
10-8 CS5_RB	Ready/Busy_B mapping to chip-select5. This field defines which of the Ready/Busy_B pin is associated to chip-select5. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7
7 —	Reserved
6-4 CS6_RB	Ready/Busy_B mapping to chip-select6. This field defines which of the Ready/Busy_B pin is associated to chip-select6. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7
3 —	Reserved
2-0 CS7_RB	Ready/Busy_B mapping to chip-select 7. This field defines which of the Ready/Busy_B pin is associated to chip-select 7. 000b - Ready/Busy-0 001b - Ready/Busy-1 010b - Ready/Busy-2 011b - Ready/Busy-3 100b - Ready/Busy-4 101b - Ready/Busy-5 110b - Ready/Busy-6 111b - Ready/Busy-7

17.3.28 Write Protect Map register (WP_MAP)

17.3.28.1 Offset

Register	Offset
WP_MAP	408h

17.3.28.2 Function

This register represents the write protect register fields.

17.3.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	CS0_WP			Reserved	CS1_WP			Reserved	CS2_WP			Reserved	CS3_WP		
W																
Reset	u	u	u	u	0	0	0	1	0	0	1	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	CS4_WP			Reserved	CS5_WP			Reserved	CS6_WP			Reserved	CS7_WP		
W																
Reset	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1

17.3.28.4 Fields

Field	Function
31 —	Reserved
30-28 CS0_WP	Write Protect_B mapping to chip-select0. This field defines which of the WP_B pin is associated to chip-select0. 000b - WP-0 001b - WP-1

Table continues on the next page...

Field	Function
	010b - WP-2 011b - WP-3 100b - WP-4 101b - WP-5 110b - WP-6 111b - WP-7
27 —	Reserved
26-24 CS1_WP	WP_B mapping to chip-select1. This field defines which of the WP_B pin is associated to chip-select1. 000b - WP-0 001b - WP-1 010b - WP-2 011b - WP-3 100b - WP-4 101b - WP-5 110b - WP-6 111b - WP-7
23 —	Reserved
22-20 CS2_WP	WP_B mapping to chip-select2. This field defines which of the WP_B pin is associated to chip-select2. 000b - WP-0 001b - WP-1 010b - WP-2 011b - WP-3 100b - WP-4 101b - WP-5 110b - WP-6 111b - WP-7
19 —	Reserved
18-16 CS3_WP	WP_B mapping to chip-select3. This field defines which of the WP_B pin is associated to chip-select3. 000b - WP-0 001b - WP-1 010b - WP-2 011b - WP-3 100b - WP-4 101b - WP-5 110b - WP-6 111b - WP-7
15 —	Reserved
14-12 CS4_WP	WP_B mapping to chip-select4. This field defines which of the WP_B pin is associated to chip-select4.
11 —	Reserved
10-8 CS5_WP	WP_B mapping to chip-select5. This field defines which of the WP_B pin is associated to chip-select5.
7 —	Reserved

Table continues on the next page...

Field	Function
6-4 CS6_WP	WP_B mapping to chip-select6. This field defines which of the WP_B pin is associated to chip-select6. 000b - WP-0 001b - WP-1 010b - WP-2 011b - WP-3 100b - WP-4 101b - WP-5 110b - WP-6 111b - WP-7
3 —	Reserved
2-0 CS7_WP	Write Protect_B mapping to chip-select7. This field defines which of the WP_B pin is associated to chip-select7. 000b - WP-0 001b - WP-1 010b - WP-2 011b - WP-3 100b - WP-4 101b - WP-5 110b - WP-6 111b - WP-7

17.3.29 General Control register (GCR)

17.3.29.1 Offset

Register	Offset
GCR	40Ch

17.3.29.2 Function

The reset value of this field is passed through parameter.

17.3.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SOFT_RST_ALL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TBCTL_TRN_TIME					Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.29.4 Fields

Field	Function
31 SOFT_RST_ALL	Software Reset All: It is used to reset the whole IFC hardware (NAND, NOR, and GPCM). With this reset only the event/error status registers of IFC will be cleared. Programming registers will retain their value. The software reset mechanism is implemented such that IFC will send the pending transactions on system bus with good response. In case of pending read transaction, garbage data will be supplied back. Once the whole IFC is in IDLE state, the software can clear this bit. IFC hardware stops software from clearing this bit till all the processing has been done and IFC hardware is in idle state. If this bit is set, hardware will clear it once reset operation is completed. Software should poll this bit to get cleared before initiating any new transaction. 0b - No Software reset 1b - Assert Software reset
30-16 —	Reserved
15-11 TBCTL_TRN_TIME	It represents the turnaround time of external buffer in terms of number of IFC module input clock cycles. BCTL should be kept high for these many clock cycles before issuing a write transaction after a read on flash interface. See Data buffer control (BCTL) for more details.
10-0 —	Reserved

17.3.30 Common Event and Error Status register (CM_EVTER_STAT)

17.3.30.1 Offset

Register	Offset
CM_EVTER_STAT	418h

17.3.30.2 Function

Common event and error status register (CM_EVTER_STAT) indicates the cause of an error or event that cannot be classified in NAND, NOR, and GPCM.

17.3.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSER	Reserved														
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.30.4 Fields

Field	Function
31 CSER	Chip-Select Error 0b - No chip-select error 1b - A transaction was sent to IFC which is not mapped to any memory bank
30-0 —	Reserved

17.3.31 Common Event and Error Enable register (CM_EVTER_EN)

17.3.31.1 Offset

Register	Offset
CM_EVTER_EN	424h

17.3.31.2 Function

This register is used to enable/disable the logging of event and error indication in the CM_EVTER_STAT register.

17.3.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CSEREN	Reserved														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.31.4 Fields

Field	Function
31 CSEREN	Chip-Select Error Checking Enable 0b - Chip-Select Error Checking Disabled 1b - Chip-Select Error Checking Enabled
30-0 —	Reserved

17.3.32 Common Event and Error Interrupt Enable register (CM_EVTER_INTR_EN)

17.3.32.1 Offset

Register	Offset
CM_EVTER_INTR_EN	430h

17.3.32.2 Function

Common event and error interrupt enable register (CM_EVTER_INTR_EN) is used to enable/disable the generation of interrupt signals corresponding to common error and event reporting. Software must clear pending events and errors in CM_EVTER_STAT register before enabling the interrupts.

17.3.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CSERIREN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.32.4 Fields

Field	Function
31 CSERIREN	Chip-Select Error Interrupt Enable 0b - Chip-Select Error Interrupt Disabled

Table continues on the next page...

Field	Function
	1b - Chip-Select Error Interrupt Enabled
30-0 —	Reserved

17.3.33 Common Transfer Error Attributes register 0 (CM_ERATTR0)

17.3.33.1 Offset

Register	Offset
CM_ERATTR0	43Ch

17.3.33.2 Function

These registers store the attribute corresponding to the first transaction on which common error occured.

Common transfer error attribute register 0 (CM_ERATTR0) is used to register the transaction attributes corresponding to the first common error occured.

17.3.33.3 Diagram

17.3.33.4 Fields

Field	Function
31 ERTYP	Transaction type of the error 0b - Write Transaction 1b - Read Transaction
30-28 —	Reserved
27-20 ERAID	ID of the error transaction
19-16 —	Reserved
15-8 ERSRCID	Source ID of the error transaction, always zero.
7-0 —	Reserved

17.3.34 Common Transfer Error Attributes register 1 (CM_ERATTR1)

17.3.34.1 Offset

Register	Offset
CM_ERATTR1	440h

17.3.34.2 Function

Common transfer error attribute register1 (CM_ERATTR1) is used to register the transaction address corresponding to the first error.

17.3.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.34.4 Fields

Field	Function
31-0 ERADDR	32 bits of transaction address corresponding to error transaction

17.3.35 Clock Control register (CCR)

17.3.35.1 Offset

Register	Offset
CCR	44Ch

17.3.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CLKDIV				Reserved				CLK_DLY			
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INV_CLK_EN	Reserved														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.35.3 Fields

Field	Function
31-28 —	Reserved
27-24 CLKDIV	Clock division ratio: This field is used to generate the IFC clock from the IFC module input clock. The IFC clock is sent out after inverting it according to the INV_CLK_EN setting. 0000b - Reserved 0001b - Divide by 2 0010b - 0011b - Divide by 4 0100b - Reserved 0101b - 0110b - Reserved 0111b - Divide by 8 1000b - Reserved 1001b - Reserved 1010b - Reserved 1011b - 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b -
23-20 —	Reserved
19-16 CLK_DLY	IFC Clock Delay This field specifies the number of IFC module input clocks by which external clock is delayed. 0000b - No delay 0001b - 1 IFC module input clocks delay introduced 0010b - 2 IFC module input clocks delay introduced 1111b - 15 IFC module input clocks delay introduced

Table continues on the next page...

Field	Function
15 INV_CLK_EN	IFC Clock Inversion This field specifies whether IFC clock is inverted before sending out. 0b - IFC clock is not inverted 1b - IFC clock is inverted
14-0 —	Reserved

17.3.36 Clock Status register (CSR)

17.3.36.1 Offset

Register	Offset
CSR	450h

17.3.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CLK_STAT	Reserved														
W																
Reset	u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.36.3 Fields

Field	Function
31 CLK_STAT	Clock Status. This bit is cleared by hardware in case any of the field of CCR is changed. It is again set when external clock is stable. New GPCM operation can be initiated only after clock is stable for new CLK_DIV ratio.

Table continues on the next page...

Field	Function
	0b - Clock is unstable 1b - Clock is stable
30-0 —	Reserved

17.3.37 DDR Clock Control register Low (DDR_CCR_LOW)

17.3.37.1 Offset

Register	Offset
DDR_CCR_LOW	454h

17.3.37.2 Function

This register is used for programming the clock divider when NV-DDR interface is at frequencies up to 100 MHz.

17.3.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		GLOBAL_DDR_CLK_EN	PGM_DDR_CLK_STOP	Reserved	DDR_LOW_CLKDIV				Reserved		TDQSS	Reserved			
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.37.4 Fields

Field	Function
31-30 —	Reserved
29 GLOBAL_DDR_CLK_EN	Switches off the IFC DDR clock. Used when device is not programmed as Source Synchronous. 0b - Clock disabled. 1b - Clock enabled.
28 PGM_DDR_CLK_STOP	DDR Clock Stop in case of Program Operation (except SET FEATURES).This field is valid only when CSOR[NAND_MODE] = 2'b01 0b - Clock Enabled 1b - Clock Stopped
27 —	Reserved
26-23 DDR_LOW_CLKDIV	Clock division ratio: This field is used to generate the divided clock from ip_clk. 0000- Divide by 2 0001- Divide by 4 0010- Divide by 6 0011- Divide by 8 0100- Divide by 10 0101- Divide by 12 0110- Divide by 14 0111- Divide by 16 1000- Divide by 18 1001- Divide by 20 1010- Divide by 22 1011- Divide by 24 1100- Divide by 26 1101- Divide by 28 1110- Divide by 30 1111- Divide by 32 Note- After reset, divided by 4 clock is sent out
22-21 —	Reserved
20 TDQSS	Data input to first DQS transition delay. This field is valid only when CSOR[NAND_MODE] = 2'b01 0b - 0.75 IFC_DDR_CLOCK period 1b - 1.25 IFC_DDR_CLOCK period
19-0 —	Reserved

17.3.38 NAND Configuration register (NCFGR)

17.3.38.1 Offset

Register	Offset
NCFGR	1_0000h

17.3.38.2 Function

A separate 1 KB memory-mapped region is assigned to NAND-specific registers.

17.3.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R / W	BOOT	Reserved	SRAM_INIT_EN	Reserved	SINGLE_DATA_MODE	Reserved			ADDR_MODE		Reserved					NUM_LOOP
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	NUM_LOOP				Reserved				NUM_WAIT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.38.4 Fields

Field	Function
31 BOOT	NAND flash auto-boot load mode: During system boot from NAND flash, this bit remains set to alter the use of the FCM buffer RAM. Software should clear BOOT once FCM is to be restored for normal operation. Setting BOOT without auto-boot in progress only alters the mapping of the buffer RAM.

Table continues on the next page...

Field	Function
	0b - NAND FCM is operating in normal functional mode, with a 32-KB FCM buffer RAM. In this case normal SRAM buffer mapping applies. 1b - Flash is accessed in boot mode. SRAM buffer mapping gets changed and whole 16 KB NAND flash main area in the buffer looks as contiguous linear addressable memory region. This bit is set by hardware when por_cfg_rcw_load or por_cfg_boot_load occurs with NAND as bootable device.
30 —	Reserved
29 SRAM_INIT_EN	SRAM Initialization Enable This bit is provided to trigger the auto initialization of complete SRAM with FF data. Software must set this bit before initiating first program operation on NAND post reset. IFC clears this bit after finishing this operation. After setting this bit to 1, software must poll this bit. Reading back value 0 confirms that IFC hardware has filled the SRAM and cleared this bit. 0b - No auto SRAM Initialization. 1b - Triggers the IFC sram initialization logic which fills all sram buffer locations with hFF data
28 —	Reserved
27 SINGLE_DATA_MODE	This bit is valid only when CSORn[NAND_MODE] is configured as NV-DDR Mode. In case of source synchronous operation the NAND flash repeats a byte of data on both the rising and falling edge of the DQS for certain commands, namely GET FEATURES, READ ID. For the SET FEATURES command, IFC has to repeat each byte of data for the rising and falling edge of DQS. When this bit is set: For read operations (GET FEATURES, READ ID), IFC will ignore the repeated bytes of data sent by the NAND Flash on the falling edge of the DQS for read operations. For write operations (SET FEATURES), IFC will replicate each byte of data to be written on both rising and falling edge of DQS. 0b - Single data mode disabled 1b - Single data mode enabled
26-24 —	Reserved
23-22 ADDR_MODE	Addressing Mode: Applicable for the current active CS. Applicable only with opcodes CA0 and RA0. Others Reserved 00b - ROW0/COL0, ROW0+1/COL0, ROW0+2/COL0 and so on, in each iteration defined by NUM_LOOP field (Incremental by Page Size) 01b - ROW0/COL0 addresses in first iteration, ROW1/COL1 address in second iteration, ROW2/COL2 address in third iteration, ROW3/COL3 address in fourth iteration
21-17 —	Reserved
16-12 NUM_LOOP	Number of loop iterations of FIR sequences for multi page operations 5h00 - 1 time execution of FIR sequence 5h01 - 2 time execution of FIR sequence ..... 5h1E - 31 time execution of FIR sequence 5h1F- 32 time execution of FIR sequence

Table continues on the next page...

Field	Function
	Note1: Maximum permissible NUM_LOOP value depending on page size: 512 byte page: 5h1F 2KB page : 5h7 4KB page: 5h3 8KB page: 5h1 16KB page : 5h0 Note 2: In any case for ADDR_MODE = 01, NUM_LOOP should not exceed 5h3
11-8 —	Reserved
7-0 NUM_WAIT	Number of wait cycles It represents the count value for which FCM will wait when used with opcode NWAIT in FIR. The value specified in this register represents the number of IFC module input clock cycles. Minimum value to be programmed is 2.

17.3.39 NAND Flash Command register 0 (NAND_FCR0)

17.3.39.1 Offset

Register	Offset
NAND_FCR0	1_0014h

17.3.39.2 Function

The NAND flash command registers hold up to 8 NAND flash EEPROM command bytes that may be referenced by opcodes in NAND_FIR during FCM operation. The values of the commands should follow the manufacturer's datasheet for the relevant NAND flash.

17.3.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMD0								CMD1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD2								CMD3							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.39.4 Fields

Field	Function
31-24 CMD0	General purpose FCM flash command byte 0. Opcodes in FIR that issue command index 0 write CMD0 to the NAND flash command/data bus.
23-16 CMD1	General purpose FCM flash command byte 1. Opcodes in FIR that issue command index 1 write CMD1 to the NAND flash command/data bus.
15-8 CMD2	General purpose FCM flash command byte 2. Opcodes in FIR that issue command index 2 write CMD2 to the NAND flash command/data bus.
7-0 CMD3	General purpose FCM flash command byte 3. Opcodes in FIR that issue command index 3 write CMD3 to the NAND flash command/data bus

17.3.40 NAND Flash Command register 1 (NAND_FCR1)

17.3.40.1 Offset

Register	Offset
NAND_FCR1	1_0018h

17.3.40.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMD4								CMD5							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD6								CMD7							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.40.3 Fields

Field	Function
31-24 CMD4	General purpose FCM flash command byte 4. Opcodes in FIR that issue command index 4 write CMD4 to the NAND flash command/data bus.
23-16 CMD5	General purpose FCM flash command byte 5. Opcodes in FIR that issue command index 5write CMD5 to the NAND flash command/data bus.
15-8 CMD6	General purpose FCM flash command byte 6. Opcodes in FIR that issue command index 6 write CMD6 to the NAND flash command/data bus.
7-0 CMD7	General purpose FCM flash command byte 7. Opcodes in FIR that issue command index 7 write CMD7 to the NAND flash command/data bus

17.3.41 Flash Row Address register n (ROW0 - ROW3)

17.3.41.1 Offset

Register	Offset
ROW0	1_003Ch
ROW1	1_004Ch
ROW2	1_005Ch
ROW3	1_006Ch

17.3.41.2 Function

ROW*n* registers are used for addressing the NAND flash memory. These registers are used as per the field NCFGR1[ADDR_MODE]. ROW*n* register holds the row address to be issued on NAND flash interface using address phase. CSOR*n*[RAL] field determines the number of bits to be issued to the flash from ROW register during row address phase.

17.3.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.41.4 Fields

Field	Function
31-0 RA	Row Address This register holds the row address to be issued on flash during row address phase

17.3.42 Flash COL Address register n (COL0 - COL3)

17.3.42.1 Offset

Register	Offset
COL0	1_0044h
COL1	1_0054h
COL2	1_0064h
COL3	1_0074h

17.3.42.2 Function

COL*n* registers are used for addressing the NAND FLASH memory. These registers are used as per the field NCFGR1[ADDR_MODE]. The following holds the column address to be issued to flash during address phase. CSOR*n*[PGS] field determines the number of bits to be issued to flash from COL register during column address phase.

For small-page size, that is, 512 Bytes, the four lsbs of ROW*n* register is used to select the IFC buffer for data transfer to/from IFC buffer to/from flash. It indexes the page in NAND flash EEPROM at the current block, and locates the corresponding transfer buffer in the FCM buffer RAM.

The four lsbs of RA index, 1 of the 16, 1-KB buffers in the FCM buffer RAM as follows:

- 0000-The page is transferred to/from FCM buffer 0, address offsets 0x0000-0x03FF
- 0001-The page is transferred to/from FCM buffer 1, address offsets 0x0400-0x07FF
- 0010-The page is transferred to/from FCM buffer 2, address offsets 0x0800-0x0BFF
- 0011-The page is transferred to/from FCM buffer 3, address offsets 0x0C00-0x0FFF
- 0100-The page is transferred to/from FCM buffer 4, address offsets 0x1000-0x13FF
- 0101-The page is transferred to/from FCM buffer 5, address offsets 0x1400-0x17FF
- 0110-The page is transferred to/from FCM buffer 6, address offsets 0x1800-0x1BFF
- 0111-The page is transferred to/from FCM buffer 7, address offsets 0x1C00-0x1FFF
- 1000-The page is transferred to/from FCM buffer 8, address offsets 0x2000-0x23FF
- 1001-The page is transferred to/from FCM buffer 9, address offsets 0x2400-0x27FF
- 1010-The page is transferred to/from FCM buffer 10, address offsets 0x2800-0x2BFF
- 1011-The page is transferred to/from FCM buffer 11, address offsets 0x2C00-0x2FFF
- 1100-The page is transferred to/from FCM buffer 12, address offsets 0x3000-0x33FF
- 1101-The page is transferred to/from FCM buffer 13, address offsets 0x3400-0x37FF
- 1110-The page is transferred to/from FCM buffer 14, address offsets 0x3800-0x3BFF
- 1111-The page is transferred to/from FCM buffer 15, address offsets 0x3C00-0x3FFF

17.3.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MS	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			CA												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.42.4 Fields

Field	Function
31 MS	Main/spare region locator.: **NOTE:** In the case that NAND_BC[BC] = 0, MS is treated as 0. 0b - Data is transferred to/from the main region of the SRAM buffer; that is, the first 512 bytes of the buffer are used. 1b - Data is transferred to/from the spare region of the SRAM buffer; that is, the second 512 bytes of the buffer are used, but only an initial 16 bytes of spare region are defined.
30-13 —	Reserved
12-0 CA	Column Address This register holds the column address to be issued on flash during column address phase. CA indexes the first byte to transfer to/from the main or spare region of the NAND flash EEPROM and corresponding transfer buffer. In the case that NAND_BC[BC] = 0, CA is treated as 0. For MS = 0, CA can range 0x0000-0x1FF; for MS = 1, CA can range 0x000-0x00F. For a 16-bit port size, the least significant bit of the Column Address is assumed to be zero; hence, the Column Address remains a byte index at all times.

17.3.43 Flash COL Address register for 2 KB Large-Page Device (COL0_2KB - COL3_2KB)

17.3.43.1 Offset

Register	Offset
COL0_2KB	1_0044h
COL1_2KB	1_0054h
COL2_2KB	1_0064h
COL3_2KB	1_0074h

17.3.43.2 Function

For large-page size of 2 KB, two lsbs of ROW*n* register are used to select the IFC buffer for data transfer to/from IFC buffer to/from flash. It indexes the page in NAND flash EEPROM at the current block, and locates the corresponding transfer buffer in the FCM buffer RAM.

The two lsbs of RA index one of the four 4-KB buffers in the FCM buffer RAM as follows:

- 00-The page is transferred to/from FCM buffer 0, address offsets 0x0000-0x0FFF
- 01-The page is transferred to/from FCM buffer 1, address offsets 0x1000-0x1FFF
- 10-The page is transferred to/from FCM buffer 2, address offsets 0x2000-0x2FFF
- 11-The page is transferred to/from FCM buffer 3, address offsets 0x3000-0x3FFF

17.3.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MS	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			CA												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.43.4 Fields

Field	Function
31 MS	Main/spare region locator. In the case that NAND_BC[BC] = 0, MS is treated as 0. 0b - Data is transferred to/from the main region of the SRAM buffer; that is, the first 2048 bytes of the buffer are used. 1b - Data is transferred to/from the spare region of the SRAM buffer; that is, the second 2048 bytes of the buffer are used, but only an initial 64 bytes of spare region are defined.
30-13 —	Reserved
12-0 CA	Column Address This register holds the column address to be issued on flash during column address phase. CA indexes the first byte to transfer to/from the main or spare region of the NAND flash EEPROM and corresponding transfer buffer. In the case that NAND_BC[BC] = 0, CA is treated as 0. For MS = 0, CA can range 0x000-0x7FF; for MS = 1, CA can range 0x000-0x03F. For a 16-bit port size, the least significant bit of the Column Address is assumed to be zero; hence, the Column Address remains a byte index at all times.

17.3.44 Flash COL Address register for 4 KB Large-Page Device (COL0_4KB - COL3_4KB)

17.3.44.1 Offset

Register	Offset
COL0_4KB	1_0044h
COL1_4KB	1_0054h
COL2_4KB	1_0064h
COL3_4KB	1_0074h

17.3.44.2 Function

For large-page size of 4 KB, lsb of ROW*n* register is used to select the IFC buffer for data transfer to/from IFC buffer to/from flash. It indexes the page in NAND flash EEPROM at the current block, and locates the corresponding transfer buffer in the FCM buffer RAM.

The lsb of RA indexes one of the two 8 KB buffers in the FCM buffer RAM as follows:

- 0-The page is transferred to/from FCM buffer 0, address offsets 0x0000-0x1FFF
- 1-The page is transferred to/from FCM buffer 1, address offsets 0x2000-0x3FFF

17.3.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MS	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			CA												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.44.4 Fields

Field	Function
31 MS	Main/spare region locator. In the case that NAND_BC[BC] = 0, MS is treated as 0. 0b - Data is transferred to/from the main region of the SRAM buffer; that is, the first 4096 bytes of the buffer are used. 1b - Data is transferred to/from the spare region of the SRAM buffer; that is, the second 4096 bytes of the buffer are used, but only an initial 64 bytes of spare region are defined.
30-13 —	Reserved
12-0 CA	Column Address This register holds the column address to be issued on flash during column address phase. CA indexes the first byte to transfer to/from the main or spare region of the NAND flash EEPROM and corresponding transfer buffer. In the case that NAND_BC[BC] = 0, COL is treated as 0. For MS = 0, CA can range 0x000-0xFFF; for MS = 1, CA can range 0x000- CSOR*n*[SPRZ]. For a 16-bit port size, the least significant bit of the Column Address is assumed to be zero; hence, the Column Address remains a byte index at all times.

17.3.45 Flash Byte Count register for NAND Flash (NAND_BC)

17.3.45.1 Offset

Register	Offset
NAND_BC	1_0108h

17.3.45.2 Function

This register defines the NAND flash data block transfer size.

17.3.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		BC													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.45.4 Fields

Field	Function
31-14 —	Reserved
13-0 BC	Byte count determines how many bytes are transferred by the flash controller during data read (RBCD) or data write (WBCD) opcodes. In case of partial page operations, the value programmed in this field must be aligned to the port size of the device. Thus, if the bank port size is 16 bits, the lsb bit of BC (that is, bit 31) is ignored. The first byte accessed in the NAND flash is located by the COLn register, and successive bytes are transferred until BC bytes have been counted If BC = 0, an entire flash page and its spare region will be transferred by FCM, in which case COL*n*[MS] and COL*n*[CA] are treated as zero regardless of their values. **NOTE:** BC = 0 is the only setting that permits flash controller to generate and check ECC.

17.3.46 NAND Flash Instruction register 0 (NAND_FIR0)

17.3.46.1 Offset

Register	Offset
NAND_FIR0	1_0110h

17.3.46.2 Function

The following flash instruction registers hold the instructions. When any operation on a selected bank is triggered and the FIR is programmed by the user, flash controller executes 6-bit opcode at a time until all the opcodes have been fetched from all the FIR registers. There are three such registers provided for the application. A total of 15 instructions can be programmed for FLASH operations.

NAND flash instruction register0 holds first five opcodes.

Table 17-5. Instruction Opcodes

Opcode encoding (6 bits)	Opcode name	Description
0x00	NOOP	No-operation and end of operation sequence
0x01	CA0	Issue current column address (CA) as set in COL0
0x02	CA1	Issue current column address as set in COL1
0x03	CA2	Issue current column address as set in COL2
0x04	CA3	Issue current column address as set in COL3
0x05	RA0	Issue current row address as set in ROW0
0x06	RA1	Issue current row address as set in ROW1
0x07	RA2	Issue current row address as set in ROW2
0x08	RA3	Issue current row address as set in ROW3
0x09	CMD0	Issue command from NAND_FCR0[CMD0]
0x0A	CMD1	Issue command from NAND_FCR0[CMD1]
0x0B	CMD2	Issue command from NAND_FCR0[CMD2]
0x0C	CMD3	Issue command from NAND_FCR0[CMD3]
0x0D	CMD4	Issue command from NAND_FCR1[CMD4]
0x0E	CMD5	Issue command from NAND_FCR1[CMD5]
0x0F	CMD6	Issue command from NAND_FCR1[CMD6]
0x10	CMD7	Issue command from NAND_FCR1[CMD7]

Table continues on the next page...

Table 17-5. Instruction Opcodes (continued)

Opcode encoding (6 bits)	Opcode name	Description
0x11	CW0	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR0[CMD0]
0x12	CW1	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR0[CMD1]
0x13	CW2	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR0[CMD2]
0x14	CW3	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR0[CMD3]
0x15	CW4	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR1[CMD4]
0x16	CW5	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR1[CMD5]
0x17	CW6	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR1[CMD6]
0x18	CW7	Wait for TWBE time, poll R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then issue command from NAND_FCR1[CMD7]
0x19	WBCD	Write BC bytes of data from current FCM buffer to flash device
0x1A	RBCD	Wait for R/B_B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then Read BC bytes of data from flash device into current FCM RAM buffer
0x1B	BTRD	Same as RBCD except in this case deassertion of read enable will happen once the read data has been sampled (same as boot read)
0x1C	RDSTAT	Wait for TWHR (for NVDDR)/TWHRE (Async) time, then Read one byte/two bytes (8b/16b port) of data from flash device into RS0 and RS1 field of FSR.
0x1D	NWAIT	Wait for NCFGR[NUM_WAIT] Clock cycles
0x1E	WFR	Wait for TWBE time, poll Ready
0x1F	SBRD	Wait for R/B to return high or time-out (depends upon IFC_NANDCR[FTOCNT]), then Read 8/16 bits of data from NAND flash memory into the NAND_MDR register. COLn register will determine the location from where the data will be fetched in a given page. Deassertion of read enable will happen once the read data has been sampled. With this opcode NAND_BC register field value will be ignored. Although the value is ignored, it is mandatory that the NAND_BC register hold a non-zero value else the Column Address would be treated as 0 (refer Flash COL Address register n (COL0 - COL3))
0x20	UA	Issue current row address as set in ROW3 The SRAM Buffer used for data transfers is always Buffer 0

Table continues on the next page...

Table 17-5. Instruction Opcodes (continued)

Opcode encoding (6 bits)	Opcode name	Description
0x21	RB	Wait for TWHR (for NVDDR)/TWHRE (Async) time, then Read BC bytes of data from flash device into current FCM RAM buffer. Deassertion of read enable will happen once the read data has been sampled
Others	-	Reserved

17.3.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OP0						OP1						OP2			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OP2		OP3						OP4						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.46.4 Fields

Field	Function
31-26 OP0	Opcode0
25-20 OP1	Opcode1
19-14 OP2	Opcode2
13-8 OP3	Opcode3
7-2 OP4	Opcode4
1-0 —	Reserved

17.3.47 NAND Flash Instruction register 1 (NAND_FIR1)

17.3.47.1 Offset

Register	Offset
NAND_FIR1	1_0114h

17.3.47.2 Function

NAND flash instruction register1 holds other five opcodes.

17.3.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OP5						OP6						OP7			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OP7		OP8						OP9						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.47.4 Fields

Field	Function
31-26 OP5	Opcode5
25-20 OP6	Opcode6
19-14 OP7	Opcode7

Table continues on the next page...

Field	Function
13-8 OP8	Opcode8
7-2 OP9	Opcode9
1-0 —	Reserved

17.3.48 NAND Flash Instruction register 2 (NAND_FIR2)

17.3.48.1 Offset

Register	Offset
NAND_FIR2	1_0118h

17.3.48.2 Function

NAND flash instruction register2 holds last five opcodes.

17.3.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OP10						OP11						OP12			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OP12		OP13						OP14						Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.48.4 Fields

Field	Function
31-26 OP10	Opcode10
25-20 OP11	Opcode11
19-14 OP12	Opcode12
13-8 OP13	Opcode13
7-2 OP14	Opcode14
1-0 —	Reserved

17.3.49 NAND Chip-Select register (NAND_CSEL)

17.3.49.1 Offset

Register	Offset
NAND_CSEL	1_015Ch

17.3.49.2 Function

This register tells the NAND FCM about the selected chip-select on which a program or read operation has to be performed. As NAND flash is accessed through an SRAM buffer (memory-mapped), software tells the NAND FCM which chip-select is desired using this register.

17.3.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSEL			Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.49.4 Fields

Field	Function
31-29 —	Reserved
28-26 CSEL	Chip-Select for NAND flash operation 000b - Chip-select0 001b - Chip-select1 010b - Chip-select2 011b - Chip-select3 111b - Chip select7
25-0 —	Reserved

17.3.50 NAND Operation Sequence Start (NANDSEQ_STRT)

17.3.50.1 Offset

Register	Offset
NANDSEQ_STRT	1_0164h

17.3.50.2 Function

This register is used to trigger the operation on NAND flash.

The following figure shows the NANDSEQ_STRT register. Only one operation can be triggered at a time. Software has to trigger the operation and the chip will clear this after completing the operation.

17.3.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R / W	NAND_FIR_STRT	Reserved							AUTO_ERS	Reserved		AUTO_PGM	Reserved		AUTO_CPB	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	Reserved	AUTO_RD	Reserved		AUTO_STAT_RD	Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.50.4 Fields

Field	Function
31 NAND_FIR_STRT	NAND flash operation start Writing 1 to this register bit triggers normal operation sequence programmed in NAND FIR registers on NAND flash.
30-24 —	Reserved
23 AUTO_ERS	Automatic erase Writing 1 to this register bit triggers automatic erase operation **NOTE:** The FIR sequence executed is {CW0, RA0, CMD1, and NOOP}
22-21	Reserved

Table continues on the next page...

Field	Function
—	
20 AUTO_PGM	Automatic program Writing 1 to this register bit triggers automatic program operation **NOTE:** The FIR sequence executed is {CW0, CA0, RA0, WBCD, CMD1, and NOOP}. This sequence will not support small page program, hence user cannot use auto program for small page device.
19-18 —	Reserved
17 AUTO_CPB	Automatic copyback Writing 1 to this register bit triggers automatic copy back operation **NOTE:** The FIR sequence executed is {CW0, CA0, RA0, CMD1, CW2, CA1, RA1, CMD3, CW4, RDSTAT, and NOOP}
16-15 —	Reserved
14 AUTO_RD	Automatic read operation Writing 1 to this register bit triggers automatic read operation **NOTE:** The FIR sequence executed is: For small-page device: {CW0, CA0, RA0, RBCD, NOOP} For large-page device: {CW0, CA0, RA0, CMD1, RBCD, NOOP}
13-12 —	Reserved
11 AUTO_STAT_RD	Automatic status read Writing 1 to this register bit triggers automatic read status **NOTE:** The FIR sequence executed is {CW0, RDSTAT, NOOP}
10-0 —	Reserved

17.3.51 NAND Event and Error Status register (NAND_EVTER_STAT)

17.3.51.1 Offset

Register	Offset
NAND_EVTER_STAT	1_016Ch

17.3.51.2 Function

NAND event and error status register (NAND_EVTER_STAT) indicates the cause of an error or event corresponding to NAND flash.

The following figure shows the register fields. It is write-1-to-clear register.

17.3.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OPC	Reserved			FTOE R	WPE R	ECCE R	Reserved								
W	W1C				W1C	W1C	W1C									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RCW_DN	BOOT_DN	Reserved		BBI_SRCH_SEL	Reserved										
W	W1C	W1C			W1C											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.51.4 Fields

Field	Function
31 OPC	NAND flash operation complete event. OPC indicates that all instructions in FIR has been executed. **NOTE:** 1. The software should not poll this bit during auto boot. Instead RCW_DN/ BOOT_DN should be used. However, software must clear this bit post auto boot. **NOTE:** 2. This bit is set even if a flash timeout, ECC or write protect error occurs. For example, if a flash timeout error is detected for a read operation intiated on the NAND flash memory device, IFC will dump BC bytes worth of garbage data from the NAND flash interface into the internal SRAM and set this bit. Similarly, in case of NAND write protect error, IFC asserts chip select and sends the program data to NAND device, it is device which ignores the data as write protect signal is asserted. User must wait for operation completion (poll for OPC bit getting set) even in case of timeout before initiating next operation. As the transaction goes into NAND, OPC will be set at the end of FIR execution. OPC bit indicates that all the instructions in the FIRs have been executed. Also, it implies that OPC should be polled first and then FSR for completion. 0b - NAND flash operation is not complete

Table continues on the next page...

Field	Function
	1b - NAND flash operation completed
30-28 —	Reserved
27 FTOER	Flash timeout error 0b - No flash interface timeout 1b - Flash interface timeout occured
26 WPER	Write protect error **NOTE:** Error is asserted when the write operation starts executing on the flash interface. 0b - No write protect error 1b - A write is attempted to the memory bank which is write protected
25 ECCER	ECC error **NOTE:** With the clearing of this error status bit, the ECCSTAT0/1/2/3 register contents will also be cleared. 0b - No uncorrectable ECC Error occured on page read operation 1b - Uncorrectable ECC error occured in page read operation
24-16 —	Reserved
15 RCW_DN	This bit is set when one flash page has been read from the flash device and written into the SRAM buffer during RCW load. It is write-1-to-clear bit and can be cleared by software after RCW loading. This event does not have any corresponding event enable and interrupt enable bit.
14 BOOT_DN	This event is set when 8 KB flash data has been read from the flash device and written into the SRAM buffer during BOOT load. It is write-1-to-clear bit and can be cleared by software after BOOT loading. This events will not have any corresponding event enable and interrupt enable bit.
13-12 —	Reserved
11 BBI_SRCH_SEL	Bad Block Indicator search select: This status register bit represents the location of bad block indication in each block of NAND flash device connected at chip-select0. This field will be updated with input port value por_cfg_bbi_srch_sel when por_cfg_boot_load /por_cfg_rcw_load pulse occurs. 0b - Read bad block indicator (BBI) corresponding to page0 and page1 of each block of the NAND flash device connected at chipselect 0, to identify the first good block during auto-boot 1b - Read bad block indicator (BBI) corresponding to page0 and the last page of each block of the NAND flash device connected at chipselect 0, to identify the first good block during auto-boot
10-0 —	Reserved

17.3.52 NAND Page Read Completion Event Status register (PGRDCMPL_EVT_STAT)

17.3.52.1 Offset

Register	Offset
PGRDCMPL_EVT_STAT	1_0174h

17.3.52.2 Function

NAND flash page read completion event register indicates the status of the pages read from NAND flash and stored in SRAM buffer. 16 bits of this register represent 16 sectors each of 512 bytes (used in ECC decoding). There can be 1, 4, 8, and 16 sectors in each page of size 512 bytes, 2 KB, 4 KB . Event interrupt will be generated if one complete page has been written in SRAM buffer (when ecc_dec_en = 0) or when one complete page has been fixed in the SRAM after decoding (ecc_dec_en = 1). Software can read the corresponding page whose data has been completely written in SRAM (ECC fixed) and clear the corresponding bits in this register.

24/40-bit ECC is computed on 1 KB sector and applicable for 4 KB page sizes. However the definition of this register remains same except the fact that 512 is the byte size and not the sector size.

NOTE

The register gets cleared when software clears the OPC bit in NAND_EVTER_STAT register. Even if per page event interrupt is used for reading the data from SRAM, NAND_EVTER_STAT[OPC] is the true indication of NAND operation completion.

The following figure shows the register fields. It is write-1-to-clear register.

17.3.52.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SEC_DONE															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.52.4 Fields

Field	Function
31-16 SEC_DONE	For small page: Each of the 16 bits of this field represent one small page read data completion status Bit 0 set-Page 0 done, that is, flash read data is completely written and fixed in SRAM buffer Bit 1 set-Page 1 done Bit 2 set-Page 2 done Bit 3 set-Page 3 done ... Bit15 Set-Page 15 done For 2KB Page: SRAM buffer can accommodate 4 page (2KB) each containing four 512 byte sectors SEC_DONE[0:3] = 4'b1111: Page 0 done SEC_DONE[4:7] = 4'b1111: Page 1 done SEC_DONE[8:11] = 4'b1111: Page 2 done SEC_DONE[12:15] = 4'b1111: Page 3 done For 4KB Page: SRAM buffer can accommodate 2 page (4KB) each containing eight 512 byte sectors SEC_DONE[0:7] = 8'b1111_1111: Page 0 done SEC_DONE[8:15] = 8'b1111_1111: Page 1 done For 8KB Page: Only one page can be buffered in SRAM. SEC_DONE[0:15] = 16'b 1111_1111_1111_1111: Single page of 8KB done. **NOTE:** Event interrupt will be generated when one complete page read is done. Software has to read the corresponding page from SRAM buffer and clear the bits from this register. For example if page size is 2 KB and page 0 is being written in SRAM buffer, then SEC_DONE[0:15] will be 16'HF000, now software has to write the data 16'HF000 in this register to clear it.
15-0 —	Reserved

17.3.53 NAND Event and Error Enable register (NAND_EVTER_EN)

17.3.53.1 Offset

Register	Offset
NAND_EVTER_EN	1_0180h

17.3.53.2 Function

Event and Error Enable Register (NAND_EVTER_EN) is used to enable/disable the logging of event and error indication in the NAND_EVTER_STAT and PGRDCMPL_EVT_STAT registers.

17.3.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OPCEN	Reserved	PGRDCMPLEN	Reserved	FTOEREN	WPEREN	ECCEREN	Reserved								
Reset	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.53.4 Fields

Field	Function
31 OPCEN	NAND flash operation complete event enable 0b - NAND flash operation complete event is disabled 1b - NAND flash operation completed event is enabled
30 —	Reserved
29 PGRDCMPLEN	NAND flash page read completion event enable 0b - Per page read event disable 1b - Event will be generated on per page flash read operation completion. Status about the page whose data is available in sram buffer is given by PGRDCMPL_EVT_STAT register.
28 —	Reserved
27 FTOEREN	Flash time out error enable (for R/B timeoutand DQS timeout) 0b - Flash timeout error is disabled 1b - Flash timeout error is enabled
26	Write protect error checking enable.

Table continues on the next page...

Field	Function
WPEREN	0b - No write protect error checking 1b - Write protect error checking is enabled
25 ECCEREN	ECC error logging enable. 0b - ECCER event is not logged in NAND_EVTER_STAT register 1b - ECCER event is logged in NAND_EVTER_STAT register
24-0 —	Reserved

17.3.54 NAND Event and Error Interrupt Enable register (NAND_EVTER_INTR_EN)

17.3.54.1 Offset

Register	Offset
NAND_EVTER_INTR_EN	1_018Ch

17.3.54.2 Function

Event and Error Interrupt Enable Register (NAND_EVTER_INTR_EN) is used to enable/disable the generation of interrupt signals corresponding to error and event reporting. Software must clear pending events and errors in NAND_EVTER_STAT and PGRDCMPL_EVT_STAT register before enabling the interrupts.

17.3.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OPCIREN	Reserved	PGRDCMPLIREN	Reserved	FTOERIREN	WPERIREN	ECCERIREN	Reserved								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.54.4 Fields

Field	Function
31 OPCIREN	NAND flash operation complete event interrupt enable 0b - NAND flash operation Complete Event Interrupt is disabled 1b - NAND flash operation completed Event Interrupt is Enabled
30 —	Reserved
29 PGRDCMPLIREN	Page read completion event interrupt enable 0b - NAND flash page read event interrupt is disabled (per page basis) 1b - NAND flash page read event interrupt is enabled
28 —	Reserved
27 FTOERIREN	Flash timeout error interrupt enable (for R/B timeoutand DQS timeout) 0b - Flash timeout error interrupt is disabled 1b - Flash timeout error interrupt is enabled
26 WPERIREN	Write protect error interrupt enable 0b - Write protect error interrupt disable 1b - Write protect error interrupt is enabled
25 ECCERIREN	ECC error interrupt enable 0b - ECC Error Interrupt is disabled 1b - ECC Error Interrupt is enabled
24-0 —	Reserved

17.3.55 NAND Transfer Error Attributes register 0 (NAND_ERATTR0)

17.3.55.1 Offset

Register	Offset
NAND_ERATTR0	1_0198h

17.3.55.2 Function

Transfer error attribute register 0 (NAND_ERATTR0) is used to register the transaction attributes corresponding to the first error.

These registers store the attribute corresponding to the first transaction on which an error occured.

17.3.55.3 Diagram

17.3.55.4 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Field	Function
—	
28-26 ERCS	Chip-Select Corresponding to NAND Error 000b - Bank0 001b - Bank1 010b - Bank2 111b - Bank7
25-20 —	Reserved
19 ERTTYPE	Transaction type of NAND error 0b - Write 1b - Read
18-0 —	Reserved

17.3.56 NAND Transfer Error Attributes register 1 (NAND_ERATTR1)

17.3.56.1 Offset

Register	Offset
NAND_ERATTR1	1_019Ch

17.3.56.2 Function

Transfer error attribute register (NAND_ERATTR1) is used to register the transaction address corresponding to the first error.

17.3.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ER_ROWAD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ER_ROWAD															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.56.4 Fields

Field	Function
31-0 ER_ROWAD	Row address corresponding to error transaction

17.3.57 NAND Flash Status register (NAND_FSR)

17.3.57.1 Offset

Register	Offset
NAND_FSR	1_01E0h

17.3.57.2 Function

Flash status register (NAND_FSR) contains read status data from NAND flash.

17.3.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RS0								RS1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.57.4 Fields

Field	Function
31-24 RS0	First byte of data read from read status operation
23-16 RS1	Second byte of data read from read status operation
15-0 —	Reserved

17.3.58 ECC Status and Result of Flash Operation register 0 (ECCSTAT0)

17.3.58.1 Offset

Register	Offset
ECCSTAT0	1_01E8h

17.3.58.2 Function

ECC status registers are used to store the number of ECC errors occured on read operation. It has different meanings for 4/8 and 24/40 bit ECC Modes as the sector size is different for 4/8 and 24/40 bit modes.

ECC is computed on sector basis, and in the SRAM buffer there can be 32 such sectors of 512 byte each for 4 and 8 bit ECC (Galois field GF-2^{13}) and 8 sectors of 1 KB each for 24 and 40 bit ECC (Galois Field 2^{14}). Hence, each field of these registers represents number of errors in each sector. Depending on the page index mapped in the SRAM buffer, the corresponding field in the ECCSTAT0 will be updated.

As SRAM buffer can have 8 sectors of 1KB each, only the first eight register fields in ECCSTAT0/1 registers are defined with width 6 to represent 24 and 40 bit ECC along with 4 and 8 bit ECC.

As SRAM buffer can have 16 sectors of 1KB each, hence only first sixteen register fields in ECCSTAT0 register are defined with width 6 to represent 24 and 40 bit ECC along with 4 and 8 bit ECC.

NOTE

1. These registers will be updated whenever decoder is enable (ECC_DEC_EN =1b1) irrespective of ECC Error Event Indication is enabled or not (ECCEREN).
2. With the clearing of NAND_EVTER_STAT[ECCER] register bit the contents of ECCSTAT0/1/2/3 registers will also be cleared. These registers will also be cleared with soft_reset.
3. For every NAND sequence start, all the ECCSTAT registers will be cleared. It is implemented in order to flush the previously logged information.

17.3.58.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		NUMER0						Reserved		NUMER1					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NUMER2						Reserved		NUMER3					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.58.4 Fields

Field	Function
31-30 —	Reserved
29-24 NUMER0	Number of ECC errors on sector #0 of SRAM buffer For 4/8 bit ECC Mode (512 byte sector): 000000 No Error 000001 1 Bit Error 000010 2 Bit Error 000011 3 bit Error 000100 4 Bit Error 000101 5 Bit Error 000110 6 Bit Error 000111 7 BIt Error 001000 8 Bit Error 001111 - Uncorrectable Errors Others Reserved For 24/40 bit ECC Mode (1KB sector): 000000 No Error 000001 - 101000 1 Bit Error to 40 bit Error in this sector 111111 Uncorrectable Errors Others Reserved

Table continues on the next page...

Field	Function
23-22 —	Reserved
21-16 NUMER1	Number of ECC errors on sector #1 of SRAM buffer
15-14 —	Reserved
13-8 NUMER2	Number of ECC errors on sector #2 of SRAM buffer
7-6 —	Reserved
5-0 NUMER3	Number of ECC errors on sector #3 of SRAM buffer

17.3.59 ECC Status and Result of Flash Operation register 1 (ECCSTAT1)

17.3.59.1 Offset

Register	Offset
ECCSTAT1	1_01ECh

17.3.59.2 Function

ECC Status Register (ECCSTAT1) is used to store the number of ECC errors occured on Sector 4-7 during page read operation.

As SRAM buffer can have 16 sectors of 1KB each, hence only first sixteen register fields in ECCSTAT0 register are defined with width 6 to represent 24 and 40 bit ECC along with 4 and 8 bit ECC.

17.3.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		NUMER4						Reserved		NUMER5					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NUMER6						Reserved		NUMER7					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.59.4 Fields

Field	Function
31-30 —	Reserved
29-24 NUMER4	Number of ECC errors on sector #4 of SRAM buffer
23-22 —	Reserved
21-16 NUMER5	Number of ECC errors on sector #5 of SRAM buffer
15-14 —	Reserved
13-8 NUMER6	Number of ECC errors on sector #6 of SRAM buffer
7-6 —	Reserved
5-0 NUMER7	Number of ECC errors on sector #7 of SRAM buffer

17.3.60 ECC Status and Result of Flash Operation Register2 (ECCSTAT2)

17.3.60.1 Offset

Register	Offset
ECCSTAT2	1_01F0h

17.3.60.2 Function

ECC Status Register (ECCSTAT2) is used to store the number of ECC errors occured on Sector8-11 during page read operation.

17.3.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		NUMER8						Reserved		NUMER9					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NUMER10						Reserved		NUMER11					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.60.4 Fields

Field	Function
31-30 —	Reserved
29-24	Number of ECC errors on sector #8 of SRAM buffer

Table continues on the next page...

Field	Function
NUMER8	
23-22 —	Reserved
21-16 NUMER9	Number of ECC errors on sector #9 of SRAM buffer
15-14 —	Reserved
13-8 NUMER10	Number of ECC errors on sector #10 of SRAM buffer
7-6 —	Reserved
5-0 NUMER11	Number of ECC errors on sector #11 of SRAM buffer

17.3.61 ECC Status and Result of Flash Operation Register3 (ECCSTAT3)

17.3.61.1 Offset

Register	Offset
ECCSTAT3	1_01F4h

17.3.61.2 Function

ECC Status Register (ECCSTAT3) is used to store the number of ECC errors occured on Sector12-15 during page read operation.

17.3.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		NUMER12						Reserved		NUMER13					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NUMER14						Reserved		NUMER15					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.61.4 Fields

Field	Function
31-30 —	Reserved
29-24 NUMER12	Number of ECC errors on sector #12 of SRAM buffer
23-22 —	Reserved
21-16 NUMER13	Number of ECC errors on sector #13 of SRAM buffer
15-14 —	Reserved
13-8 NUMER14	Number of ECC errors on sector #14 of SRAM buffer
7-6 —	Reserved
5-0 NUMER15	Number of ECC errors on sector #15 of SRAM buffer

17.3.62 ECC Status and Result of Flash Operation Register4 (ECCSTAT4)

17.3.62.1 Offset

Register	Offset
ECCSTAT4	1_01F8h

17.3.62.2 Function

ECC Status Register (ECCSTAT4) is used to store the number of ECC errors occured on Sector16-19 during page read operation.

17.3.62.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				NUMER16				Reserved				NUMER17			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				NUMER18				Reserved				NUMER19			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.62.4 Fields

Field	Function
31-28 —	Reserved
27-24 NUMER16	Number of ECC errors on sector #16 of SRAM buffer
23-20 —	Reserved
19-16	Number of ECC errors on sector #17 of SRAM buffer

Table continues on the next page...

Field	Function
NUMER17	
15-12 —	Reserved
11-8 NUMER18	Number of ECC errors on sector #18 of SRAM buffer
7-4 —	Reserved
3-0 NUMER19	Number of ECC errors on sector #19 of SRAM buffer

17.3.63 ECC Status and Result of Flash Operation Register5 (ECCSTAT5)

17.3.63.1 Offset

Register	Offset
ECCSTAT5	1_01FCh

17.3.63.2 Function

ECC Status Register (ECCSTAT5) is used to store the number of ECC errors occured on Sector20-23 during page read operation.

17.3.63.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				NUMER20				Reserved				NUMER21			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				NUMER22				Reserved				NUMER23			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.63.4 Fields

Field	Function
31-28 —	Reserved
27-24 NUMER20	Number of ECC errors on sector #20 of SRAM buffer
23-20 —	Reserved
19-16 NUMER21	Number of ECC errors on sector #21 of SRAM buffer
15-12 —	Reserved
11-8 NUMER22	Number of ECC errors on sector #22 of SRAM buffer
7-4 —	Reserved
3-0 NUMER23	Number of ECC errors on sector #23 of SRAM buffer

17.3.64 ECC Status and Result of Flash Operation Register6 (ECCSTAT6)

17.3.64.1 Offset

Register	Offset
ECCSTAT6	1_0200h

17.3.64.2 Function

ECC Status Register (ECCSTAT6) is used to store the number of ECC errors occured on Sector24-24 during page read operation.

17.3.64.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				NUMER24				Reserved				NUMER25			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				NUMER26				Reserved				NUMER27			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.64.4 Fields

Field	Function
31-28 —	Reserved
27-24 NUMER24	Number of ECC errors on sector #24 of SRAM buffer
23-20 —	Reserved
19-16 NUMER25	Number of ECC errors on sector #125of SRAM buffer
15-12 —	Reserved
11-8 NUMER26	Number of ECC errors on sector #26 of SRAM buffer
7-4 —	Reserved
3-0 NUMER27	Number of ECC errors on sector #27 of SRAM buffer

17.3.65 ECC Status and Result of Flash Operation Register7 (ECCSTAT7)

17.3.65.1 Offset

Register	Offset
ECCSTAT7	1_0204h

17.3.65.2 Function

ECC Status Register (ECCSTAT7) is used to store the number of ECC errors occured on Sector28-31 during page read operation.

17.3.65.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				NUMER28				Reserved				NUMER29			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				NUMER30				Reserved				NUMER31			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.65.4 Fields

Field	Function
31-28 —	Reserved
27-24 NUMER28	Number of ECC errors on sector #28 of SRAM buffer
23-20 —	Reserved
19-16 NUMER29	Number of ECC errors on sector #29 of SRAM buffer
15-12 —	Reserved
11-8	Number of ECC errors on sector #30 of SRAM buffer

Table continues on the next page...

Field	Function
NUMER30	
7-4 —	Reserved
3-0 NUMER31	Number of ECC errors on sector #31 of SRAM buffer

17.3.66 NAND Control register (NANDCR)

17.3.66.1 Offset

Register	Offset
NANDCR	1_0278h

17.3.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			FTOCNT				Reserved								
Reset	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.66.3 Fields

Field	Function
31-29 —	Reserved
28-25 FTOCNT	Flash timeout count 0000b - 256 cycles of IFC module input clock 0001b - 512 cycles of IFC module input clock

Table continues on the next page...

Field	Function
	0010b - 1024 cycles of IFC module input clock 0011b - 2048 cycles of IFC module input clock 0100b - 4096 cycles of IFC module input clock 0101b - 8192 cycles of IFC module input clock 0110b - 16,384 cycles of IFC module input clock 0111b - 32,768 cycles of IFC module input clock 1000b - 65,536 cycles of IFC module input clock 1001b - 131,072 cycles of IFC module input clock 1010b - 262,144 cycles of IFC module input clock 1011b - 524,288 cycles of IFC module input clock 1100b - 1,048,576 cycles of IFC module input clock 1101b - 2,097,152 cycles of IFC module input clock 1110b - 4,194,304 cycles of IFC module input clock 1111b - 8,388,608 cycles of IFC module input clock
24-0 —	Reserved

17.3.67 NAND Autoboot Trigger register (NAND_AUTOBOOT_TRGR)

17.3.67.1 Offset

Register	Offset
NAND_AUTOBOOT_TRGR	1_0284h

17.3.67.2 Function

This register is used to trigger the booting on the NAND flash. This register is provided if por_cfg_rcw_load and por_cfg_boot_load signals are not coming through the pins for the booting.

It is a self-clearing register; hence, the user just has to write 1 in appropriate field to trigger the auto-boot operation and hardware will clear it.

17.3.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		Reserved	0	Reserved												
W	RCW_LD		BOOT_LD													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.67.4 Fields

Field	Function
31 RCW_LD	RCW Load **NOTE:** The values on the various por_cfg pins should be driven valid when this bit is set. 0b - No RCW loading 1b - Trigger RCW load from NAND flash (same functionality that can be achieved by sending pulse on por_cfg_rcw_load)
30 —	Reserved
29 BOOT_LD	BOOT Load **NOTE:** The values on the various por_cfg pins should be driven valid when this bit is set. 0b - No BOOT loading 1b - Trigger autoboot loading from NAND flash (same functionality that can be achieved by sending pulse on por_cfg_boot_load)
28-0 —	Reserved

17.3.68 NAND Flash Memory Data register (NAND_MDR)

17.3.68.1 Offset

Register	Offset
NAND_MDR	1_028Ch

17.3.68.2 Function

This register is used to store one beat of data read from NAND flash memory when opcode, "SBRD" is used in the FIR register. This register is provided for reading 1 or 2 bytes of data from 8- or 16-bit NAND flash. The location from where the data gets fetched can be controlled by column register (COL*n*). When NAND FIR is programmed with opcode "SBRD", IFC always reads 1/2 bytes of data irrespective of the value of BC (no. of bytes) programmed in NAND_BC register.

Also note that the read data fetched from NAND flash is only stored in this register and SRAM buffer does not get updated as this opcode is intended for register access only.

17.3.68.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDATA0								RDATA1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.68.4 Fields

Field	Function
31-24 RDATA0	1st read data byte when opcode SBRD is used for read
23-16 RDATA1	Second read data byte when opcode SBRD is used for read (only valid for 16 bit NAND flash)

Table continues on the next page...

Field	Function
15-0 —	Reserved

17.3.69 Nand DLL Low Config 0 Register (NAND_DLL_LOW_CFG0)

17.3.69.1 Offset

Register	Offset
NAND_DLL_LOW_CFG0	1_0300h

17.3.69.2 Function

This registers is used to configure the DLL used to facilitate the shifiting on the incoming DQS to the centre of the data eye during read operations. This register is used to configure the DLL for interface frequency up to 133 MHz (based on CSOR_EXT[MODE_FREQ])

17.3.69.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLL_ENABLE	DLL_RESET	Reserved													
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.69.4 Fields

Field	Function
31 DLL_ENABLE	This bit Enables the DLL. 0b - DLL disabled. 1b - DLL enabled.
30 DLL_RESET	This bit resets the DLL. When this bit is set, TAP 0 for both the master and the slave delay lines gets selected. 0b - DLL is not reset. 1b - DLL is reset.
29-0 —	Reserved

17.3.70 Nand DLL Low Config 1 Register (NAND_DLL_LOW_CFG1)

17.3.70.1 Offset

Register	Offset
NAND_DLL_LOW_CFG1	1_0304h

17.3.70.2 Function

This registers is used to configure the DLL used to facilitate shifiting of the incoming DQS to the centre of the data eye during read operations.This register is used to configure the DLL for interface frequency up to 133 Mhz (based on CSOR_EXT[MODE_FREQ])

17.3.70.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLL_PD_PULSE_STRETCH_SEL	Reserved											DLL_REF_UPDATE_INT			
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DLL_SLV_UPDATE_INT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.70.4 Fields

Field	Function
31 DLL_PD_PULSE_STRETCH_SEL	Used to select between a 2 delay cell or 4 delays cells for reliable detection of pulse width by the pulse width detection logic. 0b - 4 delay cell selected. 1b - 2 delay cell selected.
30-20 —	Reserved
19-16 DLL_REF_UPDATE_INT	This field is used to override the default update interval of the reference delay line. The default value of the update interval is 2 REF_CLK cycles. When this field is programmed, the value of the update interval is 2 + DLL_REF_UPDATE_INT 0000 - Update interval value is 2 + 0 REF clock cycles. 0001 - Update interval value is 2 + 1 REF clock cycles. 0010 - Update interval value is 2 + 2 REF clock cycles. 0011 - Update interval value is 2 + 3 REF clock cycles. ... 1111 - Update interval value is 2 + 15 REF clock cycles.
15-8 —	Reserved

Table continues on the next page...

Field	Function
7-0 DLL_SLV_UPDATE_INT	This field is used to override the default update interval of the slave delay line. The default value of the update interval is 256 REF_CLK cycles. This field is valid only if a non zero value is programmed. 00000000- Update interval value is 256 REF clock cycles. 00000001- Update interval value is 1 REF clock cycles. 00000010- Update interval value is 2 REF clock cycles. 00000011- Update interval value is 3 REF clock cycles. ... 11111111- Update interval value is 255 REF clock cycles.

17.3.71 NAND DLL Low Status Register (NAND_DLL_LOW_STAT)

17.3.71.1 Offset

Register	Offset
NAND_DLL_LOW_STAT	1_030Ch

17.3.71.2 Function

This register is used to store the status of the DLL for interface frequency up to 133 MHz

17.3.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLL_STS_REF_LOCK	Reserved			DLL_STS_SLV_LOCK	Reserved							DLL_STS_REF_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLL_STS_REF_SEL				Reserved				DLL_STS_SLV_SEL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.71.4 Fields

Field	Function
31 DLL_STS_REF_LOCK	DLL Reference Delay line lock status
30-28 —	Reserved
27 DLL_STS_SLV_LOCK	DLL Slave Delay Chain lock status
26-20 —	Reserved
19-12 DLL_STS_REF_SEL	Status of selected tap for reference delay line
11-8 —	Reserved

Table continues on the next page...

Field	Function
7-0 DLL_STS_SLV_SEL	Status of selected tap for slave delay line

17.3.72 NOR Event and Error Status register (NOR_EVTER_STAT)

17.3.72.1 Offset

Register	Offset
NOR_EVTER_STAT	1_0400h

17.3.72.2 Function

As with NAND FCM, 1 KB memory-mapped region is allocated for the NOR-specific registers. This section describes these registers.

NOR event and error status register (NOR_EVTER_STAT) indicates the cause of an error or event corresponding to NOR flash.

The following figure shows the register fields. It is write-1-to-clear register.

17.3.72.3 Diagram

17.3.72.4 Fields

Field	Function
31 OPC_NOR	NOR Command Sequence Operation Complete Event Indication **NOTE:** Not valid in case of read operations which do not require a command sequence to be performed 0b - NOR Command Sequence Operation not completed 1b - Indicates the completion of a Command Sequence performed on the NOR flash device. A command sequence is said to have completed once all the phases (as indicated in the NORCR[NUM_PHASE]) have been sent to the NOR flash device.
30-27 —	Reserved
26 WPER	Write Protect Error 0b - No write protect error 1b - A write is attempted to the memory bank which is write protected
25 —	Reserved
24 STOER	Command sequence timeout error 0b - No command sequence timeout from system side 1b - Command sequence timeout from system side
23-0 —	Reserved

17.3.73 NOR Event and Error Enable register (NOR_EVTER_EN)

17.3.73.1 Offset

Register	Offset
NOR_EVTER_EN	1_040Ch

17.3.73.2 Function

Event and Error Enable Register (NOR_EVTER_EN) is used to enable/disable the logging of event and error indication in the NOR_EVTER_STAT register.

17.3.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OPCEN_NOR	Reserved				WPEREN	Reserved	STOEREN	Reserved							
Reset	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.73.4 Fields

Field	Function
31 OPCEN_NOR	NOR Command Sequence operation complete event enable 0b - OPC_NOR Event is not enabled 1b - OPC_NOR Event is enabled
30-27 —	Reserved
26 WPEREN	Write Protect Error Checking Enable 0b - No write protect error checking 1b - Write protect error checking is enabled
25 —	Reserved
24 STOEREN	Command Sequence Timeout Error Enable 0b - No command sequence timeout checking 1b - Command sequence timeout error checking enable
23-0 —	Reserved

17.3.74 NOR Event and Error Interrupt enable register (NOR_EVTER_INTR_EN)

17.3.74.1 Offset

Register	Offset
NOR_EVTER_INTR_EN	1_0418h

17.3.74.2 Function

Event and Error Interrupt Enable Register (NOR_EVTER_INTR_EN) is used to enable/disable the generation of interrupt signals corresponding to error and event reporting. Software must clear pending events and errors in NOR_EVTER_STAT register before enabling the interrupts.

17.3.74.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OPCIREN_NOR	Reserved				WPERIREN	Reserved	STOERIREN	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.74.4 Fields

Field	Function
31 OPCIREN_NOR	NOR command sequence operation complete event Interrupt enable 0b - Event Interrupt is not enabled 1b - Event Interrupt is enabled
30-27 —	Reserved
26	Write protect error interrupt enable

Table continues on the next page...

Field	Function
WPERIREN	0b - Write protect error interrupt disable 1b - Write protect error interrupt is enabled
25 —	Reserved
24 STOERIREN	NOR command sequence timeout error interrupt enable 0b - Command Sequence Timeout error interrupt disable 1b - Command Sequence Timeout error interrupt is enabled
23-0 —	Reserved

17.3.75 NOR Transfer Error Attributes register 0 (NOR_ERATTR0)

17.3.75.1 Offset

Register	Offset
NOR_ERATTR0	1_0424h

17.3.75.2 Function

These registers store the attribute corresponding to the first transaction on which error occured.

Transfer error attribute register 0 (NOR_ERATTR0) is used to register the transaction attributes corresponding to the first error.

17.3.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERSRCID								Reserved				ERAID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERAID				Reserved			ERCS			Reserved					ERTYPE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.75.4 Fields

Field	Function
31-24 ERSRCID	SRCID corresponding to error transaction
23-20 —	Reserved
19-12 ERAID	ID of the error transaction. This field is valid only for write protect error.
11-9 —	Reserved
8-6 ERCS	Chip-select corresponding to NOR error 000b - Bank0 001b - Bank1 010b - Bank2 111b - Bank7
5-1 —	Reserved
0 ERTYPE	Type of the transaction 0b - Write 1b - Read

17.3.76 NOR Transfer Error Attribute register 1 (NOR_ERATTR1)

17.3.76.1 Offset

Register	Offset
NOR_ERATTR1	1_0428h

17.3.76.2 Function

Transfer error attribute register (NOR_ERATTR1) is used to register the transaction address corresponding to the first error.

17.3.76.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.76.4 Fields

Field	Function
31-0 ERADDR	In the case of a write protect error, this field refelcts the address corresponding to which a write cycle was received from the system side. In the case of a sequence timeout error, this field reflects the last address received from the system side

17.3.77 NOR Transfer Error Attribute register 2 (NOR_ERATTR2)

17.3.77.1 Offset

Register	Offset
NOR_ERATTR2	1_042Ch

17.3.77.2 Function

Transfer error attribute register (NOR_ERATTR2) is used to register the transaction SRCID corresponding to the first error.

17.3.77.3 Diagram

17.3.77.4 Fields

Field	Function
31-20	Reserved

Table continues on the next page...

Field	Function
—	
19-16 ER_NUM_PHASE_EXP	Number of phase expected in command sequence which timed-out by system side. **NOTE:** This attribute is valid only for sequence timeout error. 0000b - 1 Phase 0001b - 2 Phase 1111b - 16 Phases
15-12 —	Reserved
11-8 ER_NUM_PHASE_PER	Actual no. of command sequence phases performed on NOR flash before timeout occured **NOTE:** This attribute is valid only for sequence timeout error 0000b - 0 Phase 0001b - 1 Phase 1111b - 15 Phases
7-0 —	Reserved

17.3.78 NOR Control register (NORCR)

17.3.78.1 Offset

Register	Offset
NORCR	1_0440h

17.3.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				NUM_PHASE				Reserved				STOCNT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.78.3 Fields

Field	Function
31-28 —	Reserved
27-24 NUM_PHASE	No. of Address/Data phase on device for which chip-select has to be asserted: This field is used for the NOR devices for which CS has to be asserted for multiple address data cycles. It can be used for command based NOR, where CS remains asserted during unlock cycles and actual address/data phase. For example, if the total number of phases (that is, command and address/data) to be sent corresponding to a particular operation (such as program, block erase, and chip erase) is 5, then this field should be programmed to 0100. This would indicate to the NOR FCM that the CE needs to be asserted for five consecutive commands sent from the system side (there by for a complete operation) 0000b - 1 phase 0001b - 2 phase 0010b - 3 phase 0011b - 4 phase 0100b - 5 phase 0101b - 6 phase 0110b - 7 phase 0111b - 8 phase 1000b - 9 phase 1001b - 10 phase 1010b - 11 phase 1011b - 12 phase 1100b - 13 phase 1101b - 14 phase 1110b - 15 phase 1111b - 16 phase
23-20 —	Reserved
19-16 STOCNT	Sequence Timeout Count **NOTE:** This counter is used for timeout on sequential transactions on command based NOR. 0000b - 256 cycles of IFC module input clock 0001b - 512 cycles of IFC module input clock 0010b - 1024 cycles of IFC module input clock 0011b - 2048 cycles of IFC module input clock 0100b - 4096 cycles of IFC module input clock 0101b - 8192 cycles of IFC module input clock 0110b - 16,384 cycles of IFC module input clock 0111b - 32,768 cycles of IFC module input clock 1000b - 65,536 cycles of IFC module input clock 1001b - 131,072 cycles of IFC module input clock 1010b - 262,144 cycles of IFC module input clock 1011b - 524,288 cycles of IFC module input clock 1100b - 1,048,576 cycles of IFC module input clock 1101b - 2,097,152 cycles of IFC module input clock 1110b - 4,194,304 cycles of IFC module input clock 1111b - 8,388,608 cycles of IFC module input clock
15-0 —	Reserved

17.3.79 GPCM Event and Error Status register (GPCM_EVTER_STAT)

17.3.79.1 Offset

Register	Offset
GPCM_EVTER_STAT	1_0800h

17.3.79.2 Function

A separate 1 KB memory-mapped region is allocated for the GPCM specific registers.

GPCM Event and Error Status Register (GPCM_EVTER_STAT) indicates the cause of an error or event corresponding to GPCM flash devices.

17.3.79.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					TOER	Reserved	PER	Reserved	ABER	Reserved					
W						W1C		W1C		W1C						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.79.4 Fields

Field	Function
31-27 —	Reserved

Table continues on the next page...

Field	Function
26 TOER	Timeout Error **NOTE:** Timeout for Normal GPCM mode will only be observed if IFCTA_B is configured as acknowledgement signal for transaction access completion, that is, by setting 1 in CSOR*n*[WGETA] and CSOR*n*[RGETA]. Timeout error will occur if IFC is waiting for the acknowledgement signal IFCTA_B to come and timeout counter value specified in CSOR[GPTO] counter has expired 0b - No Timeout Error 1b - Timeout observed for read/write transaction.
25 —	Reserved
24 PER	Parity Error 0b - No Parity Error 1b - Parity Error for read/write transaction.
23 —	Reserved
22 ABER	Abort Error. Abort for Normal GPCM mode will only be observed if access is terminated by IFCTA_B when CSOR*n*[RGETA] is programmed to 0. It is valid for read transaction and no error is reported for write transaction. 0b - No abort error 1b - Abort happened for the current read transaction
21-0 —	Reserved

17.3.80 GPCM Event and Error Enable register (GPCM_EVTER_EN)

17.3.80.1 Offset

Register	Offset
GPCM_EVTER_EN	1_080Ch

17.3.80.2 Function

Event and Error Enable Register (GPCM_EVTER_EN) is used to enable/disable the logging of event and error indication in the GPCM_EVTER_STAT register.

17.3.80.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					TOEREN	Reserved	PEREN	Reserved	ABEREN	Reserved					
Reset	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.80.4 Fields

Field	Function
31-27 —	Reserved
26 TOEREN	Timeout error checking enable 0b - No Timeout Error checking 1b - Timeout Error checking is enabled
25 —	Reserved
24 PEREN	Parity error checking enable. 0b - No Parity Error Checking 1b - Parity Error checking Enabled
23 —	Reserved
22 ABEREN	Abort Error Checking Enable 0b - No Abort Error checking 1b - Abort Error checking is enabled
21-0 —	Reserved

17.3.81 GPCM Event and Error Interrupt enable register (GPCM_EVTER_INTR_EN)

17.3.81.1 Offset

Register	Offset
GPCM_EVTER_INTR_EN	1_0818h

17.3.81.2 Function

Event and Error Interrupt Enable Register (GPCM_EVTER_INTR_EN) is used to enable/disable the generation of interrupt signals corresponding to error and event reporting. Software must clear pending events and errors in GPCM_EVTER_STAT register before enabling the interrupts.

17.3.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					TOERIREN	Reserved	PERIREN	Reserved	ABERIREN	Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.81.4 Fields

Field	Function
31-27 —	Reserved
26 TOERIREN	Tmeout Error Interrupt Enable 0b - Timeout Error interrupt disable 1b - Timeout Error interrupt is enabled
25 —	Reserved

Table continues on the next page...

Field	Function
24 PERIREN	Parity Error Interrupt Enable 0b - Parity Error interrupt disable 1b - Parity Error interrupt is enabled
23 —	Reserved
22 ABERIREN	Abort Error Interrupt Enable 0b - Abort Error interrupt disable 1b - Abort Error interrupt is enabled
21-0 —	Reserved

17.3.82 GPCM Transfer Error Attributes register 0 (GPCM_ERATTR0)

17.3.82.1 Offset

Register	Offset
GPCM_ERATTR0	1_0824h

17.3.82.2 Function

Transfer error attribute register 0 (GPCM_ERATTR0) is used to register the transaction attributes corresponding to error occured.

17.3.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERSRCID								Reserved				ERAID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERAID				Reserved			ERCS			Reserved					ERTYPE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.82.4 Fields

Field	Function
31-24 ERSRCID	SRCID corresponding to error transaction
23-20 —	Reserved
19-12 ERAID	ID of the error transaction
11-9 —	Reserved
8-6 ERCS	ERCS 000b - Bank0 001b - Bank1 010b - Bank2 111b - Bank7
5-1 —	Reserved
0 ERTYPE	Type of the transaction 0b - Write 1b - Read

17.3.83 GPCM Transfer Error Attributes register 1 (GPCM_ERATTR1)

17.3.83.1 Offset

Register	Offset
GPCM_ERATTR1	1_0828h

17.3.83.2 Function

Transfer error attribute register (GPCM_ERATTR1) is used to register the transaction address corresponding to error occured.

17.3.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.83.4 Fields

Field	Function
31-0 ERADDR	Address corresponding to error transaction

17.3.84 GPCM Transfer Error Attributes register 2 (GPCM_ERATTR2)

17.3.84.1 Offset

Register	Offset
GPCM_ERATTR2	1_082Ch

17.3.84.2 Function

Transfer error attribute register (GPCM_ERATTR2) is used to register the transaction attributes corresponding to error occured.

17.3.84.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				PERR_BEAT		Reserved		PERR_BYTE				Reserved			PERR_AD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.84.4 Fields

Field	Function
31-12 —	Reserved
11-10 PERR_BEAT	This gives information on which beat of address/data parity error is observed. This has value from 0 to 3. For example, if port size is 16 bit and parity error is observed on second beat, then value 1 is reported in the status register. This field is valid for Generic ASIC mode.
9-8 —	Reserved

Table continues on the next page...

Field	Function
7-4 PERR_BYTE	Parity Error on byte. For GPCM there are four parity error status bit, one per byte. A bit is set for the byte that has parity error (bit 24 represents byte 0, the most significant byte lane). This field is valid for Normal GPCM mode.
3-1 —	Reserved
0 PERR_AD	Parity Error reported in address or data phase. Address phase is only valid for Generic ASIC mode of GPCM. 0b - Address phase 1b - Data Phase

17.3.85 GPCM Status register (GPCM_STAT)

17.3.85.1 Offset

Register	Offset
GPCM_STAT	1_0830h

17.3.85.2 Function

This register is used to reflect the busy status of the GPCM controller.

17.3.85.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GPCM_BSY	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

17.3.85.4 Fields

Field	Function
31 GPCM_BSY	GPCM_BSY 0b - No transaction being done by GPCM controller. 1b - GPCM controller is busy with transactions.
30-0 —	Reserved

17.4 IFC functional description

The IFC is used to interface with external NAND flash, NOR flash, SRAM, EPROM, and generic ASIC devices.

To achieve this functionality, the logic is partitioned in independent flash control machines (FCMs) in the IFC to control the timings of NAND flash, NOR flash, and GPCM separately. Eight independent chip-selects are provided, but they all share the same pins; hence, only one memory can be accessed at a time based on the machine-select bits of the chip-select property register for that bank (CSPR*n*[MSEL]). If a bank match occurs, the corresponding machine (NAND, NOR, or GPCM) takes ownership of the external signals that control the access and maintains control until the transaction ends.

This figure is a functional block diagram of the flash memory interface and its signal muxing.

Figure 17-4. Flash memory interface and muxing (with NVDDR NAND)

The IFC supports the following types of flash control machines (FCMs):

- NOR FCM provides interfacing with NOR flash memories that have a 8-/16-bit-wide data bus. NOR FCM-controlled banks are used primarily for booting (direct memory-mapped) and code storage.
- The NAND FCM interfaces to NAND flash EEPROMs with 8- and 16-bit data buses. If NAND is chosen as the booting device, after reset, the NAND FCM can load boot code into SRAM buffer for execution. Following boot, the NAND FCM provides a flexible instruction sequencer that allows a user-defined command, address, and data transfer sequence of up to 15 steps to be executed against a memory-mapped buffer RAM. An advance ECC algorithm (BCH codes) is implemented to correct up to 4-/8-bit errors per sector of 512 bytes and 24/40 bit erros per 1KB sector.
- GPCM provides an interface to simple, synchronous, memory-mapped devices.

Each memory bank can be assigned to any of these types of machines through the machine-select bits of the chip-select property register for that chip-select (CSPRn[MSEL]).

17.4.1 General architecture

The basic architecture of the IFC allows bank selection through address decoding and address/data pin muxing.

17.4.1.1 Bank selection through address decoding

Banks can be selected via address decoding.

The defined base addresses are written to CSPR*n*[BA] and CSPR*n*_EXT[BA_EXT], while the corresponding address masks are written to AMASK*n*[AM]. Each time a local system access is requested, the internal transaction address is compared with each bank. Address decoding logic is explained in Address Mask register (AMASK0 - AMASK7). If a match is found on a memory controller bank, the attributes defined in the CSPR*n* and CSOR*n* for that bank are used to control the memory access. If a match is found in more than one bank, the lowest-numbered bank handles the memory access (that is, bank 0 has priority over bank 1).

17.4.1.2 Address Data Pin Multiplexing

This section describes the address data pin muxing scheme at IFC interface.

IFC provides muxing of address and data bits on AD bus and the muxing is controlled by signal address latch enable (ALE). There are two modes supported to supply either address msb (most significant bits) or lsb (least significant bits) on AD bus. Register field CSORn[ADM_SHFT_MODE] determines the mode of address-data pin muxing.

Figure 17-5. Address/Data Multiplexing

- AD[0] is the least significant address bit of muxed address carried by the AD bus and AD[M-1] is the most significant bitm where M is determined by the chip's AD configuration.
- ADDR[0] is the least significant bit of unmuxed address carried by the ADDR bus and ADDR[N-1] is the most significant address bit, where the value of N is chosen by the chip depending on its address requirement.

17.4.1.3 Mode-0 Pin Muxing (CSOR[ADM_SHFT_MODE] = 0)

The application of this mode is in an external latched based system (Unmuxed Parallel NOR/GPCM) where ALE is connected to latch enable of external latch where the most significant address bits will be latched. This mode is used to connect parallel NOR devices (having dedicated Address and data pins) as well as GPCM based interfaces.

In this mode, system address gets right shifted by CSORn[ADM_SHFT] shift value and assigned to AD[M-1:0]. The value of ADM_SHFT should be programmed based on number of dedicated address pins (N) on the chip. Shifted address is latched by external latch at the falling edge of ALE.

Address lsb's can be picked from ADDR bus where ADDR[0] carries the least significant address bit of system address.

As ADDR[0] carries the byte address to interface with x16 NOR device, the address lsb, that is, ADDR[0] must be left unconnected on the board as x16 NOR device expects word address.

Example-

SoC has exposed 16 bit AD bus and 10 bit ADDR. Conventional NOR of various sizes are interfaced through it. ADM_SHFT value of 10 should be programmed as SoC has exposed 10 ADDR pins. The IFC right shifts the system address by 10 positions to provide 16 address most significant bits on AD bus.

Figure 17-6. Conventional NOR connection on x16 AD SoC

17.4.1.4 Mode-1 Pin Muxing (CSOR[ADM_SHFT_MODE] = 1)

The application of this mode is in Address Data Multiplexed (ADM) NOR devices. They expect address least significant bits on AD bus during AVD_B assertion. This mode is not supported by GPCM.

In this mode as AD pins can't be left unconnected on board, and IFC will do word alignment (based on memory's port size) of system address. Word aligned sytem address gets right shifted by CSORn[ADM_SHFT] value and assigned to ADDR[N-1:0] bus.

The ADM_SHFT value should be same as number of AD pins (N) exposed by SoC except x32 AD case where this value should be 0.

Example-

SoC has exposed 16bit AD bus. ADM NOR of various sizes are interfaced through it. ADM_SHFT value 16 should be programmed as SoC has exposed 16 AD bus. IFC right shifts system address by value 16 to provide 16 address least significant bits (word aligned) on AD bus.

Figure 17-7. ADM NOR connectivity on x16 SoC

17.4.2 Programming model for flash interface timing

The IFC follows a counter-based approach to generate the required timings on the flash side.

Programming registers (FTIM*n*) are provided that correspond to the various flash timing parameters. The values to be programmed in these registers are in terms of the number of IFC input clock-cycles derived by the following formula:

Timing counter value = [{Flash Timing Parameter(ns) x IFC module input clock frequency (MHz)} + 1000 - 1] / 1000

In this approach, counters running at the IFC module input clock generate the appropriate values for the flash interface signals depending on the value programmed in the Timing Registers (FTIM0-FTIM3). These timing registers must be programmed per the values specified in the flash device datasheet mapped to the waveforms shown in NAND asynchronous mode timings, Unmuxed (parallel) asynchronous NOR read timings, and Simple asynchronous NOR write timings.

17.4.3 Booting methods

Booting can be performed from NAND or NOR flash on the IFC.

- NOR is directly memory-mapped, so booting from NOR can be done as simple read operations without any special arrangements. Default timing parameters are loaded with the assertion of rcw_load/boot_load, details of this requirement is explained in NOR boot.
- NAND is not directly memory-mapped, so the boot code is loaded first into an SRAM buffer from where the code can be executed. See NAND flash control machine for more information about booting from NAND.

The flash timing registers that control chip-select 0 have default values corresponding to the slowest-mode devices to guarantee boot.

Reset initialization is performed by the reset controller and the default configuration is determined by cfg_rcw_src and/or RCW settings.

17.4.4 Software reset handling

Software reset requires that software read and clear specific registers.

Software follows these steps as part of software reset handling:

1. Read all the event and error status registers.
2. Clear the NANDSEQ_STRT register.
3. Clear the NAND_AUTOBOOT_TRGR register.
4. Apply soft reset.

After the soft reset has been applied, the hardware automatically clears all the event and error status registers. None of the other registers are reset. Therefore, the initial configuration done by the software is preserved.

17.4.4.1 System bus handling

Any pending transactions on the system bus are terminated with a valid response.

During software reset assertion, providing a good response that corresponds to the pending transaction on the IFC helps prevent a corresponding interrupt. The core expects an error interrupt whenever there is an error response from the system bus. Therefore, in the case of a software reset and an error response is provided, the core goes into a hang state because there is no corresponding interrupt. If there are pending read transactions, the garbage data is supplied back; if there is a pending write transaction, the data is not written in the memory.

The hardware does not allow the software to clear the soft reset bit until all pending transactions on the system bus have been gracefully terminated (with a good response). When there are no more transactions pending on the system bus, the hardware allows the software to clear the soft reset bit.

17.4.4.2 Flash interface handling

All the state machines (NAND, NOR, GPCM, and GASIC) are reset to the idle state.

All related FIFOs and registers are flushed, so the flash interface would be in the idle state.

17.4.5 Data buffer control (BCTL)

The IFC provides a data buffer control signal (BCTL) that can be disabled (remain high) by setting CSOR*n*[BCTLD]. BCTL should be used to signify the write direction when high.

NOTE

BCTL is a static value; it cannot be programmed on the fly.

If the access is a write, BCTL remains high for the whole duration. However, if the access is a read, BCTL is negated (low) so that the memory device is able to drive the bus. Note that the default (reset and bus idle) value of BCTL is also high.

While accessing slow memories, the data driven by the flash is available on the bus after deassertion of read enable. To avoid bus contention, BCTL remains low (read mode) for the time defined by the CSORn[TRHZ] field.

Apart from CSORn[TRHZ], another signal timing that requires attention during the external buffer control is buffer turn-around. The external buffer takes time to reverse the direction of the shared I/O bus. This situation is more relevant when a read is followed by a write. To handle this, GCR[TBCTL_TRN_TIME] is defined, which represents the time for which BCTL remains high before starting another flash access. The timings are shown in the following figures.

Figure 17-8. BCTL signal in NAND read followed by any other operation

Figure 17-9. BCTL signal in NOR read followed by any other operation

17.4.6 External transceiver enable (TE)

This signal is used to enable/disable the external transceiver depending on whether a slow/fast memory is connected to the bank servicing the current request.

If various fast/slow memories are connected to different banks of the IFC, the AD bus may become loaded due to sharing. Such a scenario results in poor rise/fall times and therefore limits the speed of operation of fast memories given their strict timing requirements. To address this issue, fast memories should be segregated from the slower memories using an external transceiver that is connected to the AD bus. In these scenarios, the IFC provides the configurable transceiver enable (TE) pin.

17.4.6.1 Transceiver enable during boot

The IFC can be used for booting after reset.

The IFC can obtain the ifc_TE value, which is used during booting, in the following ways:

- By sampling the TE pin (booting from slow devices connected through transceiver).

  In this scenario, the TE pin acts as an input during reset. During this time, the external transceiver should not drive anything onto the AD bus (that is, keep it tristated). This is accomplished by connecting a weak pull up/down resistor on the

TE pin so that the external transceiver is disabled. With rcw_load/boot_load, the IFC samples the TE pin in order to know the polarity of external transceiver's enable pin. With rcw_load/boot_load, the value stored in CSPR0[TE] as the default value will be opposite of the polarity of external transceiver's enable pin. Therefore, the polarity of external transceiver's enable pin is configurable with the help of pull up/down resistors.

Table 17-6. Programming of CSPRn[TE]

Transceiver Enable (TE) Input	Value of CSPRn[TE]	
	Slow Memory	Fast Memory
Active low	0	1
Active high	1	0

- By the chip booting from fast/slow device

  If boot source is a fast device and connected directly to chip (without transceiver), then the chip should drive the IFC_TE signal such that IFC disables external transceiver.

  For booting from slower devices where transceiver is present in the connectivity, then the chip should drive the IFC_TE signal such that IFC enables external transceivers.

17.4.6.2 Transceiver enable post-boot

After booting, and during normal read/write transactions, the TE pin acts as an output pin.

The configuration of this pin is done on a per-bank basis using CSPRn[TE] bits:

- If a transaction hits bank n, the logic specified by CSPRn[TE] appears on the TE pin when CS*n* gets selected by a flash interface arbitration.
- If no CS*n* is selected, the default (board tied) value appears on the TE pin. (By default, the transceiver is disabled.)

This table shows how CSPR*n*[TE] should be programmed depending on polarity of the transceiver's enable pin and type of memory connected to that particular bank.

Table 17-7. Programming of CSPRn[TE]

External transceiver input	Value of CSPRn[TE]	
	Slow memory	Fast memory

Table continues on the next page...

Table 17-7. Programming of CSPRn[TE] (continued)

Active low	0	1
Active high	1	0

17.4.6.3 Transceiver enable example

In this scenario, three chip-selects are used.

The connections are as follows:

- CS0: Slow device
- CS1: Fast device
- CS2: Slow device

Because the transceiver used in this scenario has an active-low enable (TOE), the programming of the CSPRn[TE] should be:

- 0 when a slow device is connected to the corresponding bank (to enable the transceiver).
- 1 when a fast device is connected to the corresponding bank (to disable the transceiver).

Therefore, configuration would be as follows:

- CSPR0[TE] = 0
- CSPR1[TE] = 1
- CSPR2[TE] = 0

17.5 NAND flash control machine

The NAND flash control machine (FCM) provides an interface to parallel-bus NAND flash devices.

There are separate machines for asynchronous, NVDDR mode.

17.5.1 NAND Flash Connections

The following figure shows the connection of various type of NAND devices (Asynchronous, NVDDR) with IFC.

This diagram represents the IFC I/O pins map to the various NAND devices. Actual connections should be made keeping physical effects like loading etc. in mind.

Figure 17-10. IFC async NAND flash connections

Figure 17-11. IFC NAND Flash (NVDDR) Connections

17.5.2 NAND flash synchronous mode

Only 8-bit port size NAND flash EEPROMs are supported for source Synchronous mode in ONFI 2.2. The commands, address bytes, and data are transferred on AD[0:7].

For both writes and reads, the DQS is used to qualify data. IFC signals CLE and AVD/ALE determine whether the writes are of command type (only CLE asserted), address (only AVD/ALE asserted), or write data (both AVD/ALE and CLE asserted with read/write being high). Reads are indicated by asserting both AVD/ALE and CLE, with read/write being low.

17.5.3 NAND flash asynchronous mode

For connections between an 8-bit port size NAND flash EEPROM and the IFC in FCM mode, commands, address bytes, and data are all transferred on AD[7:0] with WE_B asserted for transfers written to the chip, or RE_B asserted for transfers read from the

chip. IFC signals CLE and AVD/ALE to determine whether writes are of the type command (only CLE asserted), address (only AVD/ALE asserted), or write data (neither CLE nor AVD/ALE asserted).

For connection to a 16-bit NAND flash EEPROM to IFC in FCM mode, the high byte of data appear on AD[15:8], whereas the commands, address bytes, and the low byte of 16-bit data is placed on AD[7:0] during read and write data transfers.

17.5.4 NV-DDR Mode

Only 8-bit port size NAND flash EEPROM is supported for NV-DDR.

The commands, address bytes and data are transferred on AD[7:0]. For both writes and reads, the DQS is used to qualify data. IFC signals CLE and ALE determine whether the writes are of command type (only CLE asserted), address (only ALE asserted), or write data (both ALE and CLE asserted with W/R_B being high). Reads are indicated by asserting both ALE and CLE, with W/R_B being low.

17.5.5 Write protect

NAND/NOR flash devices come with an input pin, write-protect (WP_B), used to protect the flash data from any write.

Register bit CSPR*n*[WP] per chip-select indicates whether or not write accesses are allowed on the particular chip-select.

If a NOR device is connected and a write transaction arrives to a write-protected chip-select, then do not assert chip-select on the NOR flash device and consume the transaction internally. Hence, the protection is controlled by software and the IFC hardware. A write protect error is also generated.

When NAND operation is triggered and IFC detects transition of IFC_CSPRn[WP] from its previous value then it inserts idle cycles on the interface for TWW time where CS_B and WP_B are asserted as shown in figure below.

As a NAND device is not directly memory-mapped like a NOR device and is accessed through the instruction register (FIR), hence there is no prior information of the type of operation to be performed on the NAND. Thus, assert chip-select even if the device is write-protected and expect the device to ignore the write data sent.

Figure 17-12. Write-protect timing in NAND asynchronous mode

For synchronous mode, when cleared to zero, the WP_B signal disables the flash array program and erase operations. This signal should only be transitioned while there are no commands executing on the device. After modifying the value of WP, the host should not issue a new command to the device for at least t_{WW} delay time. The transition of the WP_B signal is asynchronous and unrelated to any CLK transition in the source synchronous data interface. The following figure shows the write protect timing requirement for source synchronous mode. The figure above shows the timing details.

Figure 17-13. Write protect timing in NAND (synchronous mode)

17.5.6 SRAM buffer

Read and write accesses to IFC banks controlled by NAND FCM do not access attached NAND flash EEPROMs directly. Rather, these accesses read and write the buffer RAM (a single, shared 32-KB space internal to the IFC and mapped by the base address of every NAND FCM bank).

Even though each NAND FCM-controlled bank has a different base address to differentiate it, all accesses to such banks access the same buffer space. External IFC signals, such as /ALE and CS*n*, do not assert upon accesses to the buffer RAM.

- To perform a page-read operation from a NAND flash device, software initializes the FCM command, mode, and address registers, and triggers the read operation on a particular bank by setting the bit in NANDSEQ_STRT register. FCM executes the sequence of op-codes held in FIR, reading data from the flash device into the shared buffer RAM. While this read is taking place, software should not access buffer RAM. After loading the complete page on buffer RAM ECC decoding can be performed if it is enabled. Once all the errors have been fixed and if operation completion interrupt is enabled, an interrupt is generated. When FCM has completed its last command, buffer RAM can be read.
- To perform a page-write operation, the software writes data into the SRAM buffer. Then, the software initializes the FCM command, mode, and address registers and triggers the write operation on a particular BANK by setting the bit in NANDSEQ_STRT register. FCM will execute the sequence of op-codes held in FIR, writing the data from shared buffer RAM to the flash device. While this write is taking place, the software is free to write data in other buffers of the FCM buffer RAM.

By programming NCFGR[NUM_LOOP] field, the NAND FCM can be used for multi-page read and write operation. A maximum of 32 pages (for 512-byte page), 8 pages (2-KB page), and 4-page (4-KB page), 2 page (8Kbyte page) and 1 page (16Kbyte page) can be used for multi-page operation. Operation completion interrupt will be sent only after completing last page operation.

17.5.6.1 Guidelines for SRAM buffer for NAND access

Guidelines for SRAM buffer for NAND access are as follows:

- The IFC supports only one operation at a time on the NAND flash; that is, the user can either read data from the SRAM buffer for a NAND flash read or write data in the SRAM buffer for a NAND flash-write.
- NAND program: The user must fill the complete program data in the IFC's SRAM buffer before setting the trigger on the IFC to start the data transfer from the SRAM

buffer to the NAND flash. When the trigger is set to transfer the data to a NAND device, the SRAM buffer must not be accessed for either read or write. Accessing the SRAM buffer during this time may result in undesirable outcome. The user must wait for the current program operation to complete before accessing the SRAM buffer for the next operation.

- NAND read: When the trigger is set in the IFC, the read data coming from the NAND device is stored in an SRAM buffer. In this operation, the SRAM buffer should not be accessed until the event-completion flag is set.

17.5.6.2 Buffer layout and page mapping for 512-byte page NAND flash

The FCM buffer space is divided into 1-KB buffers for 512-byte-page devices (CSOR*n*[PGS] = 00), mapped as shown in the figure below.

The EEPROM's page numbered P is associated with buffer number (P mod 32), where P = ROW*n*. Because the bank size set by AMASK*n*[AM] is greater than 32 KB, an identical image of the FCM buffer RAM appears replicated every 32 KB throughout the bank address space.

In the case where NAND_BC[BC] = 0, FCM transfers an entire page, comprising the 512-byte main region followed by the 16-byte spare region; the 496-byte reserved region is not accessed and remains undefined for software. However, for commands given a specific byte-count in NAND_BC[BC], COL*n*[MS] locates the starting address in either the main region (MS = 0) or the spare region (MS = 1).

Figure 17-14. SRAM buffer layout for 512-byte page device

17.5.6.3 Buffer layout and page mapping for 2-KB page NAND flash

The FCM buffer space is divided into eight 4 KB buffers for 2-KB page devices (CSOR*n*[PGS] = 01).

Each page in a 2 KB-page NAND flash comprises 2112 bytes, where 2048 bytes appear as main-region data and 64 bytes as spare-region data. The EEPROM's page numbered P is associated with buffer number (P mod 8), where P = ROW*n*. Because the bank size set by AMASK*n*[AM] is greater than 16 KB, an identical image of the FCM buffer RAM appears replicated every 16 KB throughout the bank address space.

If NAND_BC[BC] = 0, the FCM transfers an entire page comprising the 2048-byte main region followed by the 64-byte spare region; the 1984-byte reserved region is not accessed, and remains undefined for software. However, for commands given a specific byte count in NAND_BC[BC], COL*n*[MS] locates the starting address in either the main

region (MS = 0) or the spare region (MS = 1). When different IFC banks control both page devices, a 4 KB-page buffer must be assigned to either the first four or last four 512-byte page buffers.

Figure 17-15. SRAM buffer layout for 2-KB page device

17.5.6.4 Buffer layout and page mapping for 4 KB page NAND flash

The FCM buffer space is divided into four 8 KB buffers for 4 KB-page devices (CSOR*n*[PGS] = 10).

Each page in a NAND flash comprises 4224 bytes, where 4096 bytes appear as main region data and 128 bytes as spare region data. The EEPROM's page numbered P is associated with buffer number (P mod 4), where P = ROW*n*. Because the bank size set by AMASK*n*[AM] is greater than 32 KB, an identical image of the FCM buffer RAM appears replicated every 32 KB throughout the bank address space.

If NAND_BC[BC] = 0, FCM transfers an entire page comprising the 4096-byte main region followed by the CSOR*n*[SPRZ]-byte spare region.

Figure 17-16. SRAM buffer layout for 4 KB-page device

17.5.6.5 Buffer layout and page mapping for 16 KB (large) page NAND flash

The FCM buffer space is divided into single 16Kbyte buffers for 8Kbyte large-page devices (CSORn[PGS] = 3'b100), mapped as shown in Figure 17-17.

Each page in a large-page NAND flash comprises 16384 + spare region bytes, where 16384 bytes appear as main region data, and spare region bytes appear as spare region data. Since the bank size set by AMASKn[AM] will be greater than 32 Kbytes, an identical image of the FCM buffer RAM appears replicated every 32 Kbytes throughout the bank address space.

In the case that NAND_BC[BC] = 0, FCM transfers an entire page, comprising the 16KB main region followed by the CSORn[SPRZ]-byte spare region.

Figure 17-17. SRAM buffer layout for 16KB Page size device

17.5.6.6 SRAM buffer initialization requirement

Set NCFGR[SRAM_INIT_EN] bit to initialize the SRAM before initiating the first program operation. See NAND Configuration register (NCFGR) for details.

17.5.7 Use of ECC algorithms

BCH encoder and decoder algorithms are implemented to detect and correct up to 4/8 bits in each 512-byte sector and 24/40 bits in each 1 KB sector.

A Galois Field 2^{13} is used for the encoding and decoding of 4- and 8-bit ECC and a Galois field 2^{14} is used for 24-/40-bit ECC.

17.5.7.1 Generating Galois field elements

The Galois field elements can be generated by shifting the LFSR.

The following primitive polynomials are used for the computation of respective field elements.

Galois Field 2^{13} field elements

As shown in this figure, the field elements can be generated by shifting the LFSR. Each state of LFSR represents the field element.

P(x) = 11011000000001 in binary form and $1+x+x^3+x^4+x^{13}$in polynomial form.

Galois Field 2^{14} field elements

Primitive polynomial P(x) = 110000000001101; that is, $1 + x + x^{11} + x^{12} + x^{14}$

17.5.7.2 BCH encoding

This section describes BCH encoding scheme.

This figure represents the encoder implementation.

- 4-bit ECC: g(x) = 1101010101100001110101011100001000001100010010100010 1

- 8-bit ECC:
  g(x)=1100010011011111001000111010001110000010111000011100100000110000
  1101111000000111001010001001111110101
- 24-bit ECC:
  g(x)=1010000011110100000110100100000100010001000001011000111111011001
  1111001111110100001000110000001111000100100010001001000100010100101010
  1101111010100000111100101101110111110110111011000110001111101101111 0
  1001100000011100100000011100001011010111101101100111110111001111101 1
  1011001101110001011100011010000000001101100000000011001000011010011 1
  1
- 40-bit ECC:
  g(x)=1110000011101110011001100111110110100101110011101000011000000000
  0001110011111000100111100000000010100011011101010001110110110101110 1
  0100011110010111111100001101011010100100000110011000111010001111010 1
  0011010000011001101101110111101010110100011111100001001110010000111 1
  0100101100101111101001110111011011011000010101101110001001001110001 0
  1111111011010011101101000000010101110101000101100111011010101011010 0
  0000100001100110000110010100101111111100011011101100100011101100011 1
  1111110011100000011111011010001011000001111101011001101001010011001 1
  01111

In each of the generator polynomial binary forms mentioned above, the msb represents the lowest polynomial power (g0) and the lsb represents highest polynomial power (gN). The following LFSR can be used for BCH encoding.

In encoding, operation data corresponding to a sector is given to the LFSR. During this time, the following events occur:

- Switch 1 is closed and switch 2 is in the down position (A) to allow the transfer of data to the output.
- After the transfer of a sector's data, switch 1 is opened and switch 2 is moved to the up position (B).
- Now N parity bits can be read out from the LFSR. ENCODER_LFSR[N-1] represents the first parity bit out and ENCODER_LFSR[0] the last parity bit.
  - The value of N is 52, 104, 336 and 560 for 4-, 8-, 24- and 40-bit ECC.

Parity bits are stored in spare region of the page at offset 0x08. BCH parity bits are first written to SRAM buffer and then to NAND flash device during program operation. User can read the ECC bytes by reading the appropriate locations in SRAM buffer for debug purpose.

CSORn[ECC_MODE] can be used to select 4, 8, 24, 40 bit correction mode and CSORn[ECC_ENC_EN], CSORn[ECC_DEC_EN] can be used to enable/disable the ECC logic.

Data bytes are supplied to encoder on incrementing address basis. Below figure explains the 512 byte data sector and parity arrangement for 4 bit ECC mode. Bit d0 is the first bit which is entered to encoder and parity bit p0 is first bit which comes out from encoder at the end of operation. In 4 bit mode zeros are padded at the end to align parity to 8 byte boundary.

Figure 17-18. 512 byte sector and parity arrangement in 4 bit mode

Similarly in 8 bit mode 104 parity bits will be genrated by encoder for each sector, they are padded with 24 zeros to align parity to 8 byte boundary and total parity size becomes 16 bytes.

In 24 and 40 bit mode sector size is 1KB and no zero padding is required at the end of generated parity.

There can be multiple sectors in a page depending on its size however data feeding mechanism and parity arrangement will remain same as ECC algorithm works on sector basis. Sector wise parity bytes are arranged in spare region starting from offset 0x08 as shown in figure below.

For example, If there is a memory of 4KB page size and ECC requirement is of 24 bit/1KB sector then 4KB page size memory will have 4 sectors of 1KB each. Encoder will generate 336 parity bits (42 bytes) for each sector, hence total of 42x4 = 168 ECC bytes will be generated. All 168 bytes will be placed in the spare region of the NAND flash starting at byte address offset 0x08 such as:

{Sector-1 parity bits {p0, p1,...p335}, Sector-2 parity bits {p0,p1,...p335}, Sector-3 parity bits {p0, p1,...p335}, Sector-4 parity {p0,P1,...p335} }

where p0 is the first parity bit coming out from encoder and p335 the last bit.

When IFC accesses the NAND Flash device for a Read operation, it performs a BCH decoding algorithm, and indicates how many bit errors were detected and corrected. The number of errors per sector gets registered in ECCSTAT registers.

During decoding full page will be loaded into buffer RAM from Flash and then only decoding will start. This is required because BCH decoder can work when a sector's data and corresponding ECC bytes are supplied to the decoder. During decoding data and parity bits are supplied in similar fashion as required during encoding.

Figure 17-19. ECC arrangement in a page

Below figure represents the logical flow of encoding and decoding operation during program and read. During program ECC bytes will also be written back in SRAM. During read first all data bytes corresponding to a page will be written in SRAM buffer and then the decoding will start on sector basis. At a time only one operation can be performed on NAND ie. it can be either read or programmed.

Figure 17-20. BCH encoding/decoding during flash program/read

NOTE

- The BCH Encoder and Decoder should be enabled only when performing full page operations. In the case of partial page operations, the Encoder and Decoder should be disabled.
- IFC computes the ECC only for the Main Region data (ECC is not computed for the Spare Region). Thus, the decoder will detect and correct 4/8 bits of error on the Main Region data and the corresponding ECC bytes stored.

This figure represents the logical flow of encoding and decoding operation during program and read. During program, ECC bytes are also written back in SRAM. During read, first, all data bytes corresponding to a page are written in SRAM buffer, and then the decoding starts on sector basis. At any time, only one operation can be performed on the NAND; that is, it can only be either read or programmed.

Figure 17-21. BCH encoding/decoding during flash program/read

17.5.8 Programming the NAND FCM

The NAND FCM performs operations on the NAND flash interface based on the values programmed in the common and NAND register spaces.

The user is expected to program the following registers with the appropriate values before triggering the NAND FCM operation:

- Chip-select option registers
- Flash timing registers
- General control register
- NAND configuration register
- NAND flash command registers
- Flash row and column address registers
- NAND flash byte count register
- NAND flash instruction registers
- NAND chip-select register
- NAND event and error interrupt enable register (in case an interrupt needs to be generated)
- NAND control register

When all the above registers have been programmed, the NAND FCM can then be triggered through the NAND operation sequence start register.

Figure 17-22. FCM instruction sequence mechanism

NOTE

User must use the WFR opcode at the end of NAND FIR programming to probe the RDY_B/BSY_B signal before end of operation. Status poll is also recommended after every write or erase command.

17.5.8.1 FCM command instructions

There are different types of command instructions.

- Commands that issue immediately (CMD0-CMD7)
  - Asynchronous mode: These commands write a single command byte by asserting CLE and WE_B while driving an 8-bit command on the AD[7:0] (8-/16-bit port) bus. The opcode CMD*n* sources its command byte from the field FCR[CMD*n*]. Therefore, up to eight different commands can be issued in any instruction sequence.
  - NV-DDR Mode: These commands write a single command byte by asserting CLE with respect to IFC_DDR_CLK while driving an 8-bit command on the AD[7:0] bus. The opcode CMDn sources its command byte from the field FCR[CMDn]. Therefore, up to 8 different commands can be issued in any instruction sequence.
- Commands that wait for RB_B to be sampled high before issuing (CW0-CW7)

- Asynchronous mode: These commands first poll the RB_B signal to be sampled high before writing a single command write on the AD[7:0] (8-/16-bit port) bus sourced from FCR[CMDn] for opcode CW*n*. It is necessary to use CW*n* opcodes whenever the memory is expected to be in the busy state (such as following a page read, block erase or program operation) and therefore, initially unresponsive to commands.
- NV-DDR Mode: These commands first poll the R/B signal to be sampled high before writing a single command write on the AD[7:0] (8 bit port) bus sourced from FCR[CMDn] for opcode CWn. It is necessary to use CWn opcodes whenever the memory is expected to be in the busy state (such as following a page read, block erase or program operation) and therefore, initially unresponsive to commands.

Consult the manufacturer's datasheet to determine the values to be programmed into the FCR register and whether a given command in the sequence is expected to initiate device busy behavior.

17.5.8.2 FCM address instructions

Address instructions are used to issue addresses to the NAND flash device.

- Asynchronous mode: A complete address is formed by a sequence of 1 or more bytes, each written onto AD[7:0] (8-bit/16-bit port) with AVD/ALE and WE_B asserted together.
- NV-DDR Mode: A complete address is formed via a sequence of one or more bytes, each written onto AD[7:0] (8-bit port) with AVD/ALE asserted with respect to IFC_DDR_CLK.

17.5.8.3 FCM data read instructions

Read data instructions assert RE_B repeatedly to transfer one or more bytes of read data from the NAND flash device.

The different types of data read instructions provided are as follows:

- Read BC bytes of data from the NAND flash device into the internal SRAM
  - Asynchronous mode: This instruction reads NAND_BC[BC] into the current FCM SRAM buffer addressed. The data driven by the NAND flash device is sampled on the basis of the TRAD timing parameter and the RE_B is asserted for a time period equivalent to the TRP timing parameter. If NAND_BC[BC] =

0, an entire page (including the main and the spare regions) is transferred in a burst.
    - NVDDR mode: This instruction reads NAND_BC[BC] into the current FCM SRAM Buffer addressed. The data driven by the NAND flash device is sampled on the basis of the quarter cycle shifted DQS . If NAND_BC[BC] = 0, an entire page (including the main and the spare regions) is transferred in a burst.
- Read BC bytes of data from the NAND flash device into the internal SRAM during boot
    - Asynchronous mode: This instruction reads NAND_BC[BC] into the current FCM SRAM buffer addressed. The data driven by the NAND flash device is sampled on the basis of the TRAD timing parameter and the RE_B is deasserted once the read data has been sampled by the NAND FCM. If NAND_BC[BC] = 0, an entire page (including the main and the spare regions) is transferred in a burst. This instruction can also be used for non-boot applications wherein the RE_B signal needs to be asserted until the read data has been sampled.
    - NV-DDR Mode: Not valid
- Read the NAND flash device status
    - Asynchronous mode: This instruction performs a status read operation on the NAND flash device. The status returned by the device is stored in the NAND_FSR register.
    - NVDDR mode: This instruction performs a status read operation on the NAND flash device. The status returned by the device is stored in the NAND_FSR register. Two bytes (duplicate) of data will be stored in the NAND_FSR register.
- Read data into the MDR register
    - Asynchronous mode: This instruction reads 8/16 bits of data from a given page in the NAND flash device (based upon the column address programmed) into the MDR register.
    - NVDDR mode: This instruction reads 16 bits of data from a given page in the NAND flash device (based upon the column address programmed) into the MDR register.

17.5.8.4 FCM data write instructions

Write data instructions assert WE repeatedly to transfer program data to the NAND flash device.

- Asynchronous mode: Write data instructions assert WE repeatedly to transfer 1 or more bytes of program data to the NAND flash device. If NAND_BC[BC] = 0, an entire page (including the main and the spare regions) is transferred in a burst.
- NV-DDR Mode: Write data instructions assert CLE and ALE together to transfer 2 or more bytes of program data to the NAND Flash Device. The data is driven along

with centre aligned DQS for the NAND to sample the data. If NAND_BC[BC] = 0, an entire page (including the main and the spare regions) is transferred in a burst.

17.5.9 Looping FIR sequences

The loop feature provides the flexibility to loop certain FIR sequences (such as program, read, and erase) a finite number of times based on the value programmed in NCFGR[NUM_LOOP].

The row and column addresses sent to the NAND flash device during a loop operation depend on NCFGR[ADDR_MODE]. For a more detailed description of the addresses performed on the NAND flash device for every subsequent execution of the FIR sequence during a loop operation, see NAND Configuration register (NCFGR).

NOTE

For this feature to correctly execute, it is important to note that the opcodes programmed in the FIR sequence corresponding to the column and row addresses should always be CA0 and RA0. Therefore, it is recommended that only one type of operation (such program/read/erase) be performed through a loop sequence.

17.5.10 NAND DLL

The DLL is a dynamically-adaptive clock delay module. It provides the ability to programmably select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT).

This module is functional when NVDDR device is connected to IFC.

- The DLL is used to calculate the half pulse width of an internally generated reference clock (clk_ref) and has the same period as IFC_ND_DDR_CLK. This calculated half pulse width is used to calculate a delay for the incoming raw DQS (clk_in), which is then used to capture the read data

The DLL is comprised of three major components:

- One instance of the delay chain - used for reference calibration (half-phase detect)
- One instance of the delay - used as the slave delay line (performs the programmed delay)
- Phase detector module - forms the control loop with the reference delay line and provides the decoding and controls for the slave delay line

The following figure shows the block diagram of the DLL used in the IFC.

Figure 17-23. DLL block diagram

The DLL control loop consists of a counter, reference delay line, and phase detector which operate on the ref_clock reference clock input. The reference clock (clk_ref) is fed into the reference delay line. After reset, a single delay tap is selected. A phase detector is used to detect the condition where a half shift has a occurred. In addition to this, signals are generated to either increment or decrement the counter which controls the delay line (if the half-phase detect condition is not met). Any changes in the delay of the individual elements of the delay chain (due to PVT) will automatically cause the phase detector logic to determine if a change in the counter value is required. Once the half-phase shift is detected, an internal lock signal is generated.

Refer to DLL configuration guideline (valid when IFC is in NVDDR Mode) for guidelines on DLL usage.

17.5.11 NAND asynchronous mode timings

This section describes the basic NAND flash timing waveforms and the programmable timing parameters.

Figure 17-24 explains the NAND program cycle; Figure 17-25 and Figure 17-26 explain the basic read cycle waveforms and corresponding timing parameters for NON-EDO and EDO mode flash. Non-EDO mode corresponds to ONFi2.0 async mode 0/1/2/3 while EDO mode corresponds to ONFi mode 4/5.

17.5.11.1 NAND asynchronous mode program data timing

This figure shows the NAND flash program.

Figure 17-24. NAND flash program

17.5.11.2 NAND asynchronous mode read data timing

This figure shows the NAND flash read.

Figure 17-25. NAND flash read

17.5.11.3 NAND asynchronous mode calculating read data window width

Calculating the read data window width can be done in several ways.

- Non-EDO mode NAND
  - The window size for the received data is calculated as t_{RP} - t_{REA} + t_{RHOH}.
- EDO mode NAND
  - The window size for the received data (except the last data beat) is calculated as t_{REH} - (t_{REA} - t_{RP}) + t_{RLOH}.
  - The window size for last data beat received is calculated as t_{RHOH} - (t_{REA} - t_{RP}).

Based on these calculations, the minimum size of the data window width is 9 ns (for EDO ONFi mode 5). To support this data window width, a minimum of 333 MHz for the IFC module input clock is required. By increasing the value of timing parameter t_{RP}, that is, the RE pulse width, the data window size can be increased.

NOTE

The read cycle time should satisfy this condition: ((t_{RP} + t_{REH})≥ t_{RCmin}). If the sum of the t_{RPmin} and the t_{REHmin} values specified by the datasheet is less than the t_{RCmin} (from the datasheet), it is

then recommended that the t_{RPmin} value be increased to meet the t_{RCmin} timing requirement. This is recommended because increasing the t_{RPmin} value causes the data window size to increase.

Read data sampling time (TRAD) represents the read data sampling time on the NAND. The value should be programmed where the sampling at the IFC should be done at the center of the received data eye. Its value can be calculated as [T_{REAmax} (from NAND datasheet) + 2 x board delay + 1/2 received data window size]. If in EDO mode, the user should choose the TRAD value corresponding to the smaller data window calculated using the two formulae explained above for EDO.

NOTE

For EDO mode, the value of TRAD should always be less than t_{RP} + t_{REH}. If the board delay value used in the TRAD calculation results in this condition to not be met, the user may not be able to run the interface at the EDO mode frequency. In this situation, the read timing should be relaxed by increasing t_{RP} value to access the NAND device at a slower speed.

17.5.11.4 NAND asynchronous mode read status timing

This figure shows NAND read status timing.

Figure 17-26. NAND status read

17.5.12 NV-DDR mode timings

17.5.12.1 NAND synchronous mode program operation

In a program operation, the data is expected by the flash memory on both the rising and falling edges of the DQS. The DQS should be center-aligned with respect to the data.

The following figure shows the program operation and the relevant timing parameters for a IFC module input clock:external DDR clock (IFC_DDR_CLK) ratio of 1:2.

Figure 17-27. NAND synchronous mode program operation

17.5.12.2 NAND synchronous mode program with clock stopped

The user can program the IFC to save power during the data input cycles by setting the DDR clock stop bit in the IFC DDR clock control register.

This will result in the IFC_DDR_CLK signal being held high (that is, stopping the CLK).

However, the IFC_DDR_CLK will be stopped only during program phase. The following figure shows the timing relationship of a program operation with the clock stopped. The values of the AVD/ALE, CLE, and W/R_n signals are latched on the rising edge of IFC_DDR_CLK and thus, while the clock is held high, these signals are don‘t care.

The following figure shows the program operation with the clocked stopped and the relevant timing parameters for an IFC module input clock:external DDR clock (IFC_DDR_CLK) ratio of 1:2.

Figure 17-28. NAND synchronous mode program operation with clock stopped

17.5.12.3 NAND synchronous page read operation

For page read operation, the flash memory sends the read data aligned with the rising and falling edge of the DQS. In this case, the DQS is edge-aligned with the data.

The following figure shows the read page operation and the relevant timing parameters for an IFC module input clock:external DDR clock (IFC_DDR_CLK) ratio of 1:2.

Figure 17-29. NAND synchronous mode page read operation

17.6 NOR flash control machine

The NOR FCM allows an interface to muxed (parallel) NOR as well as unmuxed (ADM) NOR. Unmuxed NORs are devices which have dedicated address and data pins whereas muxed NOR devices have Address/Data Multiplexed pins and a few dedicated address pins at its interface. The CSORn[NOR_MODE] field is used to distinguish these two types of devices. Both of these types of devices may work in async as well as sync read mode, which can be set using the CSORn[SYNC_RD_EN] register field.

NOTE

Write is always asynchronous.

17.6.1 NOR boot

For NOR boot, CS0 is the boot chip-select.

Before fetching the first instruction from the NOR flash, the IFC flash timing registers are loaded with default timing parameters.

17.6.2 Unmuxed (parallel) asynchronous NOR read timings

The figures show the read cycle and burst read cycle timing diagrams.

Figure 17-30. Read cycle timing

Figure 17-31. Burst read cycle timing

17.6.3 Simple asynchronous NOR write timings

The figures show the write cycle and burst write cycle timing diagrams.

Figure 17-32. Write cycle timing

Figure 17-33. Burst write cycle timing

17.6.4 Muxed (ADM) asynchronous NOR read timings

This figure shows the read cycle timing diagram.

Figure 17-34. Read cycle timing

17.6.5 Muxed (ADM) asynchronous NOR write timings

This figure shows the write cycle timing diagram.

Figure 17-35. Write cycle timing

17.7 Unmuxed (Parallel) Synchronous NOR Read Timings

17.7.1 Continuous Mode Sync Burst Read- Unmuxed NOR

This section describes timings for Unmuxed NOR-Continuous Burst Mode Sync Read.

Figure 17-36. Continuous Mode Sync Burst Read

Figure 17-37. Unmuxed NOR-Continuous Burst Mode Sync Read Timing

17.7.2 Unmuxed (parallel) asynchronous NOR read timings

The figures show the read cycle and burst read cycle timing diagrams.

Figure 17-38. Read cycle timing

Figure 17-39. Burst read cycle timing

17.8 Muxed (ADM) Synchronous NOR Read Timings

17.8.1 Continuous Mode Sync Burst Read- ADM NOR

This section describes timings for Muxed NOR-Continuous Burst Mode Sync Read.

Figure 17-40. Muxed NOR-Continuous Burst Mode Sync Read Timing1

Figure 17-41. Muxed NOR-Continuous Burst Mode Sync Read Timing2

17.8.2 Split Mode Sync Burst Mode- ADM NOR

Following figures show Muxed NOR-Split mode sync burst read timings.

Figure 17-42. Muxed NOR-Split Mode Sync Burst Read Timing1

Figure 17-43. Muxed NOR-Split Mode Sync Burst Read Timing2

Figure 17-44. Muxed NOR-Split Mode Sync Burst Read Timing3

17.9 General purpose chip-select machine (GPCM)

The GPCM controller has two modes of operation: normal GPCM and generic ASIC.

Only one mode is operational at a time per chip select. The mode is selected by the value programmed in GPMODE field of CSOR*n*_GPCM register.

17.9.1 Normal GPCM mode of operation

The figure below explains the interface of the IFC to a device connected through the normal GPCM FCM.

Figure 17-45. Normal GPCM interface

17.9.1.1 Normal GPCM program operation

In normal GPCM, the write transaction depends on the register field CSORn[WGETA].

The following options, through an internal counter or through an external device giving an access termination signal, are explained below.

17.9.1.1.1 Normal GPCM internal counter-based program operation

Transactions are based on control-signal timings that are programmed in the flash timing registers for GPCM mode (IFC_FTIM*x*_CS*n*_GPCM) when CSOR*n*[WGETA]=0.

Assertion of AVD/ALE is aligned to the rising edge of ifc_launch_clk (an internal clock), after that all the timing parameters are in terms of the IFC module input clock (ip_clk). IFC_CLK is external clock which goes to the device and it is generated by inverting ifc_launch_clk.

Figure 17-46. Normal GPCM program operation - non-burst mode

Figure 17-47. Normal GPCM back-to-back program operation

In burst mode, the next data and next address are sent after one TWP period (the amount of time between address/data cycles). If IFCTA_B is deasserted and the burst is controlled by the timing parameter programmed in FTIM registers, the burst will

continue until either the burst length is reached or complete data is transferred. If the number of bytes of data to be transferred is less than programmed burst length multiplied by port size, then the burst will be terminated before the programmed maximum number of burst transfers.

Figure 17-48. Normal GPCM program operation - burst mode

For a normal timing-based transaction, CSOR*n*[RGETA]=0. If the access is aborted by IFCTA_B and there is remaining data to be transferred, the current burst is terminated and a new burst transaction is launched. Ongoing burst terminates at the next rising edge of ifc_launch_clk after the assertion of IFCTA_B. The next burst's starting address is the last burst address + the number of bytes that needed to be transferred in the last burst.

$Address_{n+1} = Address_n$ + (number of bytes for current burst)

The next burst starts only after deassertion of IFCTA_B. This is registered as an abort error in the status register and all transaction attributes are locked.

Since IFCTA_B is synchronized through asynchronous FIFO, bus termination occurs only after a synchronization delay of IFCTA_B. IFCTA_B should be asserted for at least one IFC_CLK cycle if it is synchronous (that is, meeting setup and hold time); otherwise, asynchronous IFCTA_B should be asserted for a minimum of two IFC_CLK cycles.

Figure 17-49. Normal GPCM program operation - transaction aborted

17.9.1.1.2 Normal GPCM external termination-based program operation

When WGETA is set, the next address and data are sent only after sampling the rising edge of IFCTA_B or after timeout (as defined in register CSORn[GPTO]).

IFCTA_B is synchronized to the IFC module input clock through asynchronous FIFO and only after the rising edge of IFCTA_B, the new beat is sent over the interface. The next data and address beat is sent in sync with the positive edge of ifc_launch_clk. The last beat of the last burst completes at the next rising edge of ifc_launch_clk after assertion of IFCTA_B, without waiting for deassertion of IFCTA_B and CS_B is deasserted.

If transaction size is greater than GPCM port size, the transaction will be split into multiple port size accesses. For non-burst mode, the last split is terminated at the rising edge of ifc_launch_clk after assertion of IFCTA_B or after timeout. For other splits, IFC waits for IFC_TA deassertion and not available for further transactions until IFC_TA is deasserted.

As IFCTA_B is synchronized through asynchronous FIFO, bus termination occurs only after the synchronization delay of the IFCTA_B signal. IFCTA_B should be asserted for at least one IFC_CLK cycle if it is synchronous (that is, meeting setup and hold time); otherwise, if it is asynchronous, IFCTA_B should be asserted for a minimum of two IFC_CLK cycles.

Figure 17-50. Normal GPCM program operation - acknowledgment mode

17.9.1.2 Normal GPCM read operation

In normal GPCM, the read operation depends on the register field CSOR*n*[RGETA]. The following options, through an internal counter or through an external device giving an access termination signal, are explained below.

17.9.1.2.1 Normal GPCM internal counter-based read operation

In this approach, read data from memory is sampled based on the programming of FTIM1_CSn_GPCM[TRAD] and FTIM3_CSn_GPCM[TAAD].

In non-burst mode, TRAD defines the pulse width of OE_B. The read data is sampled on rising edge of IFC_CLK just before the de-assertion of OE_B

By default, (CCR[INV_CLK_EN]=1 and CCR[CLK_DLY]=0) inverted clock of ifc_launch_clk goes out (IFC_CLK) to the external device. Because the address is launched at the rising edge of ifc_launch_clk, the setup time on an external device for the address phase is half the cycle period of IFC_CLK. The read data launched by the device will be sampled by the IFC at the next rising edge of IFC_CLK. The programming of TRAD_GPCM and TAAD_GPCM should be as follows:

- For non-burst mode TRAD value should be programmed as {(2 + n)* (CCR[CLK_DIV] + 1)}, where n defines the memory access time in terms of IFC_CLK. Memory takes n ifc_clk to output data after output enable is asserted or after new address is sampled. n can take values 0,1,2,3...
- For burst mode TRAD/TAAD programmed will be {(2 + n)* (CCR[CLK_DIV] + 1)}, where n defines the memory access time in terms of IFC_CLK. Memory takes n IFC_CLK to output data after output enable is asserted or after new address is sampled. n can take values 0,1,2,3...

The following figures explain the above-mentioned read data sampling schemes:

Figure 17-51. Normal GPCM read operation - non-burst mode

TRHZ: Extended hold time for slow memories to disable their bus drivers
TBCTL: Bus turn-around time

Figure 17-52. Normal GPCM back-to-back read operation

When RGETA is programmed to 0 and access is aborted by ifc_ta, then the current burst is terminated and new burst transaction is launched. The ongoing burst is terminated at the next rising edge of ifc_launch_clk after assertion of ifc_ta. The next burst starting address will be the last burst address + number of bytes that needed to be received in the last burst.

$Address_{n+1} = Address_n$ + (number bytes for current burst)

The next burst starts only after deassertion of ifc_ta.

An abort error is registered in the status registers and all the trasaction attributes are locked. Since ifc_ta is synchronized through asynchronous FIFO, bus termination occurs only after synchronization delay of ifc_ta signal. ifc_ta should be asserted for minimum two IFC_CLK cycles.

Figure 17-53. Normal GPCM read operation - burst mode

Figure 17-54. Normal GPCM read operation - transaction aborted

17.9.1.2.2 Normal GPCM external termination-based read operation

When RGETA is set to 1, each beat of the transaction waits for an external termination by IFCTA_B or timeout.

For address transfer, the next address is sent only after IFCTA_B is deasserted or after timeout. IFCTA_B is synchronized to ip_clk through the asynchronous FIFO and only after the rising edge new address beat is sent over the interface. The new address beat is sent in sync with ifc_launch_clk. The last transaction cycle does not wait for deassertion of IFCTA_B; it ends at the next rising edge of ifc_launch_clk after assertion of IFCTA_B or after timeout.

For read data sampling, the read data is sampled on the falling edge of IFCTA_B. IFCTA_B and data both are synchronized through the asynchronous FIFO.

For non-burst mode, access is terminated after IFCTA_B is asserted or timeout.

In case of a timeout, an error is registered in the GPCM status register and all transaction attributes are locked. As IFCTA_B is synchronized through the asynchronous FIFO, bus termination occurs only after a synchronization delay of the IFCTA_B signal. IFCTA_B should be asserted for at least one IFC_CLK cycle if it is synchronous (that is, meeting setup and hold time); otherwise, if asynchronous, IFCTA_B should be asserted for a minimum of two IFC_CLK cycles.

Figure 17-55. Normal GPCM read operation - acknowledgment mode

17.9.2 Generic ASIC mode of operation

This figure explains the interface of the IFC to a device connected through generic ASIC mode of GPCM (CSORn[GPMODE]=1).

Figure 17-56. Generic ASIC interface

17.9.2.1 General ASIC program operation

Before initiating any general ASIC (GASIC) operation, the chip-select (CS_B) is asserted to select the appropriate device.

The assertion of start of frame (SOF_B) along with the low value on RW_B indicates the start of address phase for write transfer. In the case of a 16-bit device, the upper 16 bits of address is driven on the AD[15:0] first, followed by the lower 16 bits on the next clock. After the address phase, the write data sequences are driven on AD lines based on the port size. Along with the address and write data phases, per-byte parity is driven on the parity lines. On the completion of write data phase, the host controller waits for the ready status (RDY_B) and the parity error sampling (based on CSORn[GAPERRD] time) before deasserting the chip-select to complete the transfer. In the case where the device is not asserting RDY_B before timeout (as defined in CSORn[GPTO]) occurs, the host terminates the transfer by deasserting the CS_B.

Figure 17-57. GASIC Program Operation with 16-bit device and zero-wait state

Figure 17-58. GASIC program operation with 32-bit device and 4-wait state

For program operation, wait state represents delay (in terms of IFC_CLK) between first write data beat versus RDY_B assertion time.

For read, wait state represents delay (in terms of IFC_CLK) between last address beat sent by controller and RDY_L assertion time. If RDY_L is asserted in next cycles of address sampled by GASIC device then it is termed as zero wait state.

17.9.2.2 Generic ASIC read data sampling

After selecting the general ASIC (GASIC) device, the host asserts start of frame (SOF_B) and drive high value on RW_B to indicate the start of address phase for read transfer.

The host drives the AD lines and the corresponding PAR parity lines in this phase to send the 16 -bit of address to the device. After completing the address phase, the host stops driving the AD lines and waits for the assertion of ready status (RDY_B) from the device. The assertion of RDY_B indicates the start of read data phase. In the case the device has not asserted RDY_B before timeout occurs, the host terminates the transfer by deasserting the CS_B.

Figure 17-59. GASIC Read Operation with 16-bit device and 2-wait states

Figure 17-60. GASIC Read Operation with 32-bit device and zero-wait state

17.10 Clock generation module

The clock generation module generates a divided clock derived from the IFC module input clock depending on the programmed value in the CCR register.

Whenever there is a change in any clock division ratio, the clock is gated for few IFC module input clocks. The clock status is reported as unstable in the CSR register.

Figure 17-61. External clock generation

17.11 Initialization/Application information

The IFC can be used with separate address and data buses or with a multiplexed address/ data bus.

This section provides guidelines for interfacing peripherals in the IFC's various modes.

17.11.1 Unmuxed (Parallel) NOR with no loop back

External latch is used to latch address msb's coming through AD bus during ALE assertion and ADM_SHFT_MODE=0. Latch output and exposed ADDR pins are connected to NOR device's address pins. If device is capable of supporting burst read, then its AVD_B port will also be connected.

Figure 17-62. Connection of Parallel Sync NOR with IFC_CLK_SYNC_OUT pin with no looping back

17.11.2 ADM NOR connection

This section describes connection of ADM Sync NOR.

Figure 17-63. Connection of ADM Sync NOR

17.11.3 Switching Interfaces in ONFi NAND

17.11.3.1 Activating the NVDDR Interface

After coming out of reset and performing device initialization, by default the asynchronous interface of the NAND device is activated. To switch to the NV-DDR interface, the following steps should be performed:

- Wait for OPC, if any previous operation is being executed on the NAND device.
- Program a FIR sequence (CMD0 - UA - WBCD - WFR - NOOP) that will issue the SET FEATURE command, followed by Wait For Ready, followed by NOOP.
- Issue the SET FEATURES (EFh) command. (CMD0)
- Write address 01h, which selects the timing mode. (UA)
- Write P1 with 1Xh for NVDDR devices, where "X" is the timing mode used in the synchronous interface (WBCD)
- Write P2-P4 as 00h-00h-00h. (WBCD)
- tWB time after the last data of the SET FEATURE command has been written the device goes into busy state (R/B_B is pulled low). After tITC time elapses, the device enters ready state (R/B_B transitions to high) and the NAND will now be in desired mode. The user is required to poll for the OPC status bit before issuing any new operation on the NAND device.

- Program the CSORn[NAND_MODE] field to "0x01" for NV-DDR mode .
- If the user wants to use auto timing parameters, write the desired timing frequency value with which the NAND device is configured, to the CSOR_EXTn[MODE_FREQ] field. This will configure the FTIM registers, corresponding to the chip select whose MODE_FREQ field has been written to, for the source synchronous device. The values programmed in the FTIM registers will be calculated based on the timing mode value programmed in CSOR_EXTn[MODE_FREQ], the clock divider value in the DDR-CCR register and the maximum permissible operating frequency (as per the ONFI) for the programmed timing mode. Also the CSOR_EXTn[AUTO_TIM_PARAMS_SEL] should be set.
- If the user wants to use the FTIM registers to program timing parameters, the AUTO_TIM_PARAMS_SEL bit should not be set.
- Before starting read operation using NVDDRmode, user must wait for DLL to get lock as specified in DLL configuration guideline (valid when IFC is in NVDDR Mode).

17.11.3.2 Switching to the asynchronous interface

To activate the asynchronous NAND interface, once the NVDDR interface is active, program the NAND_CSORn[NAND_MODE] to set the asynchronous mode.

The user should ensure that there is no activity on the NAND interface while programming this register field. Then perform the following steps:

- Wait for OPC, if any previous operation is being executed on the NAND device.
- Program the CSORn[NAND_MODE] field to 00hh.
- Program FIR sequence (CMD0 -WFR - NOOP) that will issue the reset command (FFh) to the NAND device using an asynchronous command cycle. t_{WB} time after the command is issued, the RB_B signal goes low indication the device is in busy state. After t_{RST} time, the device will be in ready state. The user should poll for the OPC status bit before issuing any new operations on the NAND device.

Once the above FIR sequence is complete, the NAND device will be in asynchronous mode and will be set to timing mode 0.

17.11.3.3 Switching timing modes when configured in NVDDR mode

To switch timing modes on the device (with NVDDR interface active) the following steps should be performed:

- Wait for OPC, if any previous operation is being executed on the NAND device.
- Program a FIR sequence (CMD0 - UA - WBCD - WFR - NOOP) that will issue the SET FEATURE command, followed by Wait For Ready, followed by NOOP.
  - Issue the SET FEATURES (EFh) command. (CMD0)
  - Write address 01h, which selects the timing mode. (UA)
  - Write P1 with 1Xh for NVDDR devices, where "X" is the timing mode used in the synchronous interface (WBCD)
  - Write P2-P4 as 00h-00h-00h. (WBCD)
- t_{WB} time after the last data of the SET FEATURE command has been written the device goes into busy state (R/B_B is pulled low). After tITC time elapses, the device enters ready state (R/B_B transitions to high) and the NAND will now be in desired mode. The user is required to poll for the OPC status bit before issuing any new operation on the NAND device.
- If the user wants to use auto timing parameters, write the desired timing frequency value with which the NAND device is configured, to the CSOR_EXTn[MODE_FREQ] field. This will configure the FTIM registers, corresponding to the chip select whose MODE_FREQ field has been written to, for the source synchronous device. The values used for interface signal timing generation will be calculated based on the timing mode value programmed in CSOR_EXTn[MODE_FREQ], the clock divider value in the DDR-CCR register and the maximum permissible operating frequency (as per the ONFI spec) for the programmed timing mode. Also the CSOR_EXTn[AUTO_TIM_PARAMS_SEL] should be set
- If the user wants to use the FTIM registers to program timing parameters, the AUTO_TIM_PARAMS_SEL bit should not be set.

17.11.4 ONFI mode timing parameters

For source-synchronous devices, there is an option to program the IFC with timing parameters as per the ONFI 2.2 timing parameters for all timing modes.

CSOR_EXT*n*[ONFI_MODE] is used to indicate the timing mode with which the chip is programmed and the CSOR_EXT*n*[ONFI_TIM_PARAMS] is set to enable this feature. Based on the value of CSOR_EXT*n*[ONFI_MODE], the value of DDR_CCR[DDR_CLK_DIV], and the maximum-permissible operating frequency for a particular timing mode (as per ONFI 2.2), the timing values are programmed in the number of IPG clock cycles.

The values of the timing parameters (rounded to the next higher integer) can be calculated as follows:

- ONFI_TIMING_PARAMETER + (MODE_PERIOD/(2 x DDR_CLK_DIV + 2) - 1)
- MODE_PERIOD/(2 x DDR_CLK_DIV + 2)

where

- ONFI_TIMING_PARAMETER is an ONFI timing parameter (such as tcad, tcs, and so on)
- MODE_PERIOD is the minimum clock period (ns) for a particular ONFI mode.
- DDR_CLK_DIV is the clock division ratio programmed in the DDR_CCR[DDR_CLK_DIV] register.

17.11.5 NAND Volume Addressing Initialization sequence

In higher density capacity implementations there may be a significant number of CS pins required for a controller to support where there are many NAND packages with two to four CS pins per package. The CS pin reduction mechanism enables a single CS pin from the controller to be shared by multiple NAND targets, thus enabling a significant reduction in the number of CS pins required by the SOC. The CS pin reduction mechanism may be utilized with any data interface (asynchronous, NV-DDR, NVDDR2).

In the CS pin reduction mechanism, each NAND package is appointed a volume address during the initialization sequence. After initialization is complete, the controller may address a particular volume (NAND target) by using the VOLUME SELECT (E1h) command.

The following steps should be used to setup volume addressing assignment for ONFi 3.1 NAND devices/packages. These steps are defined for a case where ONFi 3.1 NAND devices/packages are connected on CS0 and CS1. The same steps can be extended for a scenarion where all the Chip Selects have ONFi 3.1 NAND devices/packages connected. The initialization sequence when utilizing the chip select reduction functionality is as follows:

1. Issue a reset command to reset all targets of the NAND package connected to CS0
2. Wait for R/B# to become 1 indicating that the reset command has been completed.
3. Issue SET FEATURES (EFh) command to the volume configuration feature address to appoint the volume address for the NAND target. The volume address specified should be unique amongst all NAND targets.
4. Wait for R/B# to become 1 indicating that the SET FEATURE command has been completed. After the SET FEATURE command completes, ENo for that target is set to one and the volume is deselected until a VOLUME SELECT (E1h) command is issued that selects the volume.
5. Insert a few wait cycles using NWAIT opcode (around 10 system clock cycles)

6. Poll the NAND_VOL_ADDR_STAT[VOL_ADDR_STAT_CS0] bit after the OPC bit is set. If the value is 0 it indicates that there are more targets within the NAND package connected to CS0 that need to be assigned a volume address.
7. Insert wait cycles equal to (tENI + tENO) using the NWAIT opcode.
8. Repeat steps 3 through 6.
9. Poll the NAND_VOL_ADDR_STAT[VOL_ADDR_STAT_CS0] bitafter OPC bit is set. If the value is 1 it indicates that there are no more targets within the NAND package connected to CS0 that need to be assigned a volume address.
10. Repeat steps 1 through 9 for CS1.
11. To complete the initialization process, a VOLUME SELECT (E1h) command is issued to select the next volume that is going to execute a command.

17.11.6 DLL configuration guideline (valid when IFC is in NVDDR Mode)

The DLL (NAND_DLL_LOW is) configured using the Configuration 0 and Configuration 1 registers. The following sequence should be followed before the DLL can be used to delay the incoming data strobe:

- The NAND_DLL_LOW is enabled when the DDR clock is enabled (by setting DDR CCR LOW [2]) and CSOR[NAND_MODE] = NVDDR mode.
- The NAND_DLL_LOW is used for interface frequencies up to 133 MHz.
- The reset value of the dll_slv_dly_target field in DLL Configuration 0 register for NAND_DLL_LOW is 0x7. This will ensure a delay of REF_CLK/4 on the incoming DQS.
- The user must change the NAND_DLL_CFG1[DLL_PD_PULSE_STRETCH_SEL] bit from its default value 1’b1 to 1’b0 for reliable pulse detection and DLL lock.
- The user has to poll for the value of ‘1’ on the DLL_STS_SLV_LOCK bit of NAND_DLL_LOW_STAT (if enabled) register before issuing any access to the NAND Flash. Neglecting this would result in incorrect read data capture by IFC.
- If the CSOR_EXT[MODE_FREQ] bit is changed, then the delay chain needs to be reset to make it lock again.

17.11.7 IFC flash connections

17.11.8 Bus turnaround

Because the IFC uses multiplexed address and data, special consideration is given to avoid bus contention at bus turnaround.

The following cases must be examined:

- Address phase after previous read
- Read-data phase after address phase
- Read-modify-write cycle for parity-protected memory banks

The bus does not change direction for the following cases, so no special attention is required:

- Continued burst after the first beat
- Write-data phase after address phase
- Address phase after previous write

17.11.8.1 Address phase after previous read

During a read cycle, the memory/peripheral drives the bus and the bus transceiver drives AD.

After the data has been sampled, the output drivers of the external device must be disabled, which can take some time.

After the previous cycle ends, BCTL goes high and changes the direction of the bus transceiver. The IFC then inserts a bus turnaround time (that is, TBCTL) to avoid contention. The external device has now already placed its data signals in high impedance and no bus contention occurs.

17.11.8.2 Read-data phase after address phase

During the address phase, AD actively drives the address and BCTL is high, driving the bus transceivers in the same direction as during a write.

After the end of the address phase, BCTL goes low and changes the direction of the bus transceiver.

17.11.8.3 Read-modify-write cycle for parity protected memory banks

Principally, a read-modify-write cycle is a read cycle immediately followed by a write cycle. Because the write cycle has a new address phase in any case, this essentially is the same as an address phase after a previous read.

17.11.9 Interfacing to different port sizes

The IFC supports 8- 16-bit data port sizes.

However, the IFC requires that the portion of the data bus used for a transfer to or from a particular port size be fixed. A 16-bit port must reside on AD[15:0], and an 8-bit port on AD[7:0]. This figure shows the device connections on the data bus.

Figure 17-64. Interface to different port-size devices

17.11.10 Command sequence examples for NAND flash EEPROM

To program the IFC and FCM for executing NAND flash command sequences, obtain command codes and pause states from the relevant NAND flash device datasheet and program into FCM configuration registers.

This section describes some common sequences for multi-gigabit NAND flash EEPROMs; however, details should be verified against manufacturer's specific programming data.

Throughout these examples, it is assumed that one or more banks of the IFC have been configured under FCM control with base address, port size, ECC mode, and timing parameters configured in accordance with the device's data sheet .

17.11.10.1 NAND flash soft reset command sequence example

An example of configuring FCM to execute a soft reset command to a 2 KB-page NAND flash is shown in this table.

This sequence does not require the use of the shared FCM buffer RAM. At the conclusion of the sequence, IFC issues a command complete interrupt if interrupts are enabled.

Table 17-8. FCM register settings for soft reset

Register	Initial contents	Description
NAND_FCR0	FF00_0000h	CMD0 = FFh = reset command; other commands unused
ROW0	-	Unused
COL0	-	Unused
NAND_BC	-	Unused
NAND_FSR	-	Unused
NAND_FIR0, NAND_FIR1, NAND_FIR2	2400_0000h, 0000_0000h, 0000_0000h	OP0 = CMD0 = command 0; OP1-OP14 = NOOP

17.11.10.2 NAND flash read status command sequence example

An example of configuring FCM to execute a status read command to 2 KB-page NAND flash is shown in this table.

This sequence does not require the use of the shared FCM buffer RAM, but reads the NAND flash status into NAND_FSR.

At the conclusion of the sequence, the IFC issues a command complete interrupt if interrupts are enabled.

Table 17-9. FCM register settings for status read

Register	Initial contents	Description
NAND_FCR0	7000_0000h	CMD0 = 70h = read status command; other commands unused
ROW0	-	Unused
COL0	-	Unused
NAND_BC	-	Unused
NAND_FSR	-	Status returned in RS0
NAND_FIR0, NAND_FIR1, NAND_FIR2	25C0_0000h, 0000_0000h, 0000_0000h	OP0 = CMD0 = command 0; OP1 = RDSTAT = read status to NAND_FSR; OP2-OP14 = NOOP

17.11.10.3 NAND flash read identification command sequence example

An example of configuring FCM to execute a status ID command to 2 KB-page NAND flash is shown in this table.

This sequence uses the shared FCM buffer RAM to receive the bytes of ID during the sequence.

At the conclusion of the sequence, IFC issues a command complete interrupt if interrupts are enabled.

Table 17-10. FCM register settings for ID read

Register	Initial contents	Description
NAND_FCR0	9000_0000h	CMD0 = 90h Read ID command; other commands unused
ROW3	Row address (0000_0020h)	Locates the block and the page within that block to be accessed
COL0	0000_0000h	Locates the byte to be accessed within a given page MS = 0
NAND_BC	0000_0004h	BC = 4 to read the ONFI Signature (for ONFI devices)
NAND_FSR	-	Unused
NAND_FIR0, NAND_FIR1, NAND_FIR2	2608_4000h, 0000_0000h, 0000_0000h	OP0 = CMD0 = command 0; OP1 = UA = Send row address programmed in row address register 3. Use SRAM buffer 0 to store the read ID bytes; OP2 = RB = Read BC bytes of data into the SRAM buffer 0; OP3-OP14 = NOOP

The above sequence uses opcode UA to send the address stored in row address register 3. This is done so that the data transfer, that is, the read data is stored in buffer 0 of the internal SRAM. Thus, the UA could also be used for operations, such as read page parameter, get feature, set feature, read unique ID always using buffer 0 of the internal SRAM.

17.11.10.4 NAND flash page read command sequence example

An example of configuring FCM to execute a random page read command to 2-Kbytes page NAND flash is shown in the table below.

This sequence reads an entire page (main and spare region) into the shared FCM buffer RAM, checking ECC (if enabled) as it proceeds.

At the conclusion of the sequence, IFC will issue a command complete interrupt if interrupts are enabled. Once the sequence has completed, the shared buffer (buffer 1 for page index 5) and transfer error registers are valid.

Table 17-11. FCM register settings for page read

Register	Initial contents	Description
NAND_FCR0	0030_0000h	CMD0 = 00h = random read address entry; CMD1 = 30h = read page
ROW0	row address (for example, 0000_0005h locates page5 in block0)	Locates the block and the page within that block to be accessed
COL0	0000_0000h	Locates the byte to be accessed within a given page MS = 0
NAND_BC	0000_0000h	BC = 0 to read entire 2112-byte page
NAND_FSR	-	unused
NAND_FIR0, NAND_FIR1,NAND_FIR2	2411_4A68h, 0000_0000h, 0000_0000h	OP0 = CMD0 = command 0; OP1 = CA0 = column address; OP2 = RA0 = page address; OP3 = CMD1 = command 1; OP4 = RBCD = read BC bytes of data into FCM buffer; OP5 - OP14 = NOOP

17.11.10.5 NAND flash program command sequence example

An example of configuring FCM to execute a program command to 2 KB-page NAND flash is shown in this table.

This sequence writes an entire page (main and spare region) from the shared FCM buffer RAM, generating ECC (if enabled) as it proceeds.

The shared buffer (buffer 1 for page index 5) must be initialized by software prior to starting the sequence. At the conclusion of the sequence, the IFC issues a command complete interrupt if interrupts are enabled. The status of the programming operation is returned in NAND_FSR.

Note that operations specified by OP5 and OP6 (status read) should never be skipped while programming a NAND flash device, because it may happen that a new command is issued to the NAND flash device even when the device has not yet finished processing the previous request. This may result in unpredictable behavior.

Table 17-12. FCM register settings for page program

Register	Initial contents	Description
NAND_FCR0	8070_1000h	CMD0 = 80h = page address and data entry; CMD1 = 70h = read status CMD2 = 10h = program page;
ROW0	Row address (for example, 0000_0005h locates page5 in block0)	Locates the block and the page within that block to be accessed
COL0	0000_0000h	Locates the byte to be accessed within a given page MS = 0
NAND_BC	0000_0000h	BC = 0 to read entire 2112-byte page
NAND_FSR	-	Returns with AS0 holding program status
NAND_FIR0, NAND_FIR1, NAND_FIR2	2411_592Ch, 49C0_0000h, 0000_0000h	OP0 = CMD0 = command 0; OP1 = CA0 = column address; OP2 = RA0 = page address; OP3 = WBCD = write BC bytes of data from buffer; OP4 = CMD2 = command 2; OP5 = CW1 = wait on flash ready and issue command 1; OP6 = RDSTAT = read erase status into NAND_FSR; OP7-OP14 = NOOP

17.11.10.6 Read status command during busy period of program/ erase operation

During program and erase operation, the device permits the user to read status during its busy period.

To achieve this, use the sequence of commands in this table.

Table 17-13. FCM register settings for read status during program busy period

Register	Initial contents	Description
NAND_FCR0	8070_1000h	CMD0 = 80h = page address and data entry; CMD1 = 0x70 = read status CMD2 = 10h = program page;
ROW0	row address	Locates the block and the page within that block to be accessed

Table continues on the next page...

Table 17-13. FCM register settings for read status during program busy period (continued)

Register	Initial contents	Description
	(for example, 0000_0005h locates page5 in block0)	
COL0	0000_0000h	Locates the byte to be accessed within a given page MS = 0
NAND_BC	0000_0000h	BC = 0 to read entire 2112-byte page
NAND_FSR	-	Returns with AS0 holding program status
NAND_FIR0, NAND_FIR1, NAND_FIR2	2411_592Ch, 49C00_000h, 00000_000h	OP0 = CMD0 = command 0; OP1 = CA0 = column address; OP2 = RA0 = page address; OP3 = WBCD = write BC bytes of data from buffer; OP4 = CMD2 = command 2; OP5 = NWAIT = program NCFGR[NUM_WAIT] for tWB time. OP6 = CMD1 = command 1; OP7 = RDSTAT = read erase status into NAND_FSR; OP8–OP14 = NOOP

In this sequence, use opcode (NWAIT, CMD1) instead of CW1 to issue a read status during the device-busy phase. This is different from the sequence NAND flash program command sequence example, where a read status is issued after the device becomes ready.

17.11.10.7 Valid opcode transitions in the IFC

This table describes the valid opcode transitions.

Ensure that the FIR sequence is programmed using only valid opcode transitions.

Table 17-14. Supported opcode transitions (in FIR)

Current opcode	Next opcode supported *n*=0-7, *m*=0-3								
CMD*n*	CA*m*/ RA*m*	CMD*n*	UA	RBCD/ BTRD/ SBRD/ RDSTAT/ RB_B	CW*n*	WFR	NWAIT	NOOP	
CA*m*	CA*m*/ RA*m*	CMD*n*	UA	RDSTAT/ RB	CW*n*	WFR	NWAIT	NOOP	
RA*m*	RA*m*	CMD*n*	UA	WBCD	RDSTAT/R B	CW*n*	WFR	NWAIT	NO OP
UA*m*	RA*m*	CMD*n*	UA	WBCD	RDSTAT/R B	CW*n*	WFR	NWAIT	NO OP

Table continues on the next page...

Table 17-14. Supported opcode transitions (in FIR) (continued)

Current opcode	Next opcode supported *n*=0-7, *m*=0-3								
WBCD	CMD*n*	CW*n*	WFR	NWAIT	NOOP				
CW*n*	CA*m*/ RA*m*	CMD*n*	UA	RBCD/ BTRD/ SBRD/ RDSTAT/ RB_B	CW*n*	WFR	NWAIT	NOOP	
WFR	CMD*n*	RBCD/ SBRD	NWAIT	NOOP	-	-	-	-	
NWAIT	CMD*n*	CA*m*/ RA*m*	UA	WBCD	RDSTAT/R B	CW*n*	WFR	NOOP	
RBCD/ BTRD/ SBRD/ RDSTAT/ RB	RDSTAT	CMD*n*	CW*n*	CA*m*/ RA*m*	NWAIT	NOOP	-	-	

Chapter 18
Inter-Integrated Circuit (I2C)

18.1 The I2C module as implemented on the chip

This section provides details about how the I2C module is integrated into this chip.

The chip implements four I2C modules, I2C1 and I2C2, I2C3 and I2C4.

Table 18-1. I2C module as implemented on chip

Module name	Module base address
I2C1	0200_0000h
I2C2	0201_0000h
I2C3	0202_0000h
I2C4	0203_0000h

18.2 Overview

This chapter describes the Inter-Integrated Circuit (I^2C) module implemented on this chip and presents the following topics:

- Introduction to I^2C
- External signal descriptions
- I2C register descriptions
- Functional description
- Initialization/application information

18.3 Introduction to I^2C

This section presents the following topics:

- Definition: I²C module
- Module block diagram
- Features
- Modes of operation
- Definition: I²C conditions

18.3.1 Definition: I²C module

The I²C module is a functional unit that provides a two-wire—serial data (SDA) and serial clock (SCL)—bidirectional serial bus. This bus provides a simple and efficient method of data exchange between this chip and other devices, such as microcontrollers, EEPROMs, real-time clock devices, analog-to-digital converters, and LCDs.

18.3.2 Module block diagram

Figure 18-1. I²C block diagram

18.3.3 Features

The I²C module has the following key features:

- Compatible with I^{2}C bus standard[1]
- Operating speeds
  - Up to 100 kbps in the Standard Mode defined by the I^{2}C protocol specification
  - Up to 400 kbps in the Fast Mode defined by the I^{2}C protocol specification
  - Operation at higher baud rates (up to a maximum of module clock/20) with reduced bus loading
  - Actual baud rate dependent on the SCL rise time (which depends on external pullup resistor values and bus loading)
- Multi-master operation
- Software-programmable for one of 256 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven byte-by-byte data transfer
- Arbitration lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated start signal generation
- Acknowledge bit generation/detection
- Bus busy detection

18.3.4 Modes of operation

The I^{2}C module supports the chip modes described in Table 18-2.

Table 18-2. Chip modes supported by the I^{2}C module

Chip mode	Description	Important notes
RUN	Basic mode of operation	—
STOP	The lowest-power mode that allows the chip to turn off all the clocks to the I^{2}C module	The I^{2}C module can enter this mode when there are no active transfers (active data between valid START and STOP conditions) on the bus. See STOP mode.

Table continues on the next page...

1. Compliant with I^{2}C 2.0 standard with the exception that HS (high speed) mode is not supported

Table 18-2. Chip modes supported by the I²C module (continued)

Chip mode	Description	Important notes
DEBUG	Allows the chip to freeze all ongoing activities (such as an ongoing transaction, counter values, and register status) for debugging	See DEBUG mode.

In addition to chip modes, the I²C module has several module-specific modes. These are described in Table 18-3.

Table 18-3. Module-specific modes supported by the I²C module

Module mode	Description	Important notes
Master mode	The I²C module is the driver of the SDA line.	• Do not use the I²C module's slave address as a calling address. • The I²C module cannot be a master and a slave simultaneously.
Slave mode	The I²C module is not the driver of the SDA line.	• Enable the I²C module before a START condition from a non-I²C master is detected. • By default the I²C module performs as a slave receiver.

18.3.5 Definition: I²C conditions

The following table shows the I²C-specific conditions defined for the I²C module.

Table 18-4. I²C conditions

Condition	Description
START	A condition that denotes the beginning of a new data transfer and awakens all slaves. Each data transfer contains several bytes of data. It is defined as a high-to-low transition of SDA while SCL is high, as shown in the following figure.
STOP	A condition generated by the master to terminate a transfer and free the bus. It is defined as a low-to-high transition of SDA while SCL is high, as shown in the following figure.
Repeated START	A START condition that is generated without a STOP condition to terminate the previous transfer.

Figure 18-2. START and STOP conditions

18.4 External signal descriptions

This section presents the following topics:

- Signal overview
- Detailed external signal descriptions

18.4.1 Signal overview

The I^2C module uses the Serial Data (SDA) and Serial Clock (SCL) signals as a communication interconnect with other devices. The signal patterns driven on the SDA signal represent address, data, or read/write information at different stages of the protocol.

All devices connected to the SDA and SCL signals must have open-drain or open-collector outputs. The I^2C bus connects the SDA lines of all devices together, thus performing a logical AND of the SDA signals. Similarly, the I^2C bus connects all the SCL lines together, thus performing a logical AND of the SCL signals. The combined SDA and SCL lines also connect to their respective pullup resistors. For the electrical characteristics of these signals, see the data sheet for this chip.

18.4.2 Detailed external signal descriptions

The SDA and SCL signals are described in the following table.

Table 18-5. External signal descriptions

Signal	Description
SCL	Bidirectional serial clock line of the module, compatible with the I^2C bus specification
SDA	Bidirectional serial data line of the module, compatible with the I^2C bus specification

18.5 I2C register descriptions

The memory map for the I^2C module is given below. The total address for each register is the sum of the base address for the I^2C module and the address offset for each register.

Address location 0x0007 is a reserved location, but access to this location will not generate any bus error.

All the I^2C registers are one byte wide. Reads and writes to these registers must be byte-wide operations.

18.5.1 I2C memory map

I2C1 base address: 200_0000h

I2C2 base address: 201_0000h

I2C3 base address: 202_0000h

I2C4 base address: 203_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	I2C Bus Address Register (IBAD)	8	RW	00h
1h	I2C Bus Frequency Divider Register (IBFD)	8	RW	00h
2h	I2C Bus Control Register (IBCR)	8	RW	80h
3h	I2C Bus Status Register (IBSR)	8	W1C	80h
4h	I2C Bus Data I/O Register (IBDR)	8	RW	00h
5h	I2C Bus Interrupt Config Register (IBIC)	8	RW	00h
6h	I2C Bus Debug Register (IBDBG)	8	RW	00h

18.5.2 I2C Bus Address Register (IBAD)

18.5.2.1 Offset

Register	Offset
IBAD	0h

18.5.2.2 Function

This register contains the address that the I^2C bus responds to when addressed as a slave. This is not the address sent on the bus during the address transfer.

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.2.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	ADR							0
W								
Reset	0	0	0	0	0	0	0	0

18.5.2.4 Fields

Field	Function
7-1 ADR	Slave address Specific slave address to be used by the I^2C module. **NOTE:** The default mode of I^2C bus is slave mode for an address match on the bus.
0 —	Reserved

18.5.3 I2C Bus Frequency Divider Register (IBFD)

18.5.3.1 Offset

Register	Offset
IBFD	1h

18.5.3.2 Function

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.3.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	IBC							
Reset	0	0	0	0	0	0	0	0

18.5.3.4 Fields

Field	Function
7-0 IBC	IBC I-Bus clock rate. This field is used to prescale the bus clock for bit rate selection. See Clock rate and IBFD settings.

18.5.4 I2C Bus Control Register (IBCR)

18.5.4.1 Offset

Register	Offset
IBCR	2h

18.5.4.2 Function

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.4.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MDIS	IBIE	MSSL	TXRX	NOACK	0	0	Reserved
W						RSTA		
Reset	1	0	0	0	0	0	0	0

18.5.4.4 Fields

Field	Function
7 MDIS	Module disable This field controls the software reset of the entire I^{2}C module. **NOTE:** If the I^{2}C module is enabled in the middle of a byte transfer, the interface behaves as follows: slave mode ignores the current transfer on the bus and starts operating whenever a subsequent START condition is detected. Master mode will not be aware that the bus is busy, hence if a start cycle is initiated, the current bus cycle may become corrupt. This ultimately results in either the current bus master or the I^{2}C module losing arbitration, after which bus operation returns to normal. 0b - The module is enabled. You must clear this field before any other IBCR fields have any effect. 1b - The module is reset and disabled. This is the power-on reset situation. When high, the interface is held in reset, but registers can still be accessed. Status register fields (IBSR) are not valid when the module is disabled.
6 IBIE	I2C bus interrupt enable 0b - Interrupts from the I2C module are disabled. This does not clear any currently pending interrupt condition. 1b - Interrupts from the I2C module are enabled. An I2C interrupt occurs if IBSR[IBIF]=1.
5 MSSL	Master/slave mode select When you change this field from 0 to 1, the module generates a START signal on the bus and selects the master mode. When you change this field from 1 to 0, the module generates a STOP signal and changes the operation mode from master to slave. You should generate a STOP signal only if IBSR[IBIF]=1. The module clears this field without generating a STOP signal when the master loses arbitration. 0b - Slave mode 1b - Master mode
4 TXRX	Transmit/receive mode select This field selects the direction of master and slave transfers. When the I^{2}C module is addressed as a slave, your software must program this field based on IBSR[SRW]:

Table continues on the next page...

Field	Function
	• When IBSR[SRW]=0, program TXRX=0. • When IBSR[SRW]=1, program TXRX=1. In master mode, your software must program this field according to the type of transfer required. Therefore, for address cycles, this field will always be 1. 0b - Receive 1b - Transmit
3 NOACK	Data acknowledge disable This field specifies the value driven onto SDA during data acknowledge cycles for both master and slave receivers. When the I^2C module is enabled, it always acknowledges address matches, regardless of the value of NOACK. **NOTE:** Values written to this field are only used when the I^2C module is a receiver, not a transmitter. 0b - The module sends an acknowledge signal to the bus at the 9th clock bit after receiving one byte of data. 1b - The module does not send an acknowledge-signal response (that is, acknowledge bit = 1).
2 RSTA	Repeat START If the I^2C module is the current bus master, and you program RSTA=1, the I^2C module generates a repeated START condition. This field always reads as a 0. If you attempt a repeated START at the wrong time—if the bus is owned by another master—the result is loss of arbitration. 0b - No effect 1b - Generate repeated START condition
1 —	Reserved
0 —	Reserved Although this field supports read/write access, writing to this field is not required, and can produce unexpected results.

18.5.5 I2C Bus Status Register (IBSR)

18.5.5.1 Offset

Register	Offset
IBSR	3h

18.5.5.2 Function

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.5.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	TCF	IAAS	IBB	IBAL	0	SRW	IBIF	RXAK
W				W1C			W1C	
Reset	1	0	0	0	0	0	0	0

18.5.5.4 Fields

Field	Function
7 TCF	Transfer complete The I²C module programs this field to 0 during the time that the module transfers one byte of data. After the I²C module detects the falling edge of the 9th clock of a byte transfer, the module programs this field to 1. **NOTE:** This field is relevant to you only during or immediately after a transfer to or from the I²C module. 0b - Transfer in progress 1b - Transfer complete
6 IAAS	Addressed as a slave When its own specific address (I-Bus Address Register) is matched with the calling address, this field is set. The CPU is interrupted if IBCR[IBIE]=1. Then the CPU needs to check SRW and set the TXRX field accordingly. Writing to the I-Bus Control Register clears IAAS. 0b - Not addressed 1b - Addressed as a slave
5 IBB	Bus busy This field indicates the status of the bus. When a START signal is detected, IBB is set. If a STOP signal is detected, IBB is cleared and the bus enters idle state. **NOTE:** Software must ensure that the I²C bus is idle by checking the IBSR[IBB] field (bus busy) before switching to master mode and attempting a START cycle. 0b - Bus is idle 1b - Bus is busy
4 IBAL	Arbitration Lost The I²C module programs IBAL=1 when the arbitration procedure is lost. Arbitration is lost in the following circumstances: • SDA is sampled low when the master drives a high during an address or data transmit cycle. • SDA is sampled low when the master drives a high during the acknowledge bit of a data receive cycle. • A start cycle is attempted when the bus is busy. • A repeated start cycle is requested in slave mode. • A stop condition is detected when the master did not request it. Your software must clear this field by writing a one to it. A write of zero has no effect.
3	Reserved

Table continues on the next page...

Field	Function
—	
2 SRW	Slave Read/Write When IAAS=1, SRW indicates the value of the R/W command bit of the calling address sent from the master. SRW is valid only when all of these are true: • The I-Bus is in slave mode. • A complete address transfer has occurred with an address match. • No other transfers have been initiated. By reading this field, the CPU can detect slave transmit/receive mode according to the command of the master. 0b - Slave receive, master writing to slave 1b - Slave transmit, master reading from slave
1 IBIF	I2C bus interrupt flag IBIF=1 when one of the following conditions occurs: • Arbitration lost (IBAL=1) • Byte transfer complete • Addressed as slave (IAAS=1) • NoAck from Slave (MS and Tx bits set) • I^2C Bus going idle (IBB high-low transition and enabled by BIIE) A processor interrupt request will be caused if IBIE=1. Your software must clear IBIF by writing a one to it. A write of zero has no effect. All other conditions still apply.
0 RXAK	Received acknowledge This is the value of SDA during the acknowledge bit of a bus cycle. If RXAK=0, it indicates an acknowledge signal has been received after the completion of eight bits of data transmission on the bus. If RXAK=1, it means no acknowledge signal is detected at the ninth clock. RXAK is valid only after transfer is complete. 0b - Acknowledge received 1b - No acknowledge received

18.5.6 I2C Bus Data I/O Register (IBDR)

18.5.6.1 Offset

Register	Offset
IBDR	4h

18.5.6.2 Function

In master transmit mode, when your software writes to IBDR, the I^2C module initiates a data transfer. The module sends the most significant bit first.

In master receive mode, when your software reads IBDR, the I^2C module initiates the reception of the next data byte.

In slave mode, the same functions are available after an address match has occurred.

NOTE

The IBCR[TXRX] field must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the I^2C is configured for master transmit but a master receive is desired, then reading the IBDR will not initiate the receive.

When the the I^2C module is configured for master-receive or slave-receive modes, and your software reads IBDR, the I^2C module returns the most-recent byte received. The IBDR does not reflect every byte transmitted on the I^2C bus. Your software also cannot verify that it wrote a byte to IBDR correctly by reading the byte back.

In master transmit mode, the I^2C module takes the first byte of data written to IBDR after assertion of MSSL and uses this byte for the address transfer. This byte must comprise the calling address (in position DATA[7:1]) concatenated with the required R/W_b bit (in position D0). (In the I^2C bus specification, the R/W_b bit is presented as R/W with an overbar on the W.)

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.6.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	DATA							
Reset	0	0	0	0	0	0	0	0

18.5.6.4 Fields

Field	Function
7-0 DATA	DATA Data transmitted or received

18.5.7 I2C Bus Interrupt Config Register (IBIC)

18.5.7.1 Offset

Register	Offset
IBIC	5h

18.5.7.2 Function

To program BIIE = 1, you must ensure that IBCR[MDIS] = 0.

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.7.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	BIIE	BYTERXIE	0					
W								
Reset	0	0	0	0	0	0	0	0

18.5.7.4 Fields

Field	Function
7 BIIE	Bus idle interrupt enable You can use this configuration field to enable the generation of an interrupt after the I^{2}C bus becomes idle. After BIIE=1, a high-to-low transition of IBSR[IBB] results in IBSR[IBIF]=1. You can use this feature to inform the CPU about the completion of a STOP on the I^{2}C bus. 0b - Bus idle interrupts disabled 1b - Bus idle interrupts enabled
6 BYTERXIE	Byte receive interrupt enable You can use this field to generate an interrupt every time the I^{2}C master/slave receives a new byte. This feature can be useful when an I^{2}C master receives data from a slave that transmits on an irregular basis.

Table continues on the next page...

Field	Function
	The I^2C module updates BYTERXIE only when the module is enabled (IBCR[MDIS]=0).
5-0 —	Reserved

18.5.8 I2C Bus Debug Register (IBDBG)

18.5.8.1 Offset

Register	Offset
IBDBG	6h

18.5.8.2 Function

Access: Supervisor mode only (the module ignores user mode accesses and does not assert an error response if you access this register in user mode)

18.5.8.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	0				GLFLT_EN	0	IPG_DEBUG_HALTED	IPG_DEBUG_EN
W								
Reset	0	0	0	0	0	0	0	0

18.5.8.4 Fields

Field	Function
7-4	Reserved

Table continues on the next page...

Field	Function
—	
3 GLFLT_EN	Glitch filter enable On this chip, you must program this field to 1.
2 —	Reserved
1 IPG_DEBUG_HALTED	Debug halted This is a status field that you can read back after asserting the debug enable signal. The field informs you whether the I^2C module entered DEBUG mode or not. 0b - The I2C module is still executing a transaction. 1b - The I2C module entered DEBUG mode.
0 IPG_DEBUG_EN	Debug enable This field allows the I^2C module to enter the DEBUG mode when the IPG DEBUG signal is high. All the registers, counter values and status bits are frozen and can be accessed by the CPU. **NOTE:** If your software programs IPG_DEBUG_EN=1, and the IPG DEBUG signal becomes high in the middle of a transaction, the I^2C module enters DEBUG mode only after successfully completing the current transaction, after which no further transactions can occur until the I^2C module exits DEBUG mode. 0b - Normal operation. The bus idle interrupts are disabled. 1b - The I2C module is in DEBUG mode.

18.6 Functional description

This section presents the following topics:

- Notes about module operation
- Transactions
- Arbitration procedure
- Clock behavior
- Interrupts
- STOP mode
- DEBUG mode

18.6.1 Notes about module operation

- The I^2C module always performs as a slave receiver by default, unless explicitly programmed to be a master or slave transmitter.
- When the I^2C module is acting as a master, it must not try to call its own slave address.

18.6.2 Transactions

This section presents the following topics:

- Protocol overview
- Transaction protocol definitions
- High-level protocol steps
- START condition
- Slave address transmission
- Data transmission
- STOP condition
- Repeated START condition

18.6.2.1 Protocol overview

The following figure shows the behavior of SCL and SDA during a typical I^2C transaction.

[1] In the I²C bus specification, this bit is labeled R/$\overline{W}$.

Figure 18-3. I²C transaction protocol

18.6.2.2 Transaction protocol definitions

This section defines several important terms presented in Figure 18-3.

Table 18-6. I²C definitions

Term	Definition
START	A START condition, as defined in Definition: I²C conditions
STOP	A STOP condition, as defined in Definition: I²C conditions
Calling (slave) address	A seven-bit address used to identify a slave on the I²C bus (see I²C calling address requirements for the requirements for specifying this address)
Read/write (R/W)	A bit that specifies the direction of the data transfer to the slave as follows: • 0=The data is being transferred from the master to the slave ("write") • 1=The data is being transferred from the slave to the master ("read")
Ack	A bit that specifies the acknowledgement of a calling address, indicated by pulling SDA low

18.6.2.3 I²C calling address requirements

The calling addresses of the devices used on an I²C network are subject to the following requirements:

- Each slave must have a unique calling address.
- A master must not transmit a calling address that is the same as its own slave address.

18.6.2.4 High-level protocol steps

The I^{2}C protocol conceptually supports two types of transfers, as illustrated in Figure 18-3. The significant steps in these transfers are presented in the following table. Details of each of these steps are presented in subsequent sections.

Table 18-7. I^{2}C high-level protocol steps

Standard transfer	Repeated START transfer
1. START condition	1. START condition
2. Slave target or general call address transmission	2. Slave target or general call address transmission
3. Acknowledgment from slave	3. Acknowledgment from slave
4. Data transfer	4. Data transfer
5. STOP condition	5. Repeated START condition
6. (repeat Steps 1-4)	6. (repeat Steps 2-4 as needed)
	7. STOP condition
	8. (repeat Steps 1-7)

18.6.2.5 START condition

When the bus is free (no master device engages the bus; both SCL and SDA lines are at logical high), a master can initiate communication by sending a START condition (see Definition: I^{2}C conditions). This signal denotes the beginning of a new data transfer (which may contain several bytes of data) and brings all slaves out of their idle states.

See Clock rate and IBFD settings and I2C Bus Frequency Divider Register (IBFD) for the associated timing requirements.

18.6.2.6 Slave address transmission

The master transmits the slave address on the next clock cycle after the START condition (see START condition). The following table presents the process of slave address transmission.

Table 18-8. Slave address transmission process

Step	Action
1	The master transmits the seven-bit slave address.
2	The master transmits the R/W bit.

Table continues on the next page...

Table 18-8. Slave address transmission process (continued)

Step	Action
3	Each slave examines the transmitted address and compares it to its own. If the addresses match, the slave device returns the acknowledge bit on the ninth SCL clock cycle.
4	The master waits for the acknowledge bit and determines the next step as follows: • The acknowledge bit is set: The master must initiate a data transfer followed by either a STOP condition or a repeated START condition. • The acknowledge bit is cleared: The master must wait for SCL to return to logic zero.

18.6.2.7 Data transmission

A data transfer session has the following characteristics:

- Can transmit one or more bytes of data
- Awakens all slaves
- Proceeds on a byte-by-byte basis in the direction specified by the R/W bit sent by the calling master

The transmitted data is subject to the following requirements:

- Each data byte must consist of 8 bits.
- Data bits can be changed only while SCL is low and must be held stable while SCL is high.
- One data bit is transmitted during one SCL clock pulse.
- The most significant bit (msb) must be transmitted first.
- Each data byte must be followed by an acknowledge bit on the ninth SCL clock pulse.

If the slave receiver does not acknowledge the master, the slave must leave the SDA line high. The master can then generate a STOP condition (to abort the data transfer) or a repeated START condition.

If the master receiver does not acknowledge the slave transmitter after a byte of transmission, the slave interprets that as reaching the end of data. The slave then releases the SDA line for the master to generate a STOP or a START condition.

18.6.2.8 STOP condition

The master can terminate the communication by generating a STOP condition (see Definition: I^{2}C conditions). It can do so even if the slave has generated an acknowledge, at which point the slave must release the bus.

A master is not required to send a STOP condition at the end of every transfer. For more information, see Repeated START condition.

See Clock rate and IBFD settings and I2C Bus Frequency Divider Register (IBFD) for the associated timing requirements.

18.6.2.9 Repeated START condition

The I^{2}C protocol also supports a repeated START condition, which can be generated without a preceding STOP condition. A master device can use this condition to communicate with another slave or with the same slave in a different mode without releasing the bus. This condition is illustrated in the second timing diagram of Figure 18-3.

18.6.3 Arbitration procedure

The I^{2}C bus is a true multi-master bus that allows more than one master to be connected to the bus. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. For this clock, the low period is equal to the longest clock low period and the high period is equal to the shortest period among the masters. A data arbitration procedure determines the relative priority of the contending masters. A bus master loses arbitration if it transmits logic "1" while another master transmits logic "0". The losing masters immediately switch over to slave mode and stop driving the SDA output. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, the I^{2}C module sets IBSR[IBAL] to indicate loss of arbitration.

18.6.4 Clock behavior

This section presents the following topics:

- Clock synchronization
- Clock stretching

- Handshaking
- Clock rate and IBFD settings

18.6.4.1 Clock synchronization

Due to the wired-AND logic on the SCL line, a high-to-low transition on the SCL line affects all devices connected on the bus. The devices begin counting their low period when the master drives the SCL line low. After a device has driven SCL low, it holds the SCL line low until the clock high state is reached.

However, the change of low-to-high in a device clock may not change the state of the SCL line if another device is still within its low period. Therefore, the synchronized clock signal, SCL, is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see the following figure). When all devices concerned have counted off their low period, the synchronized SCL line is released and pulled high. Then there is no difference between the devices' clocks and the state of the SCL line, and all the devices begin counting their high periods. The first device to complete its high period pulls the SCL line low again.

Figure 18-4. I²C bus clock synchronization

18.6.4.2 Clock stretching

Slaves can use the clock synchronization mechanism to slow down the bit rate of a transfer. After the master drives SCL low, the slave can drive SCL low for the required period and then release it. If the slave's SCL low period is greater than the master's SCL low period, the resulting SCL bus signal low period is stretched.

18.6.4.3 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. Slave devices may hold the SCL low after completion of one byte transfer (nine bits). In such cases, it halts the bus clock and forces the master clock into wait state until the slave releases the SCL line.

18.6.4.4 Clock rate and IBFD settings

18.6.4.4.1 Timing definitions

Table 18-9. Timing definitions relevant to clock rate and IBFD settings

Term	Definition
SCL Divider	The factor used to prescale the CPU clock for bit rate selection. This is relevant to: • Figure 18-5 • Table 18-11 • Table 18-12
SCL period	(CPU clock period)×(SCL Divider)
SCL Hold	The required number of CPU clocks to generate a START or STOP condition. This is relevant to: • Figure 18-5 • Table 18-11 • Table 18-12
SDA Hold	This is relevant to: • Figure 18-6 • Table 18-11 • Table 18-12

Figure 18-5. SCL Divider and SDA Hold

Figure 18-6. SDA Hold time

18.6.4.4.2 Divider and hold values

Table 18-11 provides the values for:

- SCL Divider
- SDA Hold
- SCL Hold (start)
- SCL Hold (stop)

when the glitch filter is disabled. (To disable this filter, write IBDBG[GLFLT_EN]=0.)

You have up to three MUL options available for all divider values, as shown in Table 18-10. Your choice of MUL determines the internal monitor rate of the I^2C bus (SCL and SDA signals):

- A lower MUL value results in a higher sampling rate of the I^2C signals.
- A higher MUL value results in a lower sampling rate of the I^2C signals. This gives the I^2C module greater immunity against glitches in the I^2C signals.

Table 18-10. MUL values as a function of IBC

When IBC is...	MUL is...
00h–3Fh	1
40h–7Fh	2
80h–BFh	4

NOTE

The frequency divider values listed in Table 18-11 and Table 18-12 depend on the board parasitics at the SCL pad. The observed I^2C frequency may differ from the expected frequency by 10–15%.

Table 18-11. I^2C divider and hold values when glitch filter is disabled

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
00	34	7	6	11
01	36	7	7	12
02	38	8	8	13
03	40	8	9	14
04	42	9	10	15
05	44	9	11	16
06	46	10	13	18
07	54	10	16	21
08	44	7	10	15
09	48	7	12	17
0A	52	9	14	19
0B	56	9	16	21

Table continues on the next page...

Table 18-11. I²C divider and hold values when glitch filter is disabled (continued)

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
0C	60	11	18	23
0D	64	11	20	25
0E	72	13	24	29
0F	84	13	30	35
10	64	9	18	25
11	72	9	22	29
12	80	13	26	33
13	88	13	30	37
14	96	17	34	41
15	104	17	38	45
16	120	21	46	53
17	144	21	58	65
18	96	9	38	41
19	112	9	46	49
1A	128	17	54	57
1B	144	17	62	65
1C	160	25	70	73
1D	176	25	78	81
1E	208	33	94	97
1F	256	33	118	121
20	160	17	78	81
21	192	17	94	97
22	224	33	110	113
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161
26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2F	960	129	478	481
30	640	65	318	321
31	768	65	382	385

Table continues on the next page...

Table 18-11. I²C divider and hold values when glitch filter is disabled (continued)

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
32	896	129	446	449
33	1024	129	510	513
34	1152	193	574	577
35	1280	193	638	641
36	1536	257	766	769
37	1920	257	958	961
38	1280	129	638	641
39	1536	129	766	769
3A	1792	257	894	897
3B	2048	257	1022	1025
3C	2304	385	1150	1153
3D	2560	385	1278	1281
3E	3072	513	1534	1537
3F	3840	513	1918	1921
40	68	14	12	22
41	72	14	14	24
42	76	16	16	26
43	80	16	18	28
44	84	18	20	30
45	88	18	22	32
46	96	20	26	36
47	108	20	32	42
48	88	14	20	30
49	96	14	24	34
4A	104	18	28	38
4B	112	18	32	42
4C	120	22	36	46
4D	128	22	40	50
4E	144	26	48	58
4F	168	26	60	70
50	128	18	36	50
51	144	18	44	58
52	160	26	52	66
53	176	26	60	74
54	192	34	68	82
55	208	34	76	90
56	240	42	92	106
57	288	42	116	130

Table continues on the next page...

Table 18-11. I²C divider and hold values when glitch filter is disabled (continued)

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
58	192	18	76	82
59	224	18	92	98
5A	256	34	108	114
5B	288	34	124	130
5C	320	50	140	146
5D	356	50	156	162
5E	416	66	188	194
5F	512	66	236	242
60	320	34	156	162
61	384	34	188	194
62	448	66	220	226
63	512	66	252	258
64	576	98	284	290
65	640	98	316	322
66	768	130	380	386
67	960	130	476	482
68	640	66	316	322
69	768	66	380	386
6A	896	130	444	450
6B	1024	130	508	514
6C	1152	194	572	578
6D	1280	194	636	642
6E	1536	258	764	770
6F	1920	258	956	962
70	1280	130	636	642
71	1536	130	764	770
72	1792	258	892	898
73	2048	258	1020	1026
74	2304	386	1148	1154
75	2560	386	1276	1282
76	3072	514	1532	1538
77	3840	514	1916	1922
78	2560	258	1276	1282
79	3072	258	1532	1538
7A	3584	514	1788	1794
7B	4096	514	2044	2050
7C	4608	770	2300	2306
7D	5120	770	2556	2562

Table continues on the next page...

Table 18-11. I²C divider and hold values when glitch filter is disabled (continued)

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
7E	6144	1026	3068	3074
7F	7680	1026	3836	3842
80	136	28	24	44
81	144	28	28	48
82	152	32	32	52
83	160	32	36	56
84	168	36	40	60
85	176	36	44	64
86	192	40	52	72
87	216	40	64	84
88	176	28	40	60
89	192	28	48	68
8A	208	36	56	76
8B	224	36	64	84
8C	240	44	72	92
8D	256	44	80	100
8E	288	52	96	116
8F	336	52	120	140
90	256	36	72	100
91	288	36	88	116
92	320	52	104	132
93	352	52	120	148
94	384	68	136	164
95	416	68	152	180
96	480	84	184	212
97	576	84	232	260
98	358	36	152	164
99	448	36	184	196
9A	512	68	216	228
9B	576	68	248	260
9C	640	100	280	292
9D	704	100	312	324
9E	832	132	376	388
9F	1024	132	472	484
A0	640	68	312	324
A1	768	68	376	388
A2	896	132	440	452
A3	1024	132	504	516

Table continues on the next page...

Table 18-11. I²C divider and hold values when glitch filter is disabled (continued)

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
A4	1152	196	568	580
A5	1280	196	632	644
A6	1536	260	760	772
A7	1920	260	952	964
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
B0	2560	260	1272	1284
B1	3072	260	1528	1540
B2	3584	516	1784	1796
B3	4096	516	2040	2052
B4	4608	772	2296	2308
B5	5120	772	2552	2564
B6	6144	1028	3064	3076
B7	7680	1028	3832	3844
B8	5120	516	2552	2564
B9	6144	516	3064	3076
BA	7168	1028	3576	3588
BB	8192	1028	4088	4100
BC	9216	1540	4600	4612
BD	10240	1540	5112	5124
BE	12288	2052	6136	6148
BF	15360	2052	7672	7684

Table 18-12 provides the values for:

- SCL Divider
- SDA Hold
- SCL Hold (start)
- SCL Hold (stop)

when the glitch filter is enabled. (To enable this filter, write IBDBG[GLFLT_EN]=1.)

Table 18-12. I²C divider and hold values when glitch filter is enabled

IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
00	34	14	6	18
01	36	14	7	19
02	38	15	8	20
03	40	15	9	21
04	42	16	10	22
05	44	16	11	23
06	48	17	13	25
07	54	17	16	28
08	44	15	10	23
09	48	15	12	25
0A	52	17	14	27
0B	56	17	16	29
0C	60	19	18	31
0D	64	19	20	33
0E	72	21	24	37
0F	84	21	30	43
10	64	17	18	33
11	72	17	22	37
12	80	21	26	41
13	88	21	30	45
14	96	25	34	49
15	104	25	38	53
16	120	29	46	61
17	144	29	58	73
18	96	17	38	49
19	112	17	46	57
1A	128	25	54	65
1B	144	25	62	73
1C	160	33	70	81
1D	176	33	78	89
1E	208	41	94	105
1F	256	41	11	129
20	160	17	78	81
21	192	17	94	97
22	224	33	110	113
23	256	33	126	129
24	288	49	142	145
25	320	49	158	161

Table continues on the next page...

Table 18-12. I²C divider and hold values when glitch filter is enabled (continued)

26	384	65	190	193
27	480	65	238	241
28	320	33	158	161
29	384	33	190	193
2A	448	65	222	225
2B	512	65	254	257
2C	576	97	286	289
2D	640	97	318	321
2E	768	129	382	385
2F	960	129	478	481
30	640	65	318	321
31	768	65	382	385
32	896	129	446	449
33	1024	129	510	513
34	1152	193	574	577
35	1280	193	638	641
36	1536	257	766	769
37	1920	257	958	961
38	1280	129	638	641
39	1536	129	766	769
3A	1792	257	894	897
3B	2048	257	1022	1025
3C	2304	385	1150	1153
3D	2560	385	1278	1281
3E	3072	513	1534	1537
3F	3840	513	1918	1921
40	68	28	12	36
41	72	28	14	38
42	76	30	16	40
43	80	30	18	42
44	84	32	20	44
45	88	32	22	46
46	96	34	26	50
47	108	34	32	56
48	88	30	20	46
49	96	30	24	50
4A	104	34	28	54
4B	112	34	32	58
4C	120	38	36	62
4D	128	38	40	66

Table continues on the next page...

Table 18-12. I²C divider and hold values when glitch filter is enabled (continued)

4E	144	42	48	74
4F	168	42	60	86
50	128	34	36	66
51	144	34	44	74
52	160	42	52	82
53	176	42	60	90
54	192	50	68	98
55	208	50	76	106
56	240	58	92	122
57	288	58	116	146
58	192	34	76	98
59	224	34	92	114
5A	256	50	108	130
5B	288	50	124	146
5C	320	66	140	162
5D	352	66	156	178
5E	416	82	188	210
5F	512	82	236	258
60	320	34	156	162
61	384	34	188	194
62	448	66	220	226
63	512	66	252	258
64	576	98	284	290
65	640	98	316	322
66	768	130	380	386
67	960	130	476	482
68	640	66	316	322
69	768	66	380	386
6A	896	130	444	450
6B	1024	130	508	514
6C	1152	194	572	578
6D	1280	194	636	642
6E	1536	258	764	770
6F	1920	258	956	962
70	1280	130	636	642
71	1536	130	764	770
72	1792	258	892	898
73	2048	258	1020	1026
74	2304	386	1148	1154
75	2560	386	1276	1282

Table continues on the next page...

Table 18-12. I²C divider and hold values when glitch filter is enabled (continued)

76	3072	514	1532	1538
77	3840	514	1916	1922
78	2560	258	1276	1282
79	3072	258	1532	1538
7A	3584	514	1788	1794
7B	4096	514	2044	2050
7C	4608	770	2300	2306
7D	5120	770	2556	2562
7E	6144	1026	3068	3074
7F	7680	1026	3836	3842
80	136	56	24	72
81	144	56	28	76
82	152	60	32	80
83	160	60	36	84
84	168	64	40	88
85	176	64	44	92
86	192	68	52	100
87	216	68	64	112
88	176	60	40	92
89	192	60	48	100
8A	208	68	56	108
8B	224	68	64	116
8C	240	76	72	124
8D	256	76	80	132
8E	288	84	96	148
8F	336	84	120	172
90	256	68	72	132
91	288	68	88	148
92	320	84	104	164
93	352	84	120	180
94	384	100	136	196
95	416	100	152	212
96	480	116	184	244
97	576	116	232	292
98	384	68	152	196
99	448	68	184	228
9A	512	100	216	260
9B	576	100	248	292
9C	640	132	280	324
9D	704	132	312	356

Table continues on the next page...

Table 18-12. I²C divider and hold values when glitch filter is enabled (continued)

9E	832	164	376	420
9F	1024	164	472	516
A0	640	68	312	324
A1	768	68	376	388
A2	896	132	440	452
A3	1024	132	504	516
A4	1152	196	568	580
A5	1280	196	632	644
A6	1536	260	760	772
A7	1920	260	952	964
A8	1280	132	632	644
A9	1536	132	760	772
AA	1792	260	888	900
AB	2048	260	1016	1028
AC	2304	388	1144	1156
AD	2560	388	1272	1284
AE	3072	516	1528	1540
AF	3840	516	1912	1924
B0	2560	260	1272	1284
B1	3072	260	1528	1540
B2	3584	516	1784	1796
B3	4096	516	2040	2052
B4	4608	772	2296	2308
B5	5120	772	2552	2564
B6	6144	1028	3064	3076
B7	7680	1028	3832	3844
B8	5120	516	2552	2564
B9	6144	516	3064	3076
BA	7168	1028	3576	3588
BB	8192	1028	4088	4100
BC	9216	1540	4600	4612
BD	10240	1540	5112	5124
BE	12288	2052	6136	6148
BF	15360	2052	7672	7684

18.6.5 Interrupts

This section presents the following topics:

- Interrupt vector
- Interrupt description

18.6.5.1 Interrupt vector

The I^2C module uses only one interrupt vector.

Table 18-13. Interrupt summary

Interrupt	Offset	Vector	Priority	Source	Description
I^2C interrupt	—	—	—	IBAL, TCF, IAAS, IBB fields in IBSR	When any of IBAL, TCF, or IAAS fields are 1, the I^2C module may cause an interrupt based on Arbitration lost, Transfer Complete, or Address Detect conditions. If enabled by BIIE, the de-assertion of IBB can also cause an interrupt, indicating that the bus is idle.

18.6.5.2 Interrupt description

There are five types of internal interrupts in the I^2C. The interrupt service routine can determine the interrupt type by reading the Status register.

I^2C Interrupt can be generated on the following events:

- Arbitration Lost condition (IBSR[IBAL]=1)
- Byte Transfer condition (IBSR[TCF]=1)
- Address Detect condition (IBSR[IAAS]=1)
- No Acknowledge from slave received when expected
- Bus idle (IBSR[IBB]=0)

The IBCR[IBIE] field enables the I^2C interrupt. To clear this field, program IBSR[IBIF]=1 in the interrupt service routine.

To use the Bus idle interrupt, you must enable it by using the IBIC[BIIE] field.

18.6.6 STOP mode

This mode allows the software to put the I^{2}C module in power-down state. After the I^{2}C module receives a module-stop signal at the module's boundary, the I^{2}C module completes all ongoing transactions and comes to a graceful halt.

As soon as the I^{2}C module enters STOP mode:

- The I^{2}C clock is disabled.
- No transaction can take place.
- All registers are inaccessible.

The I^{2}C module enters this mode only after successfully completing the current ongoing transaction. Hence the low-power request is acknowledged once the STOP condition occurs. To request STOP for ongoing transmission, you must wait until the transmission is complete.

See the figure below for more details.

Figure 18-7. I^{2}C stop mode behavior when master is receiving and slave is transmitting

18.6.7 DEBUG mode

This mode allows the CPU to debug the I^{2}C module by freezing all the counters, registers, and status bits. To enter this mode, the I^{2}C module must receive a DEBUG mode request, defined as this combination of actions:

- Another module on the chip sends an internal debug signal to the I^{2}C module.
- You program IBDBG[IPG_DEBUG_EN]=1.

The I^{2}C module completes all the ongoing transactions and comes to a graceful halt after receiving a DEBUG mode request.

The I^{2}C module programs IBDBG[IPG_DEBUG_HALTED]=1 to indicate that the module has successfully serviced the DEBUG mode request.

The I^{2}C module programs IBDBG[IPG_DEBUG_HALTED]=0 after the module that sent the DEBUG mode request ends that request and you program IBDBG[IPG_DEBUG_EN]=0.

As soon as the I^{2}C module enters DEBUG mode:

- No transaction can take place.
- All the registers, counters and status bits are frozen. They all can be accessed by the CPU to enable the debugging.

The I^{2}C module enters this mode only after successfully completing the current ongoing transaction. For example, if the current transaction consists of eight bytes and you initiated DEBUG mode at the time of the second byte, the IPG Debug Halted signal will be asserted after the eighth byte is transmitted/received. See the simulation result in Figure 18-8 for more details.

Figure 18-8. Simulation result of IPG Debug Halted in case of frame transmission

No more transactions can occur until the module that sent the DEBUG mode request ends that request. At that time, the I^{2}C module starts functioning normally. You can examine the IBDBG[IPG_DEBUG_HALTED] field to determine whether the I^{2}C module is in DEBUG mode. See Figure 18-9 for more details.

Figure 18-9. DEBUG mode

Figure 18-10 shows a case of DEBUG mode with repeated START transaction. In this case, if another module on the chip sends an internal debug signal to the I^{2}C module in the middle of a transaction, the I^{2}C module completes the entire transaction with multiple repeated START before the module enters DEBUG mode.

[1] In the I²C bus specification, this bit is labeled R/$\overline{W}$.

Figure 18-10. DEBUG mode with repeated START

18.7 Initialization/application information

This section presents the following topics:

- Recommended interrupt service flow
- General programming guidelines (for both master and slave mode)
- Programming guidelines specific to Master mode
- Programming guidelines specific to Slave mode

18.7.1 Recommended interrupt service flow

The following figure shows a flowchart for the recommended I^2C interrupt service routine. Deviation from the flowchart may result in unpredictable I^2C bus behavior.

Figure 18-11. Recommended I²C interrupt service routine flowchart

18.7.2 General programming guidelines (for both master and slave mode)

This section provides programming guidelines recommended for the I²C module in both master and slave mode. It presents the following topics:

- Initializing the I²C module
- Software response after a transfer

18.7.2.1 Initializing the I²C module

The following table describes how to initialize the I²C module.

Table 18-14. I²C initialization procedure

Step	Action
1	Use I2C Bus Frequency Divider Register (IBFD) to select the required division ratio to obtain SCL frequency from the system clock.
2	Use I2C Bus Address Register (IBAD) to define the slave address.
3	Clear the MDIS field in I2C Bus Control Register (IBCR) to enable the I²C interface system.
4	Use I2C Bus Control Register (IBCR) to select Master/Slave mode, Transmit/Receive mode, and whether interrupts are enabled or disabled.
5	(Optional) Use I2C Bus Interrupt Config Register (IBIC) to further refine the interrupt behavior.

18.7.2.2 Software response after a transfer

Transmission or reception of a byte causes the I²C module to program IBSR[TCF]=1. This indicates that the I²C module transmitted one byte. The I²C module also programs the interrupt flag IBSR[IBIF]=1 and generates an interrupt if your software enabled the interrupt function (by programming IBCR[IBIE]=1) during initialization. Your software can clear IBSR[IBIF] by writing a 1 to that field (in the interrupt service routine, if interrupts are used).

The I²C module programs IBSR[TCF]=0 to indicate data transfer in progress whenever your software:

- Writes to IBDR in transmit mode
- Reads the IBDR in receive mode

Do not use IBSR[TCF] as a "data transfer complete" flag. This is because the flag timing depends on a number of factors, including the I²C bus frequency. This field may not conclusively provide an indication of a transfer-complete situation. Your software must detect transfer-complete situations by using IBSR[IBIF].

Your software may service the I²C I/O in the main program by monitoring IBSR[IBIF] if your software disabled the interrupt function. Your software must poll IBSR[IBIF] instead of IBSR[TCF] because these fields function differently when arbitration is lost.

When a "Transfer Complete" interrupt occurs at the end of the address cycle, the master will always be in transmit mode—that is, the I²C module transmits the address. If master receive mode is required, indicated by R/W bit sent with slave calling address, then the master's IBCR[TXRX] field must be toggled at this stage. If Master does not receive an ACK from Slave, then transmission must be re-initiated or terminated.

In slave mode, the I²C module programs IBSR[IAAS]=1 if the Slave address (I2C Bus Address Register (IBAD)) matches the Master calling address. This is an indication that Master-Slave data communication can now start. During address cycles

(IBSR[IAAS]=1), your software reads IBSR[SRW] to determine the direction of the subsequent transfer and programs IBCR[TXRX] accordingly. For slave mode data cycles (IBSR[IAAS]=0), IBSR[SRW] is not valid. Your software must read IBCR[TXRX] to determine the direction of the current transfer.

18.7.3 Programming guidelines specific to Master mode

This section presents the following topics:

- Generating START
- Transmit/receive sequence
- Generating STOP
- Generating repeated START
- Loss of arbitration

18.7.3.1 Generating START

After the I^2C module completes its initialization procedure, the module can transmit serial data when your software selects the master transmitter mode. If the device is connected to a multi-master bus system, your software must test the state of IBSR[IBB] to check whether the serial bus is free.

Your software can configure the I^2C module to send the START condition and the first byte (the slave address) if the bus is free (IBSR[IBB]=0). The data that your software writes to the data register (I2C Bus Data I/O Register (IBDR)) comprises the slave calling address and the LSB. Your software configures the LSB to indicate the direction of transfer required from the slave.

The bus free time (the time between a STOP condition and the following START condition) is built into the I^2C module that generates the START cycle. Depending on the relative frequencies of the system clock and the SCL period, your software must wait until the I^2C bus is busy after writing the calling address to the IBDR before proceeding with the next instructions. This is illustrated in the following example.

Here is an example of the sequence of events which generates the START signal and transmits the first byte of data (slave address):

```
while (IBSR[IBB]==1)                // wait in loop for IBB flag to clear
IBCR[MS/MSL] and IBCR[]Tx/Rx] = 1   // master and transmit mode, that is,
```

```
                                        // generate start condition
IBDR = calling_address                  // send the calling address to the data register
while (bit 5, IBSR ==0)                 // wait in loop for IBB flag to be set
```

18.7.3.2 Transmit/receive sequence

The following tables present the sequences for:

- Master transmit
- Master receive
- Slave transmit
- Slave receive

Table 18-15. Master transmit sequence

Step	Action
a	Use I2C Bus Frequency Divider Register (IBFD) to select the required division ratio to obtain SCL frequency from platform clock/2.
b	Write a 0 to IBCR[MDIS] to enable the I^2C interface system.
c	Use I2C Bus Control Register (IBCR) to select Master mode, Transmit mode, and interrupt enable.
d	Write a 0 to IBSR[IBIF].
e	Write data to I2C Bus Data I/O Register (IBDR).
f	Observe changes in IBSR[TCF] : • When IBSR[TCF] becomes 0, the transfer is in progress. • When IBSR[TCF] becomes 1, the transfer is complete.
g	Wait until IBSR[IBIF] becomes 1.
h	Read the fields in I2C Bus Status Register (IBSR) to determine what happened: • If TCF = 1, the transfer completed. • If RXAK = 1, a No Acknowledge condition occurred. • If IBB = 0, the bus transitioned from Busy to Idle state. • If IBB = 1, you can ignore check of Arbitration Loss (IBAL = 1). **NOTE:** You can ignore Address Detect (IAAS = 1) for master mode (it is valid only for slave mode).
i	Examine IBSR[RXAK] for an acknowledgment from the slave.
j	Repeat steps d through i to transfer the next consecutive bytes of data.

Table 18-16. Master receive sequence

Step	Action
a	Execute steps a through i in Table 18-15 for address dispatch.
b	Write a 0 to IBSR[IBIF].
c	Write a 0 IBCR[TXRX] to select Receive mode.
d	Perform a dummy read of I2C Bus Data I/O Register (IBDR) to initiate the receive operation.
e	Wait until IBSR[TCF] becomes 1. This proves that the transfer is complete.
f	Wait until IBSR[IBIF] becomes 1.
g	Read the fields in I2C Bus Status Register (IBSR) to determine what happened:

Table continues on the next page...

Table 18-16. Master receive sequence (continued)

Step	Action
	• If TCF = 1, the reception completed. • If IBB = 0, the bus transitioned from Busy to Idle state. • If IBB = 1, you can ignore check of Arbitration Loss (IBAL = 1). **NOTE:** You can ignore the No Acknowledge condition (RXAK = 1) for receive mode.
h	Read I2C Bus Data I/O Register (IBDR) to determine the data received from the slave.
i	Change IBSR[IBIF] and IBSR[TCF] to 0 by writing 1 to those fields.
j	Repeat steps e through i to read subsequent bytes.

Table 18-17. Slave transmit sequence

Step	Action
a	Use I2C Bus Address Register (IBAD) to define the slave address.
b	Write a 0 to IBCR[MDIS] to enable the I^2C interface system.
c	Examine fields in I2C Bus Status Register (IBSR) as follows: • If IAAS = 1, examine IBSR[SRW]. • If IAAS = 1 and SRW = 1, write a 1 to IBCR[TXRX] to select Transmit mode.
d	Write data to I2C Bus Data I/O Register (IBDR).
e	Wait until IBSR[IBIF] becomes 1.
f	Wait until IBSR[RXAK] becomes 0.
g	Write a 0 to IBSR[IBIF].
h	Repeat steps d through g for the next consecutive data transfers.

Table 18-18. Slave receive sequence

Step	Action
a	Use I2C Bus Address Register (IBAD) to define the slave address.
b	Write a 0 to IBCR[MDIS] to enable the I^2C interface system.
c	Examine fields in I2C Bus Status Register (IBSR) as follows: • If IAAS = 1, examine IBSR[SRW]. • If IAAS = 1 and SRW = 0, write a 0 to IBCR[TXRX] to select Receive mode.
d	Write a 0 to IBSR[IBIF].
e	Perform a dummy read of I2C Bus Data I/O Register (IBDR) to initiate the receive operation.
f	Wait until IBSR[TCF] becomes 1. This proves that the transfer is complete.
g	Wait until IBSR[IBIF] becomes 1.
h	Read the fields in I2C Bus Status Register (IBSR) to determine what happened: • If TCF = 1, the reception completed. • If IBB = 0, the bus transitioned from Busy to Idle state. • If IBB = 1, you can ignore check of Arbitration Loss (IBAL = 1). **NOTE:** You can ignore the No Acknowledge condition (RXAK = 1) for receive mode.
i	Read I2C Bus Data I/O Register (IBDR) to determine the data received from the master.
j	Change IBSR[IBIF] and IBSR[TCF] to 0 by writing 1 to those fields.
k	Repeat steps f through j to read subsequent bytes.

18.7.3.3 Generating STOP

A data transfer ends with a STOP condition generated by the master device. A master transmitter can simply generate a STOP condition after it transmits all the data. Here is an example showing how a master transmitter generates a STOP condition.

```
if (tx_count == 0) or     // check to see if all data bytes have been transmitted
  (bit 0, IBSR == 1) {    // or if no ACK generated
    clear bit 5, IBCR     // generate STOP condition
  }
else {
 IBDR = data_to_transmit  // write byte of data to DATA register
   tx_count --            // decrement counter
 }                        // return from interrupt
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data. The master receiver can do this by programming IBCR[NOACK]=1 before reading the second-to-last byte of data. Before reading the last byte of data, the master receiver must first generate a STOP condition. Here is an example showing how a master receiver generates a STOP condition.

```
rx_count --            // decrease the rx counter
if (rx_count ==1)      // 2nd last byte to be read ?
 bit 3, IBCR = 1       // disable ACK
 if (rx_count == 0)    // last byte to be read ?
 bit 5, IBCR = 0       // generate STOP condition
else
data_received = IBDR   // read RX data and store
```

18.7.3.4 Generating repeated START

At the end of data transfer, if the master still wants to communicate on the bus, it can generate another START signal followed by another slave address without first generating a STOP signal. Here is a program example.

```
bit 2, IBCR = 1             // generate another start (restart)
IBDR == calling_address     // transmit the calling address
```

18.7.3.5 Loss of arbitration

If several masters try to engage the bus simultaneously, only one master wins and the others lose arbitration. The devices that lost arbitration are immediately switched to slave mode by the I^2C module. Their data output to the SDA line is stopped, but SCL is still generated until the end of the byte during which arbitration was lost. An interrupt occurs

at the falling edge of the ninth clock of this transfer with IBSR[IBAL]=1 and IBCR[MSSL]=0. If one master attempts to start transmission while the bus is being engaged by another master, the hardware:

- Inhibits the transmission
- Switches IBCR[MSSL] from 1 to 0 without generating a STOP condition
- Generates an interrupt to the CPU
- Programs IBSR[IBAL]=1 to indicate that the attempt to engage the bus is failed
- Does not program IBSR[TCF]=1 due to the loss of data during arbitration

When considering these cases, your slave service routine should test IBSR[IBAL] first and your software should program IBSR[IBAL]=0 if IBSR[IBAL]=1.

18.7.4 Programming guidelines specific to Slave mode

In the slave interrupt service routine, IBSR[IAAS] should be tested to check if a calling of its own address has just been received. If IBSR[IAAS]=1, software should program IBCR[TXRX]=1 according to IBSR[SRW]. Writing to the IBCR clears IBSR[IAAS] automatically. The only time you will read a value of 1 from IBSR[IAAS] is from the interrupt at the end of the address cycle where an address match occurred. Interrupts resulting from subsequent data transfers will have IAAS cleared.

A data transfer may now be initiated by writing information to IBDR for slave transmits or dummy reading from IBDR in slave receive mode. The slave will drive SCL low between byte transfers. SCL is released when the IBDR is accessed in the required mode.

In slave transmitter routine, IBSR[RXAK] must be tested before transmitting the next byte of data. Setting RXAK means an end-of-data signal from the master receiver, after which it must be switched from transmitter mode to receiver mode by software. A dummy read then releases the SCL line so that the master can generate a STOP signal.

Chapter 19
PCI Express Interface Controller

19.1 The PCI Express controller as implemented on the chip

This section provides details about how the PCI Express controller is integrated into this chip.

The following table describes the PCI Express controller integration into this chip:

Table 19-1. PCI Express controller integration

Controller	Base address	Maximum number of lanes	Supports SR-IOV
PCIe1	340_0000h	4	Yes (SR-IOV is supported only in EP mode)
PCIe1_LUT	348_0000h	—	—
PCIe1_PF0 Controls	0x34C_0000	—	—
PCIe2	350_0000h	1	No
PCIe2_LUT	358_0000h	—	—
PCIe2_PF0 Controls	0x35C_0000	—	—
PCIe3	360_0000h	2	No
PCIe3_LUT	368_0000h	—	—
PCIe3_PF0 Controls	0x36C_0000	—	—

NOTE

Power down for any module can be configured by programming the DEVDISRn register. If any module is powered down, its data access as well as register access are not allowed.

There are extra requirements specifically for PCI Express to address the following two usage scenarios:

- A PCI Express controller is selected by a specific SRDS_PRTCL_Sn option in RCW. However the end product application has no usage of the controller and thus wants to disable it.
- A PCI Express controller is not selected by a specific SRDS_PRTCL_Sn option in RCW. For both the above cases, the unused or unselected PCI Express controller must be disabled by setting the corresponding bit of the DEVDISR3. Once disabled, data access and register access to that PCI Express controller must be avoided to prevent undefined behavior. It is recommended to disable unused or unselected PCI Express controllers in PBI.

19.2 Introduction

The PCI Express interface is compatible with the *PCI Express™ Base Specification, Revision 3.0* (available from http://www.pcisig.com). It is beyond the scope of this manual to document the intricacies of the PCI Express protocol. This chapter describes the PCI Express controller of this device and provides a basic description of the PCI Express protocol. The specific emphasis is directed at how the device implements the PCI Express specification. Designers of systems incorporating PCI Express devices should refer to the specification for a thorough description of PCI Express.

NOTE

Much of the available PCI Express literature refers to a 16-bit quantity as a WORD and a 32-bit quantity as a DWORD. Note that this is inconsistent with the terminology in the rest of this manual where the terms 'word' and 'double word' refer to a 32-bit and 64-bit quantity, respectively. Where necessary to avoid confusion, the precise number of bits or bytes is specified.

19.2.1 Overview

The PCI Express controller connects the internal platform to a serial interface.

As both an initiator and a target device, the PCI Express interface is capable of high-bandwidth data transfer and is designed to support next generation I/O devices. Upon coming out of reset, the PCI Express interface performs link width negotiation and exchanges flow control credits with its link partner. Once link autonegotiation is successful, the controller is in operation.

Internally, the design contains queues to keep track of inbound and outbound transactions. There is control logic that handles buffer management, bus protocol, transaction spawning and tag generation. In addition, there are memory blocks used to store inbound and outbound data.

The PCI Express controller can be configured to operate as either a PCI Express Root Complex (RC) or Endpoint (EP) device. An RC device connects the host CPU/memory subsystem to I/O devices while an EP device typically denotes a peripheral or I/O device. In RC mode, a PCI Express type 1 configuration header is used; in EP mode, a PCI Express type 0 configuration header is used.

This figure shows a high-level block diagram of the PCI Express controller.

Figure 19-1. PCI Express Controller Block Diagram

As an initiator, the PCI Express controller supports memory read and write operations. In addition, configuration and I/O transactions are supported if the PCI Express controller is in RC mode. As a target interface, the PCI Express controller accepts read and write operations to local memory space. As an EP device, the PCI Express controller accepts

configuration transactions to the internal PCI Express configuration registers. Message generation and acceptance are supported in both RC and EP modes. Locked transactions and inbound I/O transactions are not supported.

19.2.1.1 Outbound Transactions

Outbound internal platform transactions to PCI Express are first mapped to a translation window to determine what PCI Express transactions are to be issued. A transaction from the internal platform can become a PCI Express Memory, I/O,Message, or Configuration transaction depending on the window attributes.

A transaction may be broken up into smaller sized transactions depending on the original request size, transaction type, and either the PCI Express Device Control register [MAX_PAYLOAD_SIZE] field for write requests or the PCI Express Device Control register [MAX_READ_SIZE] field for read requests. The controller performs PCI Express ordering rule checking to determine which transaction is to be sent on the PCI Express link.

In general, transactions are serviced in the order that they are received from the internal platform . Only when there is a stalled condition does the controller apply PCI Express ordering rules to outstanding transactions. For posted write transactions, once all data has been received from the internal platform , the data is forwarded to the PCI Express link and the transaction is considered as done. For non-posted write transactions, the controller waits for the completion packets to return before considering the transaction finished. For non-posted read transactions, the controller waits for all completion packets to return and then forwards all data back to the internal platform before terminating the transaction.

Note that after reset or when recovering from a link down condition, external transactions should not be attempted until the link has successfully trained. Software can poll the LTSSM state status to check the status of link training before issuing external requests.

In EP mode, after reset or when recovering from a link down condition, the SoC must not generate any outbound memory or I/O transactions until the remote host has configured the Bus Master Enable bit in the PCI Command register.

19.2.1.2 Inbound Transactions

Inbound PCI Express transactions to internal platform are first mapped to a translation window to determine what internal platform transactions are to be issued.

A transaction may be broken up into smaller sized transactions when sending to the internal platform depending on the original request size, byte enables and starting/ending addresses. The controller performs PCI Express ordering rule checking to determine what transaction is to be sent next to the internal platform.

In general, transactions are serviced in the order that they are received from the PCI Express link. Only when there is a stalled condition does the controller apply PCI Express ordering to outstanding transactions. For posted write transactions, once all data has been received from the PCI Express link, the data is forwarded to the internal platform and the transaction is considered as done. For non-posted read transactions, the controller forwards internal platform data back to the PCI Express link.

Note that the controller splits transactions at the crossing of every 256-byte-aligned boundary when sending data back to the PCI Express link.

19.2.2 Features

The following is a list of features supported by the PCI Express controller:

- Compatible with the *PCI Express™ Base Specification, Revision 3.0*
- Supports Root Complex (RC) and Endpoint (EP) configurations
- 32- and 64-bit PCI Express address support
- 40-bit internal platform address support
- x4, x2, and x1 link support.
- Supports accesses to all PCI Express memory and I/O address spaces (requestor only)
- Supports posting of processor-to-PCI Express and PCI Express-to-memory writes
- Supports strong and relaxed transaction ordering rules
- Enforces outbound PCI Express ordering rules and inbound internal platform priority
- PCI Express configuration registers (type 0 in EP mode, type 1 in RC mode)
- Baseline and advanced error reporting support
- One virtual channel (VC0)
- 256-byte maximum payload size (MAX_PAYLOAD_SIZE)
- Supports 64-bit MSI interrupts
- Credit-based flow control management handled by PCI Express core.
- Supports PCI Express messages and interrupts
- Accepts up to 256-byte transactions from the internal platform
- SR-IOV-only: Supports Functional Level Reset (FLR)
- SR-IOV-only: Supports Alternative Routing ID (ARI)
- SR-IOV-only: Supports MSI-X
    - Eight MSI-X vectors per PF or VF
- Supports Expansion ROM.

19.2.3 Modes of Operation

Several parameters that affect the PCI Express controller modes of operation are determined at power-on reset (POR) by reset configuration word (RCW) fields configured depending on SoC product.

Table 19-2. POR Parameters for PCI Express Controller

RCW Parameter	Description
Host/Agent	Selects between Root Complex (RC) and Endpoint (EP) modes
SerDes Protocol Select SRDS_PRTCL	Determines the link width
SerDes frequency divider for PCI Express	Determines the link speed

19.2.3.1 Root Complex/Endpoint Modes

The PCI Express controller can function as either a Root Complex (RC) or an Endpoint (EP) on the PCI Express link. The host/agent configuration field, RCW[HOST_AGT_PEX], determines the RC/EP mode.

19.2.3.2 Link Width

The initial link width is determined by the SerDes protocol configuration field (RCW[SRDS_PRTCL_Sn]). See Reset Configuration Word (RCW) for more information. The specific configurations are detailed in SerDes Lane Assignments and Multiplexing.

19.2.3.3 Link Speed

The initial link speed is determined by the SerDes frequency divider for PCI Express field (RCW[SRDS_DIV_PEX_Sn]).

See RCW Field Definitions for more information. The specific configurations are detailed in Reference Clocks for SerDes Protocols.

19.2.3.4 EP SR-IOV

Single Root I/O Virtualization (SR-IOV) allows for different software images to share IO resources that exist on an EP device. With SR-IOV supported in EP, different devices or different software tasks can share IO resources such as Gigabit Ethernet controllers with minimum code modifications. Additional benefits include savings in hardware cost for additional and/or dedicated IO resources, space and power consumption.

19.2.3.5 MSI-X (SR-IOV-only)

MSI-X allows the Physical or Virtual functions (as supported by the EP SR-IOV) to send independent message interrupts to the RC device.

19.3 External Signal Descriptions

The PCI Express specification defines the connection between two devices as a link, which can be composed of a single or multiple lanes. Each lane consists of a differential pair for transmitting (TX[n]_P and TX[n]_N) and a differential pair for receiving (RX[n]_P and RX[n]_N) with an embedded data clock.

Table 19-3. PCI Express Interface Signals—Detailed Signal Descriptions

Signal	I/O	Description	
SD_RX[n]_P	I	Receive data, positive. The receive data signals carry PCI Express packet information.	
		State Meaning	Asserted/Negated—Represents data being received from the PCI Express interface.
		Timing	Assertion/Negation—As described in the *PCI Express Base Specification, Revision 3.0*.
SD_RX[n]_N	I	Receive data, negative. The receive data signals carry PCI Express packet information.	
		State Meaning	Asserted/Negated—Represents the inverse of data being received from the PCI Express interface.
		Timing	Assertion/Negation—As described in the *PCI Express Base Specification, Revision 3.0*.
SD_TX[n]_P	O	Transmit data, positive. The transmit data signals carry PCI Express packet information.	
		State Meaning	Asserted/Negated—Represents data being transmitted to the PCI Express interface.
		Timing	Assertion/Negation—As described in the *PCI Express Base Specification, Revision 3.0*.
SD_TX[n]_N	O	Transmit data, negative. The transmit data signals carry PCI Express packet information.	
		State Meaning	Asserted/Negated—Represents the inverse of data being transmitted to the PCI Express interface.
		Timing	Assertion/Negation—As described in the *PCI Express Base Specification, Revision 3.0*.

19.4 Memory map/register overview

19.4.1 PCI Express configuration registers

The PCI Express module supports the same configuration registers in both the internal memory map and in PCI Express configuration space. They differ only in whether they are accessed from an internal initiator or from an external initiator on the PCI Express interface. With the exception of the registers in the PEX module internal configuration space, the configuration registers are specified by the PCI Express specification for every PCI Express device. The registers in the PEX module internal configuration space are used for chip-specific functionality and are not defined by the PCI Express specification.

Table 19-4. PCI Express memory map

	Root Complex or Non-SR-IOV Endpoint		SR-IOV Endpoint		Notes
Register space	**Offset**	**NEXT pointer**	**Offset**	**NEXT pointer**	**Notes**
Type 0 configuration header	0x000	0x40	0x000	0x40	EP use Type 0
Type 1 configuration header	0x000	0x40	N/A	—	RC use Type 1
Power management capability structure	0x040	0x50	0x040	0x50	
MSI message capability structure	0x050	0x70	0x050	0x70	
PCI Express capability structure	0x070	0x00 (NULL)	0x070	0xB0	
MSI-X capability structure	N/A	—	0x0B0	0x00 (NULL)	SR-IOV only
Advanced error reporting capability structure	0x100	0x148	0x100	0x148	
Alternative Routing ID capability structure	N/A	—	0x148	0x158	SR-IOV only
Secondary PCI Express capability structure	0x148	0x000 (NULL)	0x158	0x178	The secondary PCI Express capability structure has a different offset and NEXT pointer depending on whether SR-IOV is supported.
SR-IOV capability structure	N/A	—	0x178	0x000 (NULL)	SR-IOV only
PEX module internal configuration space	0x700	—	0x700	—	
BAR mask registers	0x1000	—	0x1000	—	

19.4.2 SR-IOV PF/VF configuration registers

For an SR-IOV endpoint, each PF/VF will have its own configuration registers as specified by the PCI Express SR-IOV specification. The PF/VF registers are located in a secondary configuration space that is indirectly accessed by first writing the VF information with VF_ACT = 1 to the PEX PFa VF control register (PEX_PFa_VF) at offset 0x4_07F8 (PF0) or 0x5_07F8 (PF1) in the PEX_LUT memory map and then accessing the desired PCI Express register offset.

See PEX_LUT memory map/registers overview for more information. Each PF/VF supports the following capability structures:

PFa/VFb Capability Structure	Offset
Type 0 configuration header	CS2: 0x000
PCI Express capability structure	CS2: 0x070
MSI-X capability structure	CS2: 0x0B0
Alternative Routing ID capability structure	CS2: 0x148

19.4.3 SR-IOV Instance (PEX1) register descriptions

This section describes the registers for the SR-IOV instance of the PEX controller (PEX1); see Non-SR-IOV Instances (PEX2 and PEX3) register descriptions for the descriptions of the registers for the non-SR-IOV instances (PEX2 and PEX3).

19.4.3.1 PCI_Express_Configuration_Registers memory map

PEX1 base address: 340_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	PCI Express Vendor ID Register (Vendor_ID_Register)	16	RO	1957h
2h	PCI Express Device ID Register (Device_ID_Register)	16	RO	80C0h
4h	PCI Express Command Register (Command_Register)	16	RW	0000h
6h	PCI Express Status Register (Status_Register)	16	W1C	0010h
8h	PCI Express Revision ID Register (Revision_ID_Register)	8	RO	10h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
9h	PCI Express Class Code Register (programming interface) (Class_Code_Register_a)	8	RO	00h
Ah	PCI Express Class Code Register (sub class) (Class_Code_Register_b)	8	RO	20h
Bh	PCI Express Class Code Register (base class) (Class_Code_Register_c)	8	RO	0Bh
Ch	PCI Express Cache Line Size Register (Cache_Line_Size_Register)	8	RW	00h
Dh	PCI Express Latency Timer Register (Latency_Timer_Register)	8	RO	00h
Eh	PCI Express Header Type Register (Header_Type_Register)	8	RO	80h
10h	PCI Express Base Address Register 0 (BAR0)	32	RW	0000_0000h
14h	PCI Express Base Address Register 1 (BAR1)	32	RW	0000_0000h
18h	PCI Express Base Address Register 2 (BAR2)	32	RW	0000_000Ch
18h	PCI Express Primary Bus Number Register (Primary_Bus_Number_Register)	8	RW	00h
19h	PCI Express Secondary Bus Number Register (Secondary_Bus_Number_Register)	8	RW	00h
1Ah	PCI Express Subordinate Bus Number Register (Subordinate_Bus_Number_Register)	8	RW	00h
1Ch	PCI Express Base Address Register 3 (BAR3)	32	RW	0000_0000h
1Ch	PCI Express I/O Base Register (IO_Base_Register)	8	RW	01h
1Dh	PCI Express I/O Limit Register (IO_Limit_Register)	8	RW	01h
1Eh	PCI Express Secondary Status Register (Secondary_Status_Register)	16	W1C	0000h
20h	PCI Express Base Address Register 4 (BAR4)	32	RW	0000_000Ch
20h	PCI Express Memory Base Register (Memory_Base_Register)	16	RW	0000h
22h	PCI Express Memory Limit Register (Memory_Limit_Register)	16	RW	0000h
24h	PCI Express Base Address Register 5 (BAR5)	32	RW	0000_0000h
24h	PCI Express Prefetchable Memory Base Register (Prefetchable_Memory_Base_Register)	16	RW	0001h
26h	PCI Express Prefetchable Memory Limit Register (Prefetchable_Memory_Limit_Register)	16	RW	0001h
28h	PCI Express Prefetchable Base Upper 32 Bits Register (Prefetchable_Base_Upper_32_Bits_Register)	32	RW	0000_0000h
2Ch	PCI Express Prefetchable Limit Upper 32 Bits Register (Prefetchable_Limit_Upper_32_Bits_Register)	32	RW	0000_0000h
2Ch	PCI Express Subsystem Vendor ID Register (Subsystem_Vendor_ID_Register)	16	RO	0000h
2Eh	PCI Express Subsystem ID Register (Subsystem_ID_Register)	16	RO	0000h
30h	PCI Express Expansion ROM Base Address Register (EP-Mode) (Expansion_ROM_BAR_Type0)	32	RW	0000_0000h
30h	PCI Express I/O Base Upper 16 Bits Register (IO_Base_Upper_16_Bits_Register)	16	RO	0000h
32h	PCI Express I/O Limit Upper 16 Bits Register (IO_Limit_Upper_16_Bits_Register)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
34h	Capabilities Pointer Register (Capabilities_Pointer_Register)	8	RO	40h
38h	PCI Express Expansion ROM Base Address Register (RC-Mode) (Expansion_ROM_BAR_Type1)	32	RW	0000_0000h
3Ch	PCI Express Interrupt Line Register (Interrupt_Line_Register)	8	RW	FFh
3Dh	PCI Express Interrupt Pin Register (Interrupt_Pin_Register)	8	RO	01h
3Eh	PCI Express Bridge Control Register (Bridge_Control_Register)	16	RW	0000h
3Eh	PCI Express Minimum Grant Register (Minimum_Grant_Register)	8	RO	00h
3Fh	PCI Express Maximum Latency Register (Maximum_Latency_Register)	8	RO	00h
40h	PCI Express Power Management Capability ID Register (Power_Management_Capability_ID_Register)	8	RO	01h
42h	PCI Express Power Management Capabilities Register (Power_Management_Capabilities_Register)	16	RO	7E23h
44h	PCI Express Power Management Status and Control Register (Power_Management_Status_and_Control_Register)	16	RW	0000h
47h	PCI Express Power Management Data Register (Power_Management_Data_Register)	8	RO	00h
50h	PCI Express MSI Message Capability ID Register (MSI_Message_Capability_ID_Register)	8	RO	05h
52h	PCI Express MSI Message Control Register (MSI_Message_Control_Register)	16	RW	0088h
54h	PCI Express MSI Message Address Register (MSI_Message_Address_Register)	32	RW	0000_0000h
58h	PCI Express MSI Message Upper Address Register (MSI_Message_Upper_Address_Register)	32	RW	0000_0000h
5Ch	PCI Express MSI Message Data Register (MSI_Message_Data_Register)	16	RW	0000h
70h	PCI Express Capability ID Register (Capability_ID_Register)	8	RO	10h
72h	PCI Express Capabilities Register (Capabilities_Register)	16	RO	0002h
74h	PCI Express Device Capabilities Register (Device_Capabilities_Register)	32	RO	0000_8001h
78h	PCI Express Device Control Register (Device_Control_Register)	16	RW	2910h
7Ah	PCI Express Device Status Register (Device_Status_Register)	16	W1C	0000h
7Ch	PCI Express Link Capabilities Register (Link_Capabilities_Register)	32	RO	0043_F483h
80h	PCI Express Link Control Register (Link_Control_Register)	16	RW	0008h
82h	PCI Express Link Status Register (Link_Status_Register)	16	RO	1000h
84h	PCI Express Slot Capabilities Register (Slot_Capabilities_Register)	32	RO	0000_0000h
88h	PCI Express Slot Control Register (Slot_Control_Register)	16	RW	03C0h
8Ah	PCI Express Slot Status Register (Slot_Status_Register)	16	W1C	0008h
8Ch	PCI Express Root Control Register (Root_Control_Register)	16	RW	0000h
8Eh	PCI Express Root Capabilities Register (Root_Capabilities_Register)	16	RW	0000h
90h	PCI Express Root Status Register (Root_Status_Register)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
94h	PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)	32	RO	0000_001Fh
98h	PCI Express Device Control 2 Register (Device_Control_2_Register)	16	RW	0000h
9Ch	PCI Express Link Capabilities 2 Register (Link_Capabilities_2_Register)	32	RO	0000_000Eh
A0h	PCI Express Link Control 2 Register (Link_Control_2_Register)	16	RW	0003h
A2h	PCI Express Link Status 2 Register (Link_Status_2_Register)	16	RO	0000h
B0h	PCI Express MSI-X Message Capability ID Register (MSI_X_Message_Capability_ID_Register)	8	RO	11h
B2h	PCI Express MSI-X Message Control Register (MSI_X_Message_Control_Register)	16	RW	0007h
B4h	PCI Express MSI-X Table Offset/BIR Register (MSI_X_Table_Offset_BIR_Register)	32	RW	0000_0001h
B8h	PCI Express MSI-X PBA Offset/BIR Register (MSI_X_PBA_Offset_BIR_Register)	32	RW	0000_1001h
100h	PCI Express Advanced Error Reporting Capability ID Register (Advanced_Error_Reporting_Capability_ID_Register)	16	RO	0001h
104h	PCI Express Uncorrectable Error Status Register (Uncorrectable_Error_Status_Register)	32	W1C	0000_0000h
108h	PCI Express Uncorrectable Error Mask Register (Uncorrectable_Error_Mask_Register)	32	RW	0000_0000h
10Ch	PCI Express Uncorrectable Error Severity Register (Uncorrectable_Error_Severity_Register)	32	RW	0046_2030h
110h	PCI Express Correctable Error Status Register (Correctable_Error_Status_Register)	32	W1C	0000_0000h
114h	PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register)	32	RW	0000_2000h
118h	PCI Express Advanced Error Capabilities and Control Register (Advanced_Error_Capabilities_and_Control_Register)	32	RW	0000_00A0h
11Ch	PCI Express Header Log Register 1 (Header_Log_Register_DWORD1)	32	RO	0000_0000h
120h	PCI Express Header Log Register 2 (Header_Log_Register_DWORD2)	32	RO	0000_0000h
124h	PCI Express Header Log Register 3 (Header_Log_Register_DWORD3)	32	RO	0000_0000h
128h	PCI Express Header Log Register 4 (Header_Log_Register_DWORD4)	32	RO	0000_0000h
12Ch	PCI Express Root Error Command Register (Root_Error_Command_Register)	32	RW	0000_0000h
130h	PCI Express Root Error Status Register (Root_Error_Status_Register)	32	W1C	0000_0000h
134h	PCI Express Correctable Error Source ID Register (Correctable_Error_Source_ID_Register)	16	RO	0000h
136h	PCI Express Error Source ID Register (Error_Source_ID_Register)	16	RO	0000h
148h	ARI Capability Header (ARI_Capability_Header)	32	RO	0001_000Eh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
14Ch	ARI Capability Register (ARI_Capability_Register)	16	RO	0000h
14Eh	ARI Control Register (ARI_Control_Register)	16	RO	0000h
158h	Secondary PCI Express Extended Capability Header (SPCIE_CAP_HEADER_REG)	32	RO	0001_0019h
15Ch	Link Control 3 Register (LINK_CONTROL3_REG)	32	RW	0000_0000h
160h	Lane Error Status Register (LANE_ERR_STATUS_REG)	32	W1C	0000_0000h
164h - 16Ah	Lane Equalization Control Register (LANE0_EQUALIZATION_CONTROL - LANE3_EQUALIZATION_CONTROL)	16	RO	7F7Fh
178h	SR-IOV Extended Capability Header (SR_IOV_Extended_Capability_ID_Register)	32	RO	0000_0010h
17Ch	PCI Express SR-IOV Capabilities Register (SR_IOV_Capabilities_Register)	32	RO	0000_0000h
180h	PCI Express SR-IOV Control Register (SR_IOV_Control_Register)	16	RO	0000h
182h	PCI Express SR-IOV Status Register (SR_IOV_Status_Register)	16	RO	0000h
184h	PCI Express SR-IOV Initial VF Register (SR_IOV_Initial_VF_Register)	16	RO	0010h
186h	PCI Express SR-IOV Total VF Register (SR_IOV_Total_VF_Register)	16	RO	0010h
188h	PCI Express SR-IOV Number VF Register (SR_IOV_Number_VF_Register)	16	RW	0000h
18Ah	PCI Express SR-IOV Function Dependency Link Register (SR_IOV_Function_Dependency_Link_Register)	8	RW	00h
18Ch	PCI Express SR-IOV First VF Offset Register (SR_IOV_First_VF_Offset_Register)	16	RO	0004h
18Eh	PCI Express SR-IOV VF Stride Register (SR_IOV_VF_Stride_Register)	16	RO	0100h
192h	PCI Express SR-IOV Device ID Register (SR_IOV_Device_ID_Register)	16	RO	0000h
194h	PCI Express SR-IOV Supported Page Size Register (SR_IOV_Supported_Page_Size_Register)	32	RO	0000_0553h
198h	PCI Express SR-IOV System Page Size Register (SR_IOV_System_Page_Size_Register)	32	RW	0000_0001h
19Ch	PCI Express SR-IOV VF Base Address Register 0 (VFBAR0)	32	RW	0000_0008h
1A0h	PCI Express SR-IOV VF Base Address Register 1 (VFBAR1)	32	RW	0000_0008h
1A4h	PCI Express SR-IOV VF Base Address Register 2 (VFBAR2)	32	RW	0000_000Ch
1A8h	PCI Express SR-IOV VF Base Address Register 3 (VFBAR3)	32	RW	0000_0000h
1ACh	PCI Express SR-IOV VF Base Address Register 4 (VFBAR4)	32	RW	0000_000Ch
1B0h	PCI Express SR-IOV VF Base Address Register 5 (VFBAR5)	32	RW	0000_0000h
1B4h	PCI Express SR-IOV Migration State Array Offset Register (SR_IOV_Migration_State_Array_Offset_Register)	32	RO	0000_0000h
71Ch	Symbol Timer Register and Filter Mask 1 Register (SYMBOL_TIMER_FILTER_1_OFF)	32	RW	0000_0280h
890h	Gen3 Control Register (GEN3_RELATED_OFF)	32	RW	0080_0001h
8BCh	DBI Read-only Write Enable Register (MISC_CONTROL_1_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8E0h	Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)	32	RW	0000_0000h
8E4h	Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)	32	RW	0000_0000h
8E8h	Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)	32	RU	0000_0000h
900h	iATU Index Register (IATU_VIEWPORT_OFF)	32	RW	0000_0000h
904h	iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_INBOUND_0)	32	RW	0000_0000h
904h	iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)	32	RW	0000_0000h
908h	iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)	32	RW	0000_0000h
908h	iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)	32	RW	0000_0000h
90Ch	iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
90Ch	iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
910h	iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
910h	iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
914h	iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_INBOUND_0)	32	RW	0000_0FFFh
914h	iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0FFFh
918h	iATU Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
918h	iATU Outbound Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
91Ch	iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
91Ch	iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
1010h	Base Address Register 0 Mask (BAR0_MASK)	32	WO	00FF_FFFFh
1014h	Base Address Register 1 Mask (BAR1_MASK)	32	WO	03FF_FFFFh
1018h	Base Address Register 2 Mask (BAR2_MASK)	32	WO	FFFF_FFFFh
101Ch	Base Address Register 3 Mask (BAR3_MASK)	32	WO	0000_00FFh
1020h	Base Address Register 4 Mask (BAR4_MASK)	32	WO	FFFF_FFFFh
1024h	Base Address Register 5 Mask (BAR5_MASK)	32	WO	0000_00FFh
1030h	Expansion ROM Base Address Register Mask (EP mode) (EXP_ROM_BAR_MASK_EP)	32	WO	00FF_FFFFh
1038h	Expansion ROM Base Address Register Mask (RC mode) (EXP_ROM_BAR_MASK_RC)	32	WO	00FF_FFFFh

19.4.3.2 PCI Express Vendor ID Register (Vendor_ID_Register)

19.4.3.2.1 Offset

Register	Offset
Vendor_ID_Register	0h

19.4.3.2.2 Function

The vendor ID register is used to identify the manufacturer of the device.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.3.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Vendor_ID															
W																
Reset	0	0	0	1	1	0	0	1	0	1	0	1	0	1	1	1

19.4.3.2.4 Fields

Field	Function
15-0 Vendor_ID	Vendor ID 0x1957 (NXP)

19.4.3.3 PCI Express Device ID Register (Device_ID_Register)

19.4.3.3.1 Offset

Register	Offset
Device_ID_Register	2h

19.4.3.3.2 Function

The device ID register is used to identify the device.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.3.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Device_ID															
W																
Reset	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

19.4.3.3.4 Fields

Field	Function
15-0 Device_ID	Device ID 1000000011000000b - LS1088A with security 1000000011000001b - LS1088A without security 1000000011000010b - LS1084A with security 1000000011000011b - LS1084A without security 1000000011100000b - LS1048A with security 1000000011100001b - LS1048A without security 1000000011100010b - LS1044A with security 1000000011100011b - LS1044A without security

19.4.3.4 PCI Express Command Register (Command_Register)

19.4.3.4.1 Offset

Register	Offset
Command_Register	4h

19.4.3.4.2 Function

The command register provides control over the ability to generate and respond to PCI Express cycles.

19.4.3.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					Interrupt_Disable	Reserved	SERR	Reserved	Parity_error_response	Reserved			Bus_master	Memory_space	IO_space
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.4.4 Fields

Field	Function
15-11 —	Reserved
10 Interrupt_Disable	Interrupt disable Controls the ability to generate INTx interrupt messages. Any INTx emulation interrupts already asserted by this device must be deasserted when this bit is set. 0b - Enables INTx interrupt messages 1b - Disables INTx interrupt messages
9 —	Reserved
8 SERR	SERR# enable Controls the reporting of fatal and non-fatal errors detected by the device to the Root Complex. **NOTE:** The error control and status bits in the command and status registers control PCI-compatible error reporting. PCI Express advanced error reporting is controlled by the PCI Express device control register described in PCI Express Device Control Register (Device_Control_Register) and the advanced error reporting registers (offsets 100h through 137h). 0b - Disables reporting 1b - Enables reporting
7 —	Reserved
6 Parity_error_response	Parity error response Controls whether this PCI Express controller responds to parity errors. **NOTE:** The error control and status bits in the command and status registers control PCI-compatible error reporting. PCI Express advanced error reporting is controlled by the PCI Express device control register described in PCI Express Device Control Register (Device_Control_Register) and the advanced error reporting registers (offsets 100h through 137h). 0b - Parity errors are ignored and normal operation continues.

Table continues on the next page...

Field	Function
	1b - Parity errors cause the appropriate bit in the PCI Express status register to be set. However, note that errors are reported based on the values set in the PCI Express error enable and detection registers.
5-3 —	Reserved
2 Bus_master	Bus master enable Indicates whether this PCI Express device is configured as a master. EP mode: Clearing this bit prevent the device from issuing any memory or I/O transactions. Because MSI interrupts are effectively memory writes, clearing this bit also disables the ability of the device to issue MSI interrupts. RC mode: Clearing this bit disables the ability of the device to forward memory transactions upstream. This causes any inbound memory transaction to be treated as an unsupported request. 0b - Disables the ability to generate PCI Express accesses 1b - Enables this PCI Express controller to behave as a PCI Express bus master
1 Memory_space	Memory space enable Controls whether this PCI Express device (as a target) responds to memory accesses. EP mode: Clearing this bit prevents the device from accepting any memory transaction. RC mode: This bit is ignored. It does not affect outbound memory transaction 0b - This PCI Express device does not respond to PCI Express memory space accesses. 1b - This PCI Express device responds to PCI Express memory space accesses.
0 IO_space	I/O space enable EP mode: Clearing this bit prevents the device from accepting any IO transaction. Note that this bit is a don't care in EP mode since the device does not support IO transaction. RC mode: This bit is ignored. It does not affect outbound IO transaction. 0b - This PCI Express device (as a target) does not respond to PCI Express I/O space accesses. 1b - This PCI Express device (as a target) does respond to PCI Express I/O space accesses.

19.4.3.5 PCI Express Status Register (Status_Register)

19.4.3.5.1 Offset

Register	Offset
Status_Register	6h

19.4.3.5.2 Function

The status register is used to record status information for PCI Express related events.

19.4.3.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Detected_parity_error	Signaled_system_error	Received_master_abort	Received_target_abort	Signaled_target_abort	Reserved		Master_data_parity_error_detected	Reserved			Capabilities_List	Interrupt_Status	Reserved		
W	W1C	W1C	W1C	W1C	W1C			W1C				W1C	W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

19.4.3.5.4 Fields

Field	Function
15 Detected_parity_error	Detected parity error Set whenever a device receives a poisoned TLP regardless of the state of bit 6 in the command register.[1]
14 Signaled_system_error	Signaled system error Set whenever a device sends a ERR_FATAL or ERR_NONFATAL message and the SERR enable bit in the command register is set.[1]
13 Received_master_abort	Received master abort Set whenever a requestor receives a completion with unsupported request completion status.[1]
12 Received_target_abort	Received target abort Set whenever a device receives a completion with completer abort completion status.[1]
11 Signaled_target_abort	Signaled target abort Set whenever a device completes a request using completer abort completion status.[1]
10-9 —	Reserved
8 Master_data_parity_error_detected	Master data parity error Set by the requestor (primary side for Type1 headers) when either the requestor receives a completion marked poisoned or the requestor poisons a write request. Note that the parity error enable bit (bit 6) in the command register must be set for this bit to be set.[1]

Table continues on the next page...

Field	Function
7-5 —	Reserved
4 Capabilities_List	Capabilities list All PCI Express devices are required to implement the PCI Express capability structure.
3 Interrupt_Status	Interrupt status Set when an INTx interrupt message is pending internally to the device. Note that this bit is associated with INTx messages and not Message Signaled Interrupts.
2-0 —	Reserved

1. The error control and status bits in the command and status registers control PCI-compatible error reporting. PCI Express advanced error reporting is controlled by the PCI Express device control register described in PCI Express Device Control Register (Device_Control_Register) and the advanced error reporting capability structure starting at offset 100h.

19.4.3.6 PCI Express Revision ID Register (Revision_ID_Register)

19.4.3.6.1 Offset

Register	Offset
Revision_ID_Register	8h

19.4.3.6.2 Function

The revision ID register is used to identify the revision of the device.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.3.6.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Revision_ID							
W								
Reset	0	0	0	1	0	0	0	0

19.4.3.6.4 Fields

Field	Function
7-0 Revision_ID	Revision ID Revision specific.

19.4.3.7 PCI Express Class Code Register (programming interface) (Class_Code_Register_a)

19.4.3.7.1 Offset

Register	Offset
Class_Code_Register_a	9h

19.4.3.7.2 Function

The class code register is comprised of three single-byte fields—base class (offset 0x0B), sub-class (offset 0x0A), and programming interface (offset 0x09)—that indicate the basic functionality of the function.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.3.7.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Programming_Interface							
W								
Reset	0	0	0	0	0	0	0	0

19.4.3.7.4 Fields

Field	Function
7-0 Programming_Interface	Programming_Interface

19.4.3.8 PCI Express Class Code Register (sub class) (Class_Code_Register_b)

19.4.3.8.1 Offset

Register	Offset
Class_Code_Register_b	Ah

19.4.3.8.2 Function

The class code register is comprised of three single-byte fields—base class (offset 0x0B), sub-class (offset 0x0A), and programming interface (offset 0x09)—that indicate the basic functionality of the function.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.3.8.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Sub_Class							
W								
Reset	0	0	1	0	0	0	0	0

19.4.3.8.4 Fields

Field	Function
7-0 Sub_Class	Sub-Class

19.4.3.9 PCI Express Class Code Register (base class) (Class_Code_Register_c)

19.4.3.9.1 Offset

Register	Offset
Class_Code_Register_c	Bh

19.4.3.9.2 Function

The class code register is comprised of three single-byte fields—base class (offset 0x0B), sub-class (offset 0x0A), and programming interface (offset 0x09)—that indicate the basic functionality of the function.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.3.9.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Base_Class							
W								
Reset	0	0	0	0	1	0	1	1

19.4.3.9.4 Fields

Field	Function
7-0 Base_Class	Base Class 0x0B-Processor

19.4.3.10 PCI Express Cache Line Size Register (Cache_Line_Size_Register)

19.4.3.10.1 Offset

Register	Offset
Cache_Line_Size_Register	Ch

19.4.3.10.2 Function

The cache line size register is provided for legacy compatibility purposes (PCI 2.3); it is not used for PCI Express device functionality.

19.4.3.10.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Cache_Line_Size							
Reset	0	0	0	0	0	0	0	0

19.4.3.10.4 Fields

Field	Function
7-0 Cache_Line_Size	Cache Line Size Represents the cache line size of the processor in terms of 32-bit words (8 32-bit words = 32 bytes). Note that for PCI Express operation this register is ignored.

19.4.3.11 PCI Express Latency Timer Register (Latency_Timer_Register)

19.4.3.11.1 Offset

Register	Offset
Latency_Timer_Register	Dh

19.4.3.11.2 Function

The latency timer register is provided for legacy compatibility purposes (PCI 2.3); it is not used for PCI Express device functionality.

19.4.3.11.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Latency_Timer							
W								
Reset	0	0	0	0	0	0	0	0

19.4.3.11.4 Fields

Field	Function
7-0 Latency_Timer	Latency_Timer Note that for PCI Express operation this register is ignored.

19.4.3.12 PCI Express Header Type Register (Header_Type_Register)

19.4.3.12.1 Offset

Register	Offset
Header_Type_Register	Eh

19.4.3.12.2 Function

The PCI Express header type register is used to identify the layout of the PCI compatible header. The reset value depends on the RC/EP mode configured at power-on reset.

19.4.3.12.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Multifunction	Header_Layout						
W								
Reset	1	0[1]	0	0	0	0	0	0

1. EP=00h; RC=01h

19.4.3.12.4 Fields

Field	Function
7 Multifunction	Multifunction Identifies whether a device supports multiple functions 0b - Single function device 1b - Multiple function device
6-0 Header_Layout	Header Layout All other encodings reserved. 0000000b - Endpoint - Type 0 layout. 0000001b - Root Complex - Type 1 layout.

19.4.3.13 PCI Express Base Address Register 0 (BAR0)

19.4.3.13.1 Offset

Register	Offset
BAR0	10h

19.4.3.13.2 Function

The PCI Express base address registers (BARs) point to the beginning of distinct address ranges which the device should claim.

19.4.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.13.4 Fields

Field	Function
31-12 ADDRESS	Base address Indicates the base address of the inbound memory window 0. The default size is 16 MB.
11-4 —	Reserved
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.3.14 PCI Express Base Address Register 1 (BAR1)

19.4.3.14.1 Offset

Register	Offset
BAR1	14h

19.4.3.14.2 Function

This register is present only in the Type 0 Header (EP mode).

Base address register 1 at offset 0x14 is used to define the inbound memory window in the 32-bit memory space.

19.4.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.14.4 Fields

Field	Function
31-12 ADDRESS	Base address Indicates the base address where the inbound memory window 1 begins. The default size of this window is 64 MB.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.3.15 PCI Express Base Address Register 2 (BAR2)

19.4.3.15.1 Offset

Register	Offset
BAR2	18h

19.4.3.15.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 2 (BAR2) and base address register 3 (BAR3) define a 64-bit inbound memory window. BAR2 defines the lower portion of the memory window; BAR3 defines the upper portion of the memory window.

19.4.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

19.4.3.15.4 Fields

Field	Function
31-12 ADDRESS	Base address (lower portion) Indicates the lower portion of the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space. 10b - Locate anywhere in 64-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.3.16 PCI Express Primary Bus Number Register (Primary_Bus_Number_Register)

19.4.3.16.1 Offset

Register	Offset
Primary_Bus_Number_Register	18h

19.4.3.16.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.16.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Primary_Bus_Number							
Reset	0	0	0	0	0	0	0	0

19.4.3.16.4 Fields

Field	Function
7-0 Primary_Bus_Number	Primary Bus Number Bus that is connected to the upstream interface. Note that this register is programmed during system enumeration; in RC mode this register should remain 0x00.

19.4.3.17 PCI Express Secondary Bus Number Register (Secondary_Bus_Number_Register)

19.4.3.17.1 Offset

Register	Offset
Secondary_Bus_Number_Register	19h

19.4.3.17.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.17.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Secondary_Bus_Number							
Reset	0	0	0	0	0	0	0	0

19.4.3.17.4 Fields

Field	Function
7-0 Secondary_Bus_Number	Secondary Bus Number Bus that is directly connected to the downstream interface. Note that this register is programmed during system enumeration; in RC mode, this register is typically programmed to 0x01.

19.4.3.18 PCI Express Subordinate Bus Number Register (Subordinate_Bus_Number_Register)

19.4.3.18.1 Offset

Register	Offset
Subordinate_Bus_Number_Register	1Ah

19.4.3.18.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.18.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Subordinate_Bus_Number							
Reset	0	0	0	0	0	0	0	0

19.4.3.18.4 Fields

Field	Function
7-0 Subordinate_Bus_Number	Subordinate Bus Number Highest bus number that is on the downstream interface.

19.4.3.19 PCI Express Base Address Register 3 (BAR3)

19.4.3.19.1 Offset

Register	Offset
BAR3	1Ch

19.4.3.19.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 2 (BAR2) and base address register 3 (BAR3) define a 64-bit inbound memory window. BAR2 defines the lower portion of the memory window; BAR3 defines the upper portion of the memory window.

19.4.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.19.4 Fields

Field	Function
31-0 ADDRESS	Base address (upper portion) Indicates the upper portion of the base address where the inbound memory window begins. If no access to local memory is to be permitted by external requesters, then all bits are programmed.

19.4.3.20 PCI Express I/O Base Register (IO_Base_Register)

19.4.3.20.1 Offset

Register	Offset
IO_Base_Register	1Ch

19.4.3.20.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.3.20.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	IO_Start_Address				Address_Decode_Type			
W								
Reset	0	0	0	0	0	0	0	1

19.4.3.20.4 Fields

Field	Function
7-4	I/O Start Address

Table continues on the next page...

Field	Function
IO_Start_Address	Specifies bits 15:12 of the I/O space start address
3-0 Address_Decode_Type	Address Decode Type Specifies the number of I/O address bits. All other settings are reserved. 0000b - 16-bit I/O address decode 0001b - 32-bit I/O address decode

19.4.3.21 PCI Express I/O Limit Register (IO_Limit_Register)

19.4.3.21.1 Offset

Register	Offset
IO_Limit_Register	1Dh

19.4.3.21.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.3.21.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	IO_Limit_Address				Address_Decode_Type			
W								
Reset	0	0	0	0	0	0	0	1

19.4.3.21.4 Fields

Field	Function
7-4 IO_Limit_Address	I/O Limit Address Specifies bits 15:12 of the I/O space ending address
3-0 Address_Decode_Type	Address Decode Type Specifies the number of I/O address bits. All other settings are reserved. 0000b - 16-bit I/O address decode 0001b - 32-bit I/O address decode

19.4.3.22 PCI Express Secondary Status Register (Secondary_Status_Register)

19.4.3.22.1 Offset

Register	Offset
Secondary_Status_Register	1Eh

19.4.3.22.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.22.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DPE	SSE	RMA	RTA	STA	Reserved		MDPE	Reserved							
W	W1C	W1C	W1C	W1C	W1C			W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.22.4 Fields

Field	Function
15	Detected parity error

Table continues on the next page...

Field	Function
DPE	This bit is set whenever the secondary side receives a poisoned TLP regardless of the state of the parity error response bit.
14 SSE	Signaled system error This bit is set when a device sends a ERR_FATAL or ERR_NONFATAL message, provided the SERR enable bit in the command register is set to enable reporting.
13 RMA	Received master abort This bit is set when the secondary side receives an unsupported request (UR) completion.
12 RTA	Received target abort This bit is set when the secondary side receives a completer abort (CA) completion.
11 STA	Signaled target abort This bit is set when the secondary side issues a CA completion.
10-9 —	Reserved
8 MDPE	Master data parity error This bit is set when the parity error response bit is set and the secondary side requester receives a poisoned completion or poisons a write request. If the parity error response bit is cleared, this bit is never set.
7-0 —	Reserved

19.4.3.23 PCI Express Base Address Register 4 (BAR4)

19.4.3.23.1 Offset

Register	Offset
BAR4	20h

19.4.3.23.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 4 (BAR4) and base address register 5 (BAR5) define a 64-bit inbound memory window. BAR4 defines the lower portion of the memory window; BAR5 defines the upper portion of the memory window.

19.4.3.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

19.4.3.23.4 Fields

Field	Function
31-12 ADDRESS	Base address (lower portion) Indicates the lower portion of the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space. 10b - Locate anywhere in 64-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.3.24 PCI Express Memory Base Register (Memory_Base_Register)

19.4.3.24.1 Offset

Register	Offset
Memory_Base_Register	20h

19.4.3.24.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.24.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Memory_Base												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.24.4 Fields

Field	Function
15-4 Memory_Base	Memory base address Specifies bits 31:20 of the non-prefetchable memory space start address. Typically used for specifying memory-mapped I/O space. **NOTE:** Inbound posted transactions hitting into the mem base/limit range are ignored; inbound non-posted transactions hitting into the mem base/limit range results in an unsupported request response.
3-0 —	Reserved

19.4.3.25 PCI Express Memory Limit Register (Memory_Limit_Register)

19.4.3.25.1 Offset

Register	Offset
Memory_Limit_Register	22h

19.4.3.25.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.25.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Memory_Limit												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.25.4 Fields

Field	Function
15-4 Memory_Limit	Memory limit address Specifies bits 31:20 of the non-prefetchable memory space ending address. Typically used for specifying memory-mapped I/O space. **NOTE:** Inbound posted transactions hitting into the mem base/limit range are ignored; inbound non-posted transactions hitting into the mem base/limit range results in unsupported request response.
3-0 —	Reserved

19.4.3.26 PCI Express Base Address Register 5 (BAR5)

19.4.3.26.1 Offset

Register	Offset
BAR5	24h

19.4.3.26.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 4 (BAR4) and base address register 5 (BAR5) define a 64-bit inbound memory window. BAR4 defines the lower portion of the memory window; BAR5 defines the upper portion of the memory window.

19.4.3.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.26.4 Fields

Field	Function
31-0 ADDRESS	Base address (upper portion) Indicates the upper portion of the base address where the inbound memory window begins. If no access to local memory is to be permitted by external requestors, then all bits are programmed.

19.4.3.27 PCI Express Prefetchable Memory Base Register (Prefetchable_Memory_Base_Register)

19.4.3.27.1 Offset

Register	Offset
Prefetchable_Memory_Base_Register	24h

19.4.3.27.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.27.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PF_Memory_Base												Address_Decode_Type			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.27.4 Fields

Field	Function
15-4 PF_Memory_Base	Prefetchable memory base address Specifies bits 31:20 of the prefetchable memory space start address.
3-0 Address_Decode_Type	Address Decode Type Specifies the number of prefetchable memory address bits. All other settings reserved. 0000b - 32-bit memory address decode 0001b - 64-bit memory address decode

19.4.3.28 PCI Express Prefetchable Memory Limit Register (Prefetchable_Memory_Limit_Register)

19.4.3.28.1 Offset

Register	Offset
Prefetchable_Memory_Limit_Register	26h

19.4.3.28.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.28.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PF_Memory_Limit												Address_Decode_Type			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.28.4 Fields

Field	Function
15-4 PF_Memory_Limit	Prefetchable memory limit address Specifies bits 31:20 of the prefetchable memory space ending address.
3-0 Address_Decode_Type	Address decode type Specifies the number of prefetchable memory address bits. All other settings reserved. 0000b - 32-bit memory address decode 0001b - 64-bit memory address decode

19.4.3.29 PCI Express Prefetchable Base Upper 32 Bits Register (Prefetchable_Base_Upper_32_Bits_Register)

19.4.3.29.1 Offset

Register	Offset
Prefetchable_Base_Upper_32_Bits_Register	28h

19.4.3.29.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PF_Base_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PF_Base_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.29.4 Fields

Field	Function
31-0 PF_Base_Upper_32_Bits	Prefetchable memory base address (upper portion) Specifies bits 64:32 of the prefetchable memory space start address when the address decode type field in the prefetchable memory base register is 0x01.

19.4.3.30 PCI Express Prefetchable Limit Upper 32 Bits Register (Prefetchable_Limit_Upper_32_Bits_Register)

19.4.3.30.1 Offset

Register	Offset
Prefetchable_Limit_Upper_32_Bits_Register	2Ch

19.4.3.30.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PF_Limit_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PF_Limit_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.30.4 Fields

Field	Function
31-0 PF_Limit_Upper_32_Bits	Prefetchable memory limit address (upper portion) Specifies bits 64-32 of the prefetchable memory space ending address when the address decode type field in the prefetchable memory limit register is 0x01.

19.4.3.31 PCI Express Subsystem Vendor ID Register (Subsystem_Vendor_ID_Register)

19.4.3.31.1 Offset

Register	Offset
Subsystem_Vendor_ID_Register	2Ch

19.4.3.31.2 Function

This register is present only in the Type 0 Header (EP mode).

The PCI Express subsystem vendor ID register is used to identify the subsystem.

19.4.3.31.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Subsystem_Vendor_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.31.4 Fields

Field	Function
15-0 Subsystem_Vendor_ID	Subsystem Vendor ID This register is present only in the Type 0 Header (EP mode). The PCI Express subsystem vendor ID register is used to identify the subsystem vendor.

19.4.3.32 PCI Express Subsystem ID Register (Subsystem_ID_Register)

19.4.3.32.1 Offset

Register	Offset
Subsystem_ID_Register	2Eh

19.4.3.32.2 Function

This register is present only in the Type 0 Header (EP mode).

The PCI Express subsystem ID register is used to identify the subsystem.

19.4.3.32.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Subsystem_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.32.4 Fields

Field	Function
15-0 Subsystem_ID	Subsystem ID

19.4.3.33 PCI Express Expansion ROM Base Address Register (EP-Mode) (Expansion_ROM_BAR_Type0)

19.4.3.33.1 Offset

Register	Offset
Expansion_ROM_BAR_Type0	30h

19.4.3.33.2 Function

The Expansion ROM Base Address register is located at offset 0x30 in the Type 0 Header (EP mode); it is located at offset 0x38 in the Type 1 Header (RC mode).

19.4.3.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ROM_Base_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROM_Base_Address					Reserved										ROMBAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.33.4 Fields

Field	Function
31-11 ROM_Base_Address	Expansion ROM base address Specifies bits 31:11 of the non-prefetchable expansion ROM space start address. Typically used for specifying memory-mapped I/O space. The default size is 16M.
10-1 —	Reserved
0 ROMBAR_EN	Expansion ROM enable This bit controls whether or not the device accepts accesses to its expansion ROM 0b - The expansion ROM address space is disabled. 1b - Address decoding is enabled.

19.4.3.34 PCI Express I/O Base Upper 16 Bits Register (IO_Base_Upper_16_Bits_Register)

19.4.3.34.1 Offset

Register	Offset
IO_Base_Upper_16_Bits_Register	30h

19.4.3.34.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.3.34.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IO_Base_Upper_16_Bits															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.34.4 Fields

Field	Function
15-0 IO_Base_Upper_16_Bits	I/O base address (upper portion) Specifies bits 31-16 of the I/O space start address when the address decode type field in the I/O base register is 0x01.

19.4.3.35 PCI Express I/O Limit Upper 16 Bits Register (IO_Limit_Upper_16_Bits_Register)

19.4.3.35.1 Offset

Register	Offset
IO_Limit_Upper_16_Bits_Register	32h

19.4.3.35.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.3.35.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IO_Limit_Upper_16_Bits															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.35.4 Fields

Field	Function
15-0 IO_Limit_Upper_16_Bits	I/O limit address (upper portion) Specifies bits 31-16 of the I/O space ending address when the address decode type field in the I/O limit register is 0x01.

19.4.3.36 Capabilities Pointer Register (Capabilities_Pointer_Register)

19.4.3.36.1 Offset

Register	Offset
Capabilities_Pointer_Register	34h

19.4.3.36.2 Function

The capabilities pointer identifies additional functionality supported by the device.

19.4.3.36.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Capabilities_Pointer							
W								
Reset	0	1	0	0	0	0	0	0

19.4.3.36.4 Fields

Field	Function
7-0 Capabilities_Pointer	Capabilities Pointer The capabilities pointer provides the offset for additional PCI-compatible registers above the common 64-byte header.

19.4.3.37 PCI Express Expansion ROM Base Address Register (RC-Mode) (Expansion_ROM_BAR_Type1)

19.4.3.37.1 Offset

Register	Offset
Expansion_ROM_BAR_Type1	38h

19.4.3.37.2 Function

The Expansion ROM Base Address register is located at offset 0x38 in the Type 1 Header (RC mode); it is located at offset 0x30 in the Type 0 Header (EP mode).

19.4.3.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ROM_Base_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROM_Base_Address					Reserved										ROMBAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.37.4 Fields

Field	Function
31-11 ROM_Base_Address	Expansion ROM base address Specifies bits 31:11 of the non-prefetchable expansion ROM space start address. Typically used for specifying memory-mapped I/O space. The default size is 16M.
10-1 —	Reserved
0 ROMBAR_EN	Expansion ROM enable This bit controls whether or not the device accepts accesses to its expansion ROM 0b - The expansion ROM address space is disabled. 1b - Address decoding is enabled.

19.4.3.38 PCI Express Interrupt Line Register (Interrupt_Line_Register)

19.4.3.38.1 Offset

Register	Offset
Interrupt_Line_Register	3Ch

19.4.3.38.2 Function

The interrupt line register is used by device drivers and OS software to communicate interrupt line routing information. Values in this register are programmed by system software and are system specific.

19.4.3.38.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Interrupt_Line							
Reset	1	1	1	1	1	1	1	1

19.4.3.38.4 Fields

Field	Function
7-0 Interrupt_Line	Interrupt line Used to communicate interrupt line routing information.

19.4.3.39 PCI Express Interrupt Pin Register (Interrupt_Pin_Register)

19.4.3.39.1 Offset

Register	Offset
Interrupt_Pin_Register	3Dh

19.4.3.39.2 Function

The interrupt pin register identifies the legacy interrupt (INTx) messages the device (or function) uses.

19.4.3.39.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Interrupt_pin							
W								
Reset	0	0	0	0	0	0	0	1

19.4.3.39.4 Fields

Field	Function
7-0 Interrupt_pin	Interrupt pin Legacy INTx message used by this device. All other settings reserved. 00000000b - This device does not use legacy interrupt (INTx) messages. 00000001b - INTA 00000010b - INTB 00000011b - INTC 00000100b - INTD

19.4.3.40 PCI Express Bridge Control Register (Bridge_Control_Register)

19.4.3.40.1 Offset

Register	Offset
Bridge_Control_Register	3Eh

19.4.3.40.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.3.40.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									Scnd_RST	Reserved		VGA_EN	ISA_EN	SERR_EN	PER
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.40.4 Fields

Field	Function
15-7 —	Reserved
6 Scnd_RST	Secondary bus reset
5-4 —	Reserved
3 VGA_EN	VGA enable
2 ISA_EN	ISA enable
1 SERR_EN	SERR enable This bit controls the propagation of ERR_COR, ERR_NONFATAL, and ERR_FATAL responses received on the secondary side.
0 PER	Parity error response This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register.

19.4.3.41 PCI Express Minimum Grant Register (Minimum_Grant_Register)

19.4.3.41.1 Offset

Register	Offset
Minimum_Grant_Register	3Eh

19.4.3.41.2 Function

This register is present only in the Type 0 Header (EP mode).

This register does not apply to PCI Express. It is present for legacy purposes.

19.4.3.41.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MIN_GNT							
W								
Reset	0	0	0	0	0	0	0	0

19.4.3.41.4 Fields

Field	Function
7-0 MIN_GNT	Minimum grant Does not apply for PCI Express.

19.4.3.42 PCI Express Maximum Latency Register (Maximum_Latency_Register)

19.4.3.42.1 Offset

Register	Offset
Maximum_Latency_Register	3Fh

19.4.3.42.2 Function

This register is present only in the Type 0 Header (EP mode).

This register does not apply to PCI Express. It is present for legacy purposes.

19.4.3.42.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MAX_LAT							
W								
Reset	0	0	0	0	0	0	0	0

19.4.3.42.4 Fields

Field	Function
7-0 MAX_LAT	Maximum latency Does not apply for PCI Express.

19.4.3.43 PCI Express Power Management Capability ID Register (Power_Management_Capability_ID_Register)

19.4.3.43.1 Offset

Register	Offset
Power_Management_Capability_ID_Register	40h

19.4.3.43.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Power_Mgmt_Capability_ID							
W								
Reset	0	0	0	0	0	0	0	1

19.4.3.43.3 Fields

Field	Function
7-0 Power_Mgmt_Capability_ID	CAP_ID Power Management = 0x01

19.4.3.44 PCI Express Power Management Capabilities Register (Power_Management_Capabilities_Register)

19.4.3.44.1 Offset

Register	Offset
Power_Management_Capabilities_Register	42h

19.4.3.44.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PME_Support					D2	D1	AUX_Curr			DSI	Reserved	PME_CLK	Version		
W																
Reset	0	1	1	1	1	1	1	0	0	0	1	0	0	0	1	1

19.4.3.44.3 Fields

Field	Function
15-11 PME_Support	PME support Indicates the power states that this device supports
10 D2	D2 support
9 D1	D1 support
8-6 AUX_Curr	AUX Current
5 DSI	Device Specific Initialization
4 —	Reserved
3 PME_CLK	PME clock Does not apply to PCI Express.
2-0 Version	Version

19.4.3.45 PCI Express Power Management Status and Control Register (Power_Management_Status_and_Control_Register)

19.4.3.45.1 Offset

Register	Offset
Power_Management_Status_and_Control_Register	44h

19.4.3.45.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PME_STAT	Data_Scale		Data_Select				PME_EN	Reserved						Power_State	
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.45.3 Fields

Field	Function
15 PME_STAT	PME Status
14-13 Data_Scale	Data Scale Obtained directly from the PCI Express base specification.
12-9 Data_Select	Data Select Obtained directly from the PCI Express base specification.
8 PME_EN	PME_En PME Enable. **NOTE:** Bitfield access is sticky.
7-2 —	Reserved
1-0 Power_State	Power State Indicates the current power state of the function. 00b - D0 01b - D1

Field	Function
	10b - D2 11b - D3

19.4.3.46 PCI Express Power Management Data Register (Power_Management_Data_Register)

19.4.3.46.1 Offset

Register	Offset
Power_Management_Data_Register	47h

19.4.3.46.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Data							
W								
Reset	0	0	0	0	0	0	0	0

19.4.3.46.3 Fields

Field	Function
7-0 Data	Data Obtained from the PCI Express base specification.

19.4.3.47 PCI Express MSI Message Capability ID Register (MSI_Message_Capability_ID_Register)

19.4.3.47.1 Offset

Register	Offset
MSI_Message_Capability_ID_Register	50h

19.4.3.47.2 Function

This register is supported only for EP mode.

19.4.3.47.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MSI_Message_Capability_ID							
W								
Reset	0	0	0	0	0	1	0	1

19.4.3.47.4 Fields

Field	Function
7-0 MSI_Message_Capability_ID	CAP_ID MSI Message = 0x05

19.4.3.48 PCI Express MSI Message Control Register (MSI_Message_Control_Register)

19.4.3.48.1 Offset

Register	Offset
MSI_Message_Control_Register	52h

19.4.3.48.2 Function

This register is supported only for EP mode.

19.4.3.48.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AC_64	MME			MMC			MSIE
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

19.4.3.48.4 Fields

Field	Function
15-8 —	Reserved
7 AC_64	64-bit address capable
6-4 MME	Multiple message enable
3-1 MMC	Multiple message capable
0 MSIE	MSI enable

19.4.3.49 PCI Express MSI Message Address Register (MSI_Message_Address_Register)

19.4.3.49.1 Offset

Register	Offset
MSI_Message_Address_Register	54h

19.4.3.49.2 Function

This register is supported only for EP mode.

19.4.3.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Message_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Message_Address														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.49.4 Fields

Field	Function
31-2 Message_Address	Message Address System-specified message address
1-0 —	Reserved. Always returns 00 on reads; write operations have no effect.

19.4.3.50 PCI Express MSI Message Upper Address Register (MSI_Message_Upper_Address_Register)

19.4.3.50.1 Offset

Register	Offset
MSI_Message_Upper_Address_Register	58h

19.4.3.50.2 Function

This register is supported only for EP mode.

19.4.3.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Message_Upper_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Message_Upper_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.50.4 Fields

Field	Function
31-0 Message_Upper_Address	Message Address (upper portion) System-specified message upper address

19.4.3.51 PCI Express MSI Message Data Register (MSI_Message_Data_Register)

19.4.3.51.1 Offset

Register	Offset
MSI_Message_Data_Register	5Ch

19.4.3.51.2 Function

This register is supported only for EP mode.

19.4.3.51.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Message_Data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.51.4 Fields

Field	Function
15-0 Message_Data	Data System-specified message data.

19.4.3.52 PCI Express Capability ID Register (Capability_ID_Register)

19.4.3.52.1 Offset

Register	Offset
Capability_ID_Register	70h

19.4.3.52.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PCI_Express_Capability_ID							
W								
Reset	0	0	0	1	0	0	0	0

19.4.3.52.3 Fields

Field	Function
7-0 PCI_Express_Capability_ID	Capability ID PCI Express = 0x10

19.4.3.53 PCI Express Capabilities Register (Capabilities_Register)

19.4.3.53.1 Offset

Register	Offset
Capabilities_Register	72h

19.4.3.53.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Interrupt_Message_Number					Slot	Device_Port_Type				Capability_Version			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

19.4.3.53.3 Fields

Field	Function
15-14 —	Reserved
13-9 Interrupt_Message_Number	Interrupt Message Number

Table continues on the next page...

Field	Function
	If this function is allocated more than one MSI interrupt number, then this register is required to contain the offset between the base Message Data and the MSI Message that is generated when any of the status bits in either the Slot Status register or the Root Port Status register, of this capability structure, are set.
8 Slot	Slot Implemented (RC mode only)
7-4 Device_Port_Type	Device/Port Type 0000b - (EP mode) 0100b - (RC mode)
3-0 Capability_Version	Capability Version Indicates the defined PCI Express capability structure version number.

19.4.3.54 PCI Express Device Capabilities Register (Device_Capabilities_Register)

19.4.3.54.1 Offset

Register	Offset
Device_Capabilities_Register	74h

19.4.3.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			FLRC	CSPLS		CSPLV								Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RBER	Reserved			EP_L1_LAT			EP_L0s_LAT			ET	PHAN_FCT		MAX_PL_SIZE_SUP		
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.54.3 Fields

Field	Function
31-29 —	Reserved
28 FLRC	Function Level Reset Capability For RC mode the reset value of this bit is 0 and for SR-IOV capable EP mode, the reset value for this bit is 1.
27-26 CSPLS	Captured Slot Power Limit Scale
25-18 CSPLV	Captured Slot Power Limit Value For RC mode, the reset value for this field is 00h; for EP mode, the value is determined by received Set_Slot_Power_Limit messages.
17-16 —	Reserved
15 RBER	Role-Based Error Reporting
14-12 —	Reserved. System software must ignore the value read from these bits. System software is permitted to write any value to these bits.
11-9 EP_L1_LAT	Endpoint L1 Acceptable Latency
8-6 EP_L0s_LAT	Endpoint L0s Acceptable Latency
5 ET	Extended Tag Field Supported
4-3 PHAN_FCT	Phantom Functions Supported
2-0 MAX_PL_SIZE_SUP	Max_Payload_Size Supported Maximum payload size supported. 001 = 256-bytes

19.4.3.55 PCI Express Device Control Register (Device_Control_Register)

19.4.3.55.1 Offset

Register	Offset
Device_Control_Register	78h

19.4.3.55.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IFLR	MAX_READ_SIZE			ENS	APE	PFE	ETE	MAX_PAYLOAD_SIZE			RO	URR	FER	NFER	CER
Reset	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0

19.4.3.55.3 Fields

Field	Function
15 IFLR	Initiate Function Level Reset
14-12 MAX_READ_SIZE	Maximum_Read_Request_Size
11 ENS	Enable No Snoop
10 APE	AUX Power PM Enable
9 PFE	Phantom Functions Enable
8 ETE	Extended Tag Field Enable
7-5 MAX_PAYLOAD_SIZE	Max_Payload_Size Maximum payload size
4 RO	Enable Relaxed Ordering
3 URR	Unsupported Request Reporting Enable
2 FER	Fatal Error Reporting Enable
1 NFER	Non-Fatal Error Reporting Enable

Table continues on the next page...

Field	Function
0 CER	Correctable Error Reporting Enable

19.4.3.56 PCI Express Device Status Register (Device_Status_Register)

19.4.3.56.1 Offset

Register	Offset
Device_Status_Register	7Ah

19.4.3.56.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										TP	APD	URD	FE D	NFE D	CED
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.56.3 Fields

Field	Function
15-6 —	Reserved
5 TP	Transactions Pending
4 APD	AUX Power Detected Note that this bit is always zero for VF.
3 URD	Unsupported Request Detected
2 FED	Fatal Error Detected
1 NFED	Non-Fatal Error Detected

Table continues on the next page...

Field	Function
0 CED	Correctable Error Detected

19.4.3.57 PCI Express Link Capabilities Register (Link_Capabilities_Register)

19.4.3.57.1 Offset

Register	Offset
Link_Capabilities_Register	7Ch

19.4.3.57.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Port_Number								Reserved	AOC	LBWN	DLLARC	SD_ERR_RPT_CAP	CPM	L1_EX_LAT	
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	L1_EX_LAT	L0s_EX_LAT			ASPM		MAX_LINK_W						MAX_LINK_SP			
W																
Reset	1	1	1	1	0	1	0	0	1	0	0	0	0	0	1	1

19.4.3.57.3 Fields

Field	Function
31-24	Port Number

Table continues on the next page...

Field	Function
Port_Number	This field indicates the PCI Express Port number for the given PCI Express Link
23 —	Reserved
22 AOC	ASPM Optionality Compliance Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests.
21 LBWN	Link Bandwidth Notification Capability In EP-mode, this bit is hardwired to 0. In RC-mode it is hardwired to 1.
20 DLLARC	Data Link Layer Active Reporting Capable Set to 1 when in RC. Set to 0 when in EP.
19 SD_ERR_RPT_CAP	Surprise Down Error Reporting Capable
18 CPM	Clock Power Management
17-15 L1_EX_LAT	L1 Exit Latency
14-12 L0s_EX_LAT	L0s Exit Latency
11-10 ASPM	Active State Power Management (ASPM) Support
9-4 MAX_LINK_W	Maximum Link Width
3-0 MAX_LINK_SP	Maximum Link Speed 0001b - 2.5 GT/s link 0010b - 5.0 GT/s 0011b - 8.0 GT/s

19.4.3.58 PCI Express Link Control Register (Link_Control_Register)

19.4.3.58.1 Offset

Register	Offset
Link_Control_Register	80h

19.4.3.58.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	Reserved				LABIE	LBMIE	HW_AUTO_WIDTH_DIS	ECPM	EXT_SYNC	CCC	RL	LD	RCB	Reserved	ASPM_CTL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

19.4.3.58.3 Fields

Field	Function
15-12 —	Reserved
11 LABIE	Link Autonomous Bandwidth Interrupt Enable
10 LBMIE	Link Bandwidth Management Interrupt Enable
9 HW_AUTO_WIDTH_DIS	Hardware Autonomous Width Disable
8 ECPM	Enable Clock Power Management
7 EXT_SYNC	Extended Synch
6 CCC	Common Clock Configuration
5 RL	Retrain Link (Reserved for EP devices). In RC mode, setting this bit initiates link retraining by directing the Physical Layer LTSSM to the Recovery state; reads of this bit always return 0.
4 LD	Link Disable (Reserved for EP devices)
3 RCB	Read Completion Boundary (RCB)
2 —	Reserved
1-0 ASPM_CTL	Active State Power Management (ASPM) Control

19.4.3.59 PCI Express Link Status Register (Link_Status_Register)

19.4.3.59.1 Offset

Register	Offset
Link_Status_Register	82h

19.4.3.59.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LABS	LBMS	Reserved	SCC	LT	Reserved	NEG_LINK_W						LINK_SP			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.59.3 Fields

Field	Function
15 LABS	Link Autonomous Bandwidth Status **NOTE:** This bit is write-1-clear in RC mode; it is read-only in EP mode
14 LBMS	Link Bandwidth Management Status **NOTE:** This bit is write-1-clear in RC mode; it is read-only in EP mode.
13 —	Reserved
12 SCC	Slot Clock Configuration
11 LT	Link Training
10 —	Reserved.
9-4 NEG_LINK_W	Negotiated link width All other encodings are reserved. The value in this field is undefined when the link is not up. 000001b - x1 000010b - x2

Table continues on the next page...

Field	Function
	000100b - x4
3-0 LINK_SP	Current Link Speed 0001b - 2.5 GT/s 0010b - 5.0 GT/s 0011b - 8.0 GT/s

19.4.3.60 PCI Express Slot Capabilities Register (Slot_Capabilities_Register)

19.4.3.60.1 Offset

Register	Offset
Slot_Capabilities_Register	84h

19.4.3.60.2 Function

This register is supported only for RC mode.

19.4.3.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Physical_Slot_Number													NOCMDCPLSUP	EMIP	SPLS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPLS	SPLV								HPD	HPS	PIP	AIP	MRLSP	PCP	ABP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.60.4 Fields

Field	Function
31-19 Physical_Slot_Number	Physical Slot Number This field indicates the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is globally unique within the chassis. This field must be initialized to 0 for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port.
18 NOCMDCPLSUP	No Command Completed Support
17 EMIP	Electromechanical Interlock Present
16-15 SPLS	Slot Power Limit Scale
14-7 SPLV	Slot Power Limit Value
6 HPD	Hot-Plug Capable **Note:** This chip does not support hot-plug capabilities.
5 HPS	Hot-Plug Surprise **Note:** This chip does not support hot-plug capabilities.
4 PIP	Power Indicator Present **Note:** This chip does not support hot-plug capabilities.
3 AIP	Attention Indicator Present **Note:** This chip does not support hot-plug capabilities.
2 MRLSP	MRL Sensor Present **Note:** This chip does not support hot-plug capabilities.
1 PCP	Power Controller Present **Note:** This chip does not support hot-plug capabilities.
0 ABP	Attention Button Present **Note:** This chip does not support hot-plug capabilities.

19.4.3.61 PCI Express Slot Control Register (Slot_Control_Register)

19.4.3.61.1 Offset

Register	Offset
Slot_Control_Register	88h

19.4.3.61.2 Function

This register is supported only for RC mode.

19.4.3.61.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved			DLLSTCHGEN	EMICTL	PCC	PIC		AIC		HPIE	CCIE	PDCE	MRLSCE	PFDE	ABPE
Reset	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

19.4.3.61.4 Fields

Field	Function
15-13 —	Reserved
12 DLLSTCHGEN	Data Link Layer State Changed Enable **Note:** This chip does not support hot-plug capabilities.
11 EMICTL	Electromechanical Interlock Control **Note:** This chip does not support hot-plug capabilities.
10 PCC	Power Controller Control **Note:** This chip does not support hot-plug capabilities.
9-8 PIC	Power Indicator Control **Note:** This chip does not support hot-plug capabilities.
7-6 AIC	Attention Indicator Control **Note:** This chip does not support hot-plug capabilities.
5 HPIE	Hot-Plug Interrupt Enable **Note:** This chip does not support hot-plug capabilities.
4 CCIE	Command Completed Interrupt Enable **Note:** This chip does not support hot-plug capabilities.
3 PDCE	Presence Detect Changed Enable **Note:** This chip does not support hot-plug capabilities.
2 MRLSCE	MRL Sensor Changed Enable **Note:** This chip does not support hot-plug capabilities.
1 PFDE	Power Fault Detected Enable **Note:** This chip does not support hot-plug capabilities.
0	Attention Button Pressed Enable

Field	Function
ABPE	**Note:** This chip does not support hot-plug capabilities.

19.4.3.62 PCI Express Slot Status Register (Slot_Status_Register)

19.4.3.62.1 Offset

Register	Offset
Slot_Status_Register	8Ah

19.4.3.62.2 Function

This register is supported only for RC mode.

19.4.3.62.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							DLLSTCHG	EM_IL_ST	PDS	MRLSS	CC	PDC	MRLSC	PFD	ABP
W								W1C				W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

19.4.3.62.4 Fields

Field	Function
15-9 —	Reserved
8 DLLSTCHG	Data Link Layer State Changed
7 EM_IL_ST	Electromechanical Interlock Status
6 PDS	Presence Detect State

Table continues on the next page...

Field	Function
	This bit indicates the presence of an adapter in the slot, reflected by the logical OR of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slotâs corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism. This register must be implemented on all Downstream Ports that implement slots. For Downstream Ports not connected to slots (where the Slot Implemented bit of the PCI Express Capabilities register is 0b), this bit must return 1b. Defined encodings are: 0b - Slot Empty 1b - Card Present in slot
5 MRLSS	MRL Sensor State 0b - MRL closed 1b - MRL open
4 CC	Command Completed
3 PDC	Presence Detect Changed
2 MRLSC	MRL Sensor Changed
1 PFD	Power Fault Detected
0 ABP	Attention Button Pressed

19.4.3.63 PCI Express Root Control Register (Root_Control_Register)

19.4.3.63.1 Offset

Register	Offset
Root_Control_Register	8Ch

19.4.3.63.2 Function

This register is supported only for RC mode.

19.4.3.63.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CRSSWVE	PMEIE	SEFEE	SENFEE	SECEE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.63.4 Fields

Field	Function
15-5 —	Reserved
4 CRSSWVE	CRS Software Visibility Enable
3 PMEIE	PME Interrupt Enable
2 SEFEE	System Error on Fatal Error Enable
1 SENFEE	System Error on Non-Fatal Error Enable
0 SECEE	System Error on Correctable Error Enable

19.4.3.64 PCI Express Root Capabilities Register (Root_Capabilities_Register)

19.4.3.64.1 Offset

Register	Offset
Root_Capabilities_Register	8Eh

19.4.3.64.2 Function

This register is supported only for RC mode.

19.4.3.64.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															CRSSWV
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.64.4 Fields

Field	Function
15-1 —	Reserved
0 CRSSWV	CRS Software Visibility

19.4.3.65 PCI Express Root Status Register (Root_Status_Register)

19.4.3.65.1 Offset

Register	Offset
Root_Status_Register	90h

19.4.3.65.2 Function

This register is supported only for RC mode.

19.4.3.65.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														PMEP	PMES
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PME_requester_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.65.4 Fields

Field	Function
31-18 —	Reserved
17 PMEP	PME Pending
16 PMES	PME Status
15-0 PME_requester_ID	PME Requester ID

19.4.3.66 PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)

19.4.3.66.1 Offset

Register	Offset
Device_Capabilities_2_Register	94h

19.4.3.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										ARI_FWD	CPL_TO_DS	CPL_TO_RS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

19.4.3.66.3 Fields

Field	Function
31-6 —	Reserved
5 ARI_FWD	ARI Forwarding Supported
4 CPL_TO_DS	Completion Timeout Disable Supported
3-0 CPL_TO_RS	Completion Timeout Ranges Supported

19.4.3.67 PCI Express Device Control 2 Register (Device_Control_2_Register)

19.4.3.67.1 Offset

Register	Offset
Device_Control_2_Register	98h

19.4.3.67.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						IDO_CPL_EN	IDO_REQ_EN	Reserved		ARIF E	CPL_TOD	CPL_TO_VAL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.67.3 Fields

Field	Function
15-10 —	Reserved
9 IDO_CPL_EN	IDO Completion Enable
8 IDO_REQ_EN	IDO Request Enable
7-6 —	Reserved
5 ARIFE	ARI Forwarding Enable Must be set when RC is communicating with ARI devices. When set will allow RC to issue configuration type 0 cycles with device number != 0.
4 CPL_TOD	Completion Timeout Disable
3-0 CPL_TO_VAL	Completion Timeout Value

19.4.3.68 PCI Express Link Capabilities 2 Register (Link_Capabilities_2_Register)

19.4.3.68.1 Offset

Register	Offset
Link_Capabilities_2_Register	9Ch

19.4.3.68.2 Function

The PCI Express link capability 2 register is shown below.

19.4.3.68.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							Crosslink_Supported	Support_Link_Speed_Vector							Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

19.4.3.68.4 Fields

Field	Function
31-9 —	Reserved
8 Crosslink_Supported	Crosslink Supported
7-1 Support_Link_Speed_Vector	Supported Link Speeds Vector This field indicates the supported Link speed(s) of the associated Port. For each bit, a value of 1 indicates that the corresponding Link speed is supported; otherwise, the Link speed is not supported. Bit definitions are: Bit 1 2.5 GT/s Bit 2 5.0 GT/s Bit 3 8.0 GT/s Bits 7:4 Reserved
0 —	Reserved

19.4.3.69 PCI Express Link Control 2 Register (Link_Control_2_Register)

19.4.3.69.1 Offset

Register	Offset
Link_Control_2_Register	A0h

19.4.3.69.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CDE				CSOS	EMC	TxM			SDE	HWASD	EC	T_LS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

19.4.3.69.3 Fields

Field	Function
15-12 CDE	Compliance Preset/De-emphasis
11 CSOS	Compliance SOS
10 EMC	Enter Modified Compliance
9-7 TxM	Transmit Margin This bit is applicable only for the Gen2 speed.
6 SDE	Selectable De-emphasis
5 HWASD	Hardware Autonomous Speed Disable
4 EC	Enter Compliance
3-0 T_LS	Target Link Speed

19.4.3.70 PCI Express Link Status 2 Register (Link_Status_2_Register)

19.4.3.70.1 Offset

Register	Offset
Link_Status_2_Register	A2h

19.4.3.70.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										LER	EP3S	EP2S	EP1S	EC	DE_LVL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.70.3 Fields

Field	Function
15-6 —	Reserved
5 LER	Link Equalization Request
4 EP3S	Equalization Phase 3 Successful
3 EP2S	Equalization Phase 2 Successful
2 EP1S	Equalization Phase 1 Successful
1 EC	Equalization Complete
0 DE_LVL	Current De-emphasis Level

19.4.3.71 PCI Express MSI-X Message Capability ID Register (MSI_X_Message_Capability_ID_Register)

19.4.3.71.1 Offset

Register	Offset
MSI_X_Message_Capability_ID_Register	B0h

19.4.3.71.2 Function

This register is supported only for EP mode.

19.4.3.71.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MSI_X_Message_Capability_ID							
W								
Reset	0	0	0	1	0	0	0	1

19.4.3.71.4 Fields

Field	Function
7-0 MSI_X_Message_Capability_ID	CAP_ID MSI Message = 0x11

19.4.3.72 PCI Express MSI-X Message Control Register (MSI_X_Message_Control_Register)

19.4.3.72.1 Offset

Register	Offset
MSI_X_Message_Control _Register	B2h

19.4.3.72.2 Function

This register is supported only for EP mode.

19.4.3.72.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSIXE	FM	Reserved			TS										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

19.4.3.72.4 Fields

Field	Function
15 MSIXE	MSI-X Enable If 1 and the MSI Enable bit in the MSI Message Control register is 0, the function is permitted to use MSI-X to request service.
14 FM	Function Mask If 1, all of the vectors associated with the function are masked, regardless of their per-vector Mask bit states. If 0, each vector's Mask bit determines whether the vector is masked or not.!
13-11 —	Reserved
10-0 TS	Table Size System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of "000_0000_0011" indicates a table size of 4.

19.4.3.73 PCI Express MSI-X Table Offset/BIR Register (MSI_X_Table_Offset_BIR_Register)

19.4.3.73.1 Offset

Register	Offset
MSI_X_Table_Offset_BIR_Register	B4h

19.4.3.73.2 Function

This register is supported only for EP mode.

19.4.3.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TABLE_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TABLE_OFFSET													TBIR		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.73.4 Fields

Field	Function
31-3 TABLE_OFFSET	Table Offset Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X Table.
2-0 TBIR	Table BIR Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into Memory Space. BIR Value Base Address register 000b - 10h 001b - 14h 010b - 18h 011b - 1Ch 100b - 20h 101b - 24h

Field	Function
	110b - Reserved 111b - Reserved

19.4.3.74 PCI Express MSI-X PBA Offset/BIR Register (MSI_X_PBA_Offset_BIR_Register)

19.4.3.74.1 Offset

Register	Offset
MSI_X_PBA_Offset_BIR _Register	B8h

19.4.3.74.2 Function

This register is supported only for EP mode.

19.4.3.74.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PBA_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PBA_OFFSET													PBIR		
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.74.4 Fields

Field	Function
31-3 PBA_OFFSET	PBA Offset Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X PBA.
2-0 PBIR	PBA BIR

Field	Function
	Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X PBA into Memory Space. BIR Value Base Address register 000b - 10h 001b - 14h 010b - 18h 011b - 1Ch 100b - 20h 101b - 24h 110b - Reserved 111b - Reserved

19.4.3.75 PCI Express Advanced Error Reporting Capability ID Register (Advanced_Error_Reporting_Capability_ID_Register)

19.4.3.75.1 Offset

Register	Offset
Advanced_Error_Reporting_Capability_ID_Register	100h

19.4.3.75.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Capability_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.75.3 Fields

Field	Function
15-0 Capability_ID	PCI Express Extended Capability ID 0x0001 = Advanced error reporting capability

19.4.3.76 PCI Express Uncorrectable Error Status Register (Uncorrectable_Error_Status_Register)

19.4.3.76.1 Offset

Register	Offset
Uncorrectable_Error_Status_Register	104h

19.4.3.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved											URE	ECRCE	MTLP	RXO	UC
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CA	CTO	FCPE	PTLP	Reserved							DLPE	Reserved			
W	W1C	W1C	W1C	W1C								W1C				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.76.3 Fields

Field	Function
31-21 —	Reserved
20 URE	Unsupported request error status.
19 ECRCE	ECRC error status.
18 MTLP	Malformed TLP status.
17 RXO	Receiver overflow status.
16	Unexpected completion status.

Table continues on the next page...

Field	Function
UC	
15 CA	Completer abort status.
14 CTO	Completion timeout status. Note that a completion timeout error is a fatal error. If a completion timeout error is detected, the system has become unstable. Hot reset is recommended to restore stability of the system.
13 FCPE	Flow control protocol error status.
12 PTLP	Poisoned TLP status.
11-5 —	Reserved
4 DLPE	Data link protocol error status.
3-0 —	Reserved

19.4.3.77 PCI Express Uncorrectable Error Mask Register (Uncorrectable_Error_Mask_Register)

19.4.3.77.1 Offset

Register	Offset
Uncorrectable_Error_Mask_Register	108h

19.4.3.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											UREM	ECRCEM	MTLPM	RXOM	UCM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAM	CTOM	FCPEM	PTLPM	Reserved							DLPEM	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.77.3 Fields

Field	Function
31-21 —	Reserved
20 UREM	Unsupported request error mask.
19 ECRCEM	ECRC error mask.
18 MTLPM	Malformed TLP mask.
17 RXOM	Receiver overflow mask.
16 UCM	Unexpected completion mask.
15 CAM	Completer abort mask.
14 CTOM	Completion timeout mask.
13 FCPEM	Flow control protocol error mask.
12 PTLPM	Poisoned TLP mask.
11-5 —	Reserved
4	Data link protocol error mask.

Table continues on the next page...

Field	Function
DLPEM	
3-0 —	Reserved

19.4.3.78 PCI Express Uncorrectable Error Severity Register (Uncorrectable_Error_Severity_Register)

19.4.3.78.1 Offset

Register	Offset
Uncorrectable_Error_Severity_Register	10Ch

19.4.3.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											URES	ECRCES	MTLPS	RXOS	UCS
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAS	CTOS	FCPES	PTLPS	Reserved							DLPES	Reserved			
Reset	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0

19.4.3.78.3 Fields

Field	Function
31-21 —	Reserved
20 URES	Unsupported request error severity.
19	ECRC error severity.

Table continues on the next page...

Field	Function
ECRCES	
18 MTLPS	Malformed TLP severity.
17 RXOS	Receiver overflow severity.
16 UCS	Unexpected completion severity.
15 CAS	Completer abort severity.
14 CTOS	Completion timeout severity.
13 FCPES	Flow control protocol error severity.
12 PTLPS	Poisoned TLP severity.
11-5 —	Reserved
4 DLPES	Data link protocol error severity.
3-0 —	Reserved

19.4.3.79 PCI Express Correctable Error Status Register (Correctable_Error_Status_Register)

19.4.3.79.1 Offset

Register	Offset
Correctable_Error_Status_Register	110h

19.4.3.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		ADVNFE	RTTO	Reserved			RNR	BDLLP	BTLP	Reserved					RXE
W			W1C	W1C				W1C	W1C	W1C						W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.79.3 Fields

Field	Function
31-14 —	Reserved
13 ADVNFE	Advisory Non-Fatal Error Status indicates the occurrence of the advisory error, and the Advisory Non-Fatal Error Mask bit in the PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register) is checked to determine whether to proceed further with logging and signaling
12 RTTO	Replay timer timeout status
11-9 —	Reserved
8 RNR	REPLAY_NUM Rollover status
7 BDLLP	Bad DLLP status
6 BTLP	Bad TLP status
5-1 —	Reserved
0 RXE	Receiver error status

19.4.3.80 PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register)

19.4.3.80.1 Offset

Register	Offset
Correctable_Error_Mask_Register	114h

19.4.3.80.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ADVNFEM	RTTOM	Reserved			RNRM	BDLLPM	BTLPM	Reserved					RXEM
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.80.3 Fields

Field	Function
31-14 —	Reserved
13 ADVNFEM	Advisory non-fatal error mask. This bit is Set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting
12 RTTOM	Replay timer timeout mask
11-9 —	Reserved
8 RNRM	REPLAY_NUM Rollover mask
7 BDLLPM	Bad DLLP mask
6	Bad TLP mask

Table continues on the next page...

Field	Function
BTLPM	
5-1 —	Reserved
0 RXEM	Receiver error mask

19.4.3.81 PCI Express Advanced Error Capabilities and Control Register (Advanced_Error_Capabilities_and_Control_Register)

19.4.3.81.1 Offset

Register	Offset
Advanced_Error_Capabilities_and_Control_Register	118h

19.4.3.81.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							ECRCCE	ECRCCC	ECRCGE	ECRCGC	First_Error_Pointer				
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

19.4.3.81.3 Fields

Field	Function
31-9 —	Reserved
8 ECRCCE	ECRC checking enable.
7 ECRCCC	ECRC checking capable.
6 ECRCGE	ECRC generation enable.
5 ECRCGC	ECRC generation capable.
4-0 First_Error_Pointer	The First Error Pointer is a read-only register that identifies the bit position of the first error reported in the Uncorrectable Error Status register.

19.4.3.82 PCI Express Header Log Register 1 (Header_Log_Register_DWORD1)

19.4.3.82.1 Offset

Register	Offset
Header_Log_Register_DWORD1	11Ch

19.4.3.82.2 Function

The first DWORD of the PCI Express header log register is shown in the figure below.

19.4.3.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_0								Byte_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_2								Byte_3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.82.4 Fields

Field	Function
31-24 Byte_0	Byte n of the TLP header associated with the error.
23-16 Byte_1	Byte n of the TLP header associated with the error.
15-8 Byte_2	Byte n of the TLP header associated with the error.
7-0 Byte_3	Byte n of the TLP header associated with the error.

19.4.3.83 PCI Express Header Log Register 2 (Header_Log_Register_DWORD2)

19.4.3.83.1 Offset

Register	Offset
Header_Log_Register_DWORD2	120h

19.4.3.83.2 Function

The second DWORD of the PCI Express header log register is shown in the figure below.

19.4.3.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_4								Byte_5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_6								Byte_7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.83.4 Fields

Field	Function
31-24 Byte_4	Byte n of the TLP header associated with the error.
23-16 Byte_5	Byte n of the TLP header associated with the error.
15-8 Byte_6	Byte n of the TLP header associated with the error.
7-0 Byte_7	Byte n of the TLP header associated with the error.

19.4.3.84 PCI Express Header Log Register 3 (Header_Log_Register_DWORD3)

19.4.3.84.1 Offset

Register	Offset
Header_Log_Register_DWORD3	124h

19.4.3.84.2 Function

The third DWORD of the PCI Express header log register is shown in the figure below.

19.4.3.84.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_8								Byte_9							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_A								Byte_B							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.84.4 Fields

Field	Function
31-24 Byte_8	Byte n of the TLP header associated with the error.
23-16 Byte_9	Byte n of the TLP header associated with the error.
15-8 Byte_A	Byte n of the TLP header associated with the error.
7-0 Byte_B	Byte n of the TLP header associated with the error.

19.4.3.85 PCI Express Header Log Register 4 (Header_Log_Register_DWORD4)

19.4.3.85.1 Offset

Register	Offset
Header_Log_Register_DWORD4	128h

19.4.3.85.2 Function

The fourth DWORD of the PCI Express header log register is shown in the figure below.

19.4.3.85.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_C								Byte_D							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_E								Byte_F							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.85.4 Fields

Field	Function
31-24 Byte_C	Byte n of the TLP header associated with the error.
23-16 Byte_D	Byte n of the TLP header associated with the error.
15-8 Byte_E	Byte n of the TLP header associated with the error.
7-0 Byte_F	Byte n of the TLP header associated with the error.

19.4.3.86 PCI Express Root Error Command Register (Root_Error_Command_Register)

19.4.3.86.1 Offset

Register	Offset
Root_Error_Command_Register	12Ch

19.4.3.86.2 Function

This register is supported only for RC mode.

19.4.3.86.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													FERE	NFERE	CERE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.86.4 Fields

Field	Function
31-3 —	Reserved
2 FERE	Fatal error reporting enable.
1 NFERE	Non-fatal error reporting enable
0 CERE	Correctable error reporting enable

19.4.3.87 PCI Express Root Error Status Register (Root_Error_Status_Register)

19.4.3.87.1 Offset

Register	Offset
Root_Error_Status_Register	130h

19.4.3.87.2 Function

This register is supported only for RC mode.

19.4.3.87.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AEIMN					Reserved										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									FEMR	NFEMR	FUF	MEFNFR	EFNFR	MECR	ECR
W										W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.87.4 Fields

Field	Function
31-27 AEIMN	Advanced error interrupt message number.
26-7 —	Reserved
6 FEMR	Fatal error messages received.
5 NFEMR	Non-fatal error messages received.
4 FUF	First uncorrectable fatal.
3 MEFNFR	Multiple ERR_FATAL/NONFATAL received.
2 EFNFR	ERR_FATAL/NONFATAL received.
1 MECR	Multiple ERR_COR received.
0 ECR	ERR_COR received.

19.4.3.88 PCI Express Correctable Error Source ID Register (Correctable_Error_Source_ID_Register)

19.4.3.88.1 Offset

Register	Offset
Correctable_Error_Source_ID_Register	134h

19.4.3.88.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_COR_Source_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.88.3 Fields

Field	Function
15-0 ERR_COR_Source_ID	Loaded with the requester ID indicated in the received ERR_COR Message when the ERR_COR Received register is not already set. Default value of this field is 0.

19.4.3.89 PCI Express Error Source ID Register (Error_Source_ID_Register)

19.4.3.89.1 Offset

Register	Offset
Error_Source_ID_Register	136h

19.4.3.89.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Error_Source_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.89.3 Fields

Field	Function
15-0 Error_Source_ID	ERR_FATAL/NONFATAL source ID

19.4.3.90 ARI Capability Header (ARI_Capability_Header)

19.4.3.90.1 Offset

Register	Offset
ARI_Capability_Header	148h

19.4.3.90.2 Function

This register applies to SR-IOV-only.

19.4.3.90.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NEXT_OFFSET												CAP_VERSION			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Extended_Capability_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

19.4.3.90.4 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset This field is a pointer to the next capability structure. **NOTE:** The reset value of this field depends on the instantiation of the module. See PCI Express configuration registers for the specific values.
19-16 CAP_VERSION	Capability Version Indicates the version of the Capability structure present.
15-0 Extended_Capability_ID	PCI Express Extended Capability ID 0x000E = ARI capability

19.4.3.91 ARI Capability Register (ARI_Capability_Register)

19.4.3.91.1 Offset

Register	Offset
ARI_Capability_Register	14Ch

19.4.3.91.2 Function

This register applies to SR-IOV-only.

19.4.3.91.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NFN								Reserved						AFGC	MFGC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.91.4 Fields

Field	Function
15-8	Next Function Number

Table continues on the next page...

Field	Function
NFN	This field indicates the Function Number of the next higher numbered Function in the Device, or 00h if there are no higher numbered Functions. Function 0 starts this linked list of Functions.
7-2 —	Reserved
1 AFGC	A ACS Function Groups Capability. Applicable only for Function 0; must be 0b for all other Functions. If 1b, indicates that the ARI Device supports Function Group level granularity for ACS P2P Egress Control via its ACS Capability structures.
0 MFGC	M MFVC Function Groups Capability. Applicable only for Function 0; must be 0b for all other Functions. If 1b, indicates that the ARI Device supports Function Group level arbitration via its Multi-Function Virtual Channel (MFVC) Capability structure.

19.4.3.92 ARI Control Register (ARI_Control_Register)

19.4.3.92.1 Offset

Register	Offset
ARI_Control_Register	14Eh

19.4.3.92.2 Function

This register applies to SR-IOV-only.

19.4.3.92.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									F G			Reserved		AFG E	MFGE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.92.4 Fields

Field	Function
15-7	Reserved

Table continues on the next page...

Field	Function
—	
6-4 FG	Function Group Assigns a Function Group Number to this Function.
3-2 —	Reserved
1 AFGE	A ACS Function Groups Enable. Applicable only for Function 0; must be hardwired to 0b for all other Functions. When set, each Function in the ARI Device must associate bits within its Egress Control Vector with Function Group Numbers rather than Function Numbers. Default value of this field is 0b. Must be hardwired to 0b if the ACS Function Groups Capability bit is 0b.
0 MFGE	M MFVC Function Groups Enable. Applicable only for Function 0; must be hardwired to 0b for all other Functions. When set, the ARI Device must interpret entries in its Function Arbitration Table as Function Group Numbers rather than Function Numbers. Default value of this field is 0b. Must be hardwired to 0b if the MFVC Function Groups Capability bit is 0b.

19.4.3.93 Secondary PCI Express Extended Capability Header (SPCIE_CAP_HEADER_REG)

19.4.3.93.1 Offset

Register	Offset
SPCIE_CAP_HEADER_REG	158h

19.4.3.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NEXT_OFFSET												CAP_VERSION			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EXTENDED_CAP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

19.4.3.93.3 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset This field is a pointer to the next capability structure. **NOTE:** The reset value of this field depends on the instantiation of the module. See PCI Express configuration registers for the specific values.
19-16 CAP_VERSION	Capability Version Indicates the version of the Capability structure present.
15-0 EXTENDED_CAP_ID	PCI Express Extended Capability ID 0019h = Secondary PCI Express extended capability

19.4.3.94 Link Control 3 Register (LINK_CONTROL3_REG)

19.4.3.94.1 Offset

Register	Offset
LINK_CONTROL3_REG	15Ch

19.4.3.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														EQ_REQ_INT_EN	PERFORM_EQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.94.3 Fields

Field	Function
31-2 —	Reserved
1 EQ_REQ_INT_EN	Link Equalization Request Interrupt Enable This bit is RW for Downstream Ports only.
0 PERFORM_EQ	Perform Equalization This bit is RW for Downstream Ports only.

19.4.3.95 Lane Error Status Register (LANE_ERR_STATUS_REG)

19.4.3.95.1 Offset

Register	Offset
LANE_ERR_STATUS_REG	160h

19.4.3.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												LANE_ERR_STATUS			
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.95.3 Fields

Field	Function
31-4 —	Reserved
3-0 LANE_ERR_STATUS	Lane Error Status Each bit indicates if the corresponding Lane detected a Lane-based error.

19.4.3.96 Lane Equalization Control Register (LANE0_EQUALIZATION_CONTROL - LANE3_EQUALIZATION_CONTROL)

19.4.3.96.1 Offset

Register	Offset
LANE0_EQUALIZATION_CONTROL	164h
LANE1_EQUALIZATION_CONTROL	166h
LANE2_EQUALIZATION_CONTROL	168h
LANE3_EQUALIZATION_CONTROL	16Ah

19.4.3.96.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	USP_RX_PRESET_HINT			USP_TX_PRESET				Reserved	DSP_RX_PRESET_HINT			DSP_TX_PRESET			
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

19.4.3.96.3 Fields

Field	Function
15 —	Reserved
14-12 USP_RX_PRESET_HINT	Upstream Port Receiver Preset Hint
11-8 USP_TX_PRESET	Upstream Port Transmitter Preset
7 —	Reserved
6-4 DSP_RX_PRESET_HINT	Downstream Port Receiver Preset Hint
3-0 DSP_TX_PRESET	Downstream Port Transmitter Preset

19.4.3.97 SR-IOV Extended Capability Header (SR_IOV_Extended_Capability_ID_Register)

19.4.3.97.1 Offset

Register	Offset
SR_IOV_Extended_Capability_ID_Register	178h

19.4.3.97.2 Function

This register applies to SR-IOV-only.

19.4.3.97.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Next_Capability_Offset												Capability_Version			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Extended_Capability_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

19.4.3.97.4 Fields

Field	Function
31-20 Next_Capability_Offset	Next Capability Offset
19-16 Capability_Version	Capability Version
15-0 Extended_Capability_ID	PCI Express Extended Capability ID 0x0010 = SR-IOV extended capability

19.4.3.98 PCI Express SR-IOV Capabilities Register (SR_IOV_Capabilities_Register)

19.4.3.98.1 Offset

Register	Offset
SR_IOV_Capabilities_Register	17Ch

19.4.3.98.2 Function

This register applies to SR-IOV-only.

19.4.3.98.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VFMIMN											Reserved				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														ARICHR	VFMC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.98.4 Fields

Field	Function
31-21 VFMIMN	VF Migration Interrupt Message Number Indicates the MSI/MSI-X vector used for migration interrupts. The value in this field is undefined if VF Migration Capable is clear.
20-2 —	Reserved
1 ARICHR	ARI Capable Hierarchy Preserved If set, the ARI Capable Hierarchy bit is preserved across certain power state transitions.
0 VFMC	VF Migration Capable If set, the PF is Migration Capable and operating under a Migration Capable MR-PCIM.

19.4.3.99 PCI Express SR-IOV Control Register (SR_IOV_Control_Register)

19.4.3.99.1 Offset

Register	Offset
SR_IOV_Control_Register	180h

19.4.3.99.2 Function

This register applies to SR-IOV-only.

19.4.3.99.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											ARICH	VFMSE	VFMIE	VFME	VFE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.99.4 Fields

Field	Function
15-5 —	Reserved
4 ARICH	ARI Capable Hierarchy PCI Express Endpoint: The device is permitted to locate VFs in Function numbers 8 to 255 of the captured Bus Number. Default value is 0. This field is RW in the lowest numbered PF of the Device and is Read Only Zero in all other PFs. Root Complex Integrated Endpoint: Not applicable. Bit is hardwired to 0
3 VFMSE	VF MSE Memory Space Enable for Virtual Functions.
2 VFMIE	VF Migration Interrupt Enable Enables/Disables VF Migration State Change Interrupt. Default value is 0b.
1 VFME	VF Migration Enable Enables/Disables VF Migration Support. Default value is 0b.
0 VFE	VF Enable Enables/Disables VFs. Default value is 0.

19.4.3.100 PCI Express SR-IOV Status Register (SR_IOV_Status_Register)

19.4.3.100.1 Offset

Register	Offset
SR_IOV_Status_Register	182h

19.4.3.100.2 Function

This register applies to SR-IOV-only.

19.4.3.100.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															VFMS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.100.4 Fields

Field	Function
15-1 —	Reserved
0 VFMS	VF Migration Status Indicates a VF Migration In or Migration Out Request has been issued by MR-PCIM. To determine the cause of the event, software may scan the VF State Array. Default value is 0b.

19.4.3.101 PCI Express SR-IOV Initial VF Register (SR_IOV_Initial_VF_Register)

19.4.3.101.1 Offset

Register	Offset
SR_IOV_Initial_VF_Register	184h

19.4.3.101.2 Function

This register applies to SR-IOV-only.

19.4.3.101.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	InitialVF															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

19.4.3.101.4 Fields

Field	Function
15-0 InitialVF	Initial VF Indicates to SR-PCIM the number of VFs that are initial associated with the PF. The minimum value of InitialVFs is 0.

19.4.3.102 PCI Express SR-IOV Total VF Register (SR_IOV_Total_VF_Register)

19.4.3.102.1 Offset

Register	Offset
SR_IOV_Total_VF_Register	186h

19.4.3.102.2 Function

This register applies to SR-IOV-only.

19.4.3.102.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TotalVF															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

19.4.3.102.4 Fields

Field	Function
15-0 TotalVF	Total VF Indicates the maximum number of VFs that could be associated with the PF. The minimum value of TotalVFs is 0.

19.4.3.103 PCI Express SR-IOV Number VF Register (SR_IOV_Number_VF_Register)

19.4.3.103.1 Offset

Register	Offset
SR_IOV_Number_VF_Register	188h

19.4.3.103.2 Function

This register applies to SR-IOV-only.

19.4.3.103.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NumberVF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.103.4 Fields

Field	Function
15-0 NumberVF	Number of VF Controls the number of VFs that are visible. Only supports a number that is a power of 2 (that is, 1, 2, 4, 8, 16...).

19.4.3.104 PCI Express SR-IOV Function Dependency Link Register (SR_IOV_Function_Dependency_Link_Register)

19.4.3.104.1 Offset

Register	Offset
SR_IOV_Function_Dependency_Link_Register	18Ah

19.4.3.104.2 Function

This register applies to SR-IOV-only.

19.4.3.104.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	FDL							
Reset	0	0	0	0	0	0	0	0

19.4.3.104.4 Fields

Field	Function
7-0 FDL	Function Dependency Link This field describes dependencies between PFs.

19.4.3.105 PCI Express SR-IOV First VF Offset Register (SR_IOV_First_VF_Offset_Register)

19.4.3.105.1 Offset

Register	Offset
SR_IOV_First_VF_Offset_Register	18Ch

19.4.3.105.2 Function

This register applies to SR-IOV-only.

19.4.3.105.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FVFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

19.4.3.105.4 Fields

Field	Function
15-0 FVFO	First VF Offset Defines the Routing ID offset of the first VF that is associated with the PF that contains this Capability structure.

19.4.3.106 PCI Express SR-IOV VF Stride Register (SR_IOV_VF_Stride_Register)

19.4.3.106.1 Offset

Register	Offset
SR_IOV_VF_Stride_Register	18Eh

19.4.3.106.2 Function

This register applies to SR-IOV-only.

19.4.3.106.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VFS															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

19.4.3.106.4 Fields

Field	Function
15-0 VFS	VF Stride Defines the Routing ID offset from one VF to the next one for all VFs associated with the PF that contains this Capability structure.

19.4.3.107 PCI Express SR-IOV Device ID Register (SR_IOV_Device_ID_Register)

19.4.3.107.1 Offset

Register	Offset
SR_IOV_Device_ID_Register	192h

19.4.3.107.2 Function

This register applies to SR-IOV-only.

19.4.3.107.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Device_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.107.4 Fields

Field	Function
15-0 Device_ID	Device ID

19.4.3.108 PCI Express SR-IOV Supported Page Size Register (SR_IOV_Supported_Page_Size_Register)

19.4.3.108.1 Offset

Register	Offset
SR_IOV_Supported_Page_Size_Register	194h

19.4.3.108.2 Function

This register applies to SR-IOV-only.

19.4.3.108.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SUPS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SUPS															
W																
Reset	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	1

19.4.3.108.4 Fields

Field	Function
31-0	Supported Page Size

Field	Function
SUPS	This field indicates the page sizes supported by the PF. This PF supports a page size of 2^{n+12} if bit n is Set. For example, if bit 0 is Set, the PF supports 4-KB page sizes. PFs are required to support 4-KB, 8-KB, 64-KB, 256-KB, 1-MB, and 4-MB page sizes. All other page sizes are optional.

19.4.3.109 PCI Express SR-IOV System Page Size Register (SR_IOV_System_Page_Size_Register)

19.4.3.109.1 Offset

Register	Offset
SR_IOV_System_Page_Size_Register	198h

19.4.3.109.2 Function

This register applies to SR-IOV-only.

19.4.3.109.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SYPS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SYPS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.109.4 Fields

Field	Function
31-0 SYPS	System Page Size Defines the page size the system will use to map the VFs memory addresses.

19.4.3.110 PCI Express SR-IOV VF Base Address Register 0 (VFBAR0)

19.4.3.110.1 Offset

Register	Offset
VFBAR0	19Ch

19.4.3.110.2 Function

This register applies to SR-IOV-only.

The PCI Express VF base address registers (VF BARs) point to the beginning of distinct address ranges which the device should claim. In EP mode, the device supports two 32-bit memory space BARs, and two 64-bit memory space BARs.

Base address register 0 at offset 0x174 and base address register 1 at offset 0x178 are used to define the inbound memory windows in the 32-bit memory space.

19.4.3.110.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

19.4.3.110.4 Fields

Field	Function
31-12 ADDRESS	Base Address Indicates the base address where the inbound memory window begins.

Table continues on the next page...

Field	Function
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00 Locate anywhere in 32-bit address space.
0 —	Reserved.

19.4.3.111 PCI Express SR-IOV VF Base Address Register 1 (VFBAR1)

19.4.3.111.1 Offset

Register	Offset
VFBAR1	1A0h

19.4.3.111.2 Function

This register applies to SR-IOV-only.

The PCI Express VF base address registers (VF BARs) point to the beginning of distinct address ranges which the device should claim. In EP mode, the device supports two 32-bit memory space BARs, and two 64-bit memory space BARs.

Base address register 0 at offset 0x174 and base address sregister 1 at offset 0x178 are used to define the inbound memory windows in the 32-bit memory space.

19.4.3.111.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

19.4.3.111.4 Fields

Field	Function
31-12 ADDRESS	Base Address Indicates the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	>Prefetchable
2-1 TYPE	Type 00 Locate anywhere in 32-bit address space.
0 —	Reserved.

19.4.3.112 PCI Express SR-IOV VF Base Address Register 2 (VFBAR2)

19.4.3.112.1 Offset

Register	Offset
VFBAR2	1A4h

19.4.3.112.2 Function

This register applies to SR-IOV-only.

The PCI Express VF base address registers (VF BARs) point to the beginning of distinct address ranges which the device should claim. In EP mode, the device supports two 32-bit memory space BARs, and two 64-bit memory space BARs.

Base address register 2 at offset 0x17C and base address register 4 at offset 0x184 are used to define the lower portion of the 64-bit inbound memory windows.

19.4.3.112.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

19.4.3.112.4 Fields

Field	Function
31-12 ADDRESS	Base Address (lower portion) Indicates the lower portion of the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 10b - Locate anywhere in 64-bit address space.
0 —	Reserved.

19.4.3.113 PCI Express SR-IOV VF Base Address Register 3 (VFBAR3)

19.4.3.113.1 Offset

Register	Offset
VFBAR3	1A8h

19.4.3.113.2 Function

This register applies to SR-IOV-only.

The PCI Express VF base address registers (VF BARs) point to the beginning of distinct address ranges which the device should claim. In EP mode, the device supports two 32-bit memory space BARs, and two 64-bit memory space BARs.

Base address register 3 at offset 0x180 and base address register 5 at offset 0x188 are used to define the upper portion of the 64-bit inbound memory windows.

19.4.3.113.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.113.4 Fields

Field	Function
31-0 ADDRESS	Base Address (upper portion) Indicates the upper portion of the base address where the inbound memory window begins. If no access to local memory is to be permitted by external requesters, then all bits are programmed.

19.4.3.114 PCI Express SR-IOV VF Base Address Register 4 (VFBAR4)

19.4.3.114.1 Offset

Register	Offset
VFBAR4	1ACh

19.4.3.114.2 Function

This register applies to SR-IOV-only.

The PCI Express VF base address registers (VF BARs) point to the beginning of distinct address ranges which the device should claim. In EP mode, the device supports two 32-bit memory space BARs, and two 64-bit memory space BARs.

Base address register 2 at offset 0x17C and base address register 4 at offset 0x184 are used to define the lower portion of the 64-bit inbound memory windows.

19.4.3.114.3 Diagram

19.4.3.114.4 Fields

Field	Function
31-12 ADDRESS	Base Address (lower portion) Indicates the lower portion of the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.

Table continues on the next page...

Field	Function
3 PREF	Prefetchable
2-1 TYPE	Type 10b - Locate anywhere in 64-bit address space.
0 —	Reserved.

19.4.3.115 PCI Express SR-IOV VF Base Address Register 5 (VFBAR5)

19.4.3.115.1 Offset

Register	Offset
VFBAR5	1B0h

19.4.3.115.2 Function

This register applies to SR-IOV-only.

The PCI Express VF base address registers (VF BARs) point to the beginning of distinct address ranges which the device should claim. In EP mode, the device supports two 32-bit memory space BARs, and two 64-bit memory space BARs.

Base address register 3 at offset 0x180 and base address register 5 at offset 0x188 are used to define the upper portion of the 64-bit inbound memory windows.

19.4.3.115.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.115.4 Fields

Field	Function
31-0 ADDRESS	Base Address (upper portion) Indicates the upper portion of the base address where the inbound memory window begins. If no access to local memory is to be permitted by external requesters, then all bits are programmed.

19.4.3.116 PCI Express SR-IOV Migration State Array Offset Register (SR_IOV_Migration_State_Array_Offset_Register)

19.4.3.116.1 Offset

Register	Offset
SR_IOV_Migration_State_Array_Offset_Register	1B4h

19.4.3.116.2 Function

This register applies to SR-IOV-only.

19.4.3.116.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSAO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSAO													MSBIR		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.116.4 Fields

Field	Function
31-3	VF Migration State Offset

Table continues on the next page...

Field	Function
MSAO	
2-0 MSBIR	VF Migration State BIR

19.4.3.117 Symbol Timer Register and Filter Mask 1 Register (SYMBOL_TIMER_FILTER_1_OFF)

19.4.3.117.1 Offset

Register	Offset
SYMBOL_TIMER_FILTER_1_OFF	71Ch

19.4.3.117.2 Function

The Filter Mask 1 Register modifies the receive filtering and error handling rules. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule

19.4.3.117.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASK_RADM_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DISABLE_FC_WD_TIMER	Reserved				SKP_INT_VAL										
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

19.4.3.117.4 Fields

Field	Function
31-16 MASK_RADM_1	Filter Mask 1 The Filter Mask 1 Register modifies the receive filtering and error handling rules. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31]: CX_FLT_MASK_RC_CFG_DISCARD 0: For RC filter to not allow CFG transaction being received 1: For RC filter to allow CFG transaction being received [30]: CX_FLT_MASK_RC_IO_DISCARD 0: For RC filter to not allow IO transaction being received 1: For RC filter to allow IO transaction being received [29]: CX_FLT_MASK_MSG_DROP 0: Drop MSG TLP (except for Vendor MSG). 1: Do not Drop MSG (except for Vendor MSG). Send message TLPs to your application. This bit only controls message TLPs other than Vendor MSGs. Vendor MSGs are controlled by Filter Mask Register 2, bits [1:0]. **NOTE:** The forwarding of message TLPs may be undesirable. If so, write a 0 to this bit to disable forwarding of message TLPs. [28]: CX_FLT_MASK_CPL_ECRC_DISCARD Only used when completion queue is advertized with infinite credits and is in store-and-forward mode. 0: Discard completions with ECRC errors 1: Allow completions with ECRC errors to be passed up Reserved field for SW. [27]: CX_FLT_MASK_ECRC_DISCARD 0: Discard TLPs with ECRC errors 1: Allow TLPs with ECRC errors to be passed up [26]: CX_FLT_MASK_CPL_LEN_MATCH 0: Enforce length match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err 1: MASK length match for completions [25]: CX_FLT_MASK_CPL_ATTR_MATCH 0: Enforce attribute match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask attribute match for completions [24]: CX_FLT_MASK_CPL_TC_MATCH 0: Enforce Traffic Class match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask Traffic Class match for completions [23]: CX_FLT_MASK_CPL_FUNC_MATCH 0: Enforce function match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask function match for completions

Table continues on the next page...

Field	Function
	[22]: CX_FLT_MASK_CPL_REQID_MATCH 0: Enforce Req. Id match for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask Req. Id match for completions [21]: CX_FLT_MASK_CPL_TAGERR_MATCH 0: Enforce Tag Error Rules for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask Tag Error Rules for completions [20]: CX_FLT_MASK_LOCKED_RD_AS_UR 0: Treat locked Read TLPs as UR for EP. Treat locked Read TLPs as supported for RC. 1: Treat locked Read TLPs as Supported for EP. Treat locked Read TLPs as UR for RC. [19]: CX_FLT_MASK_CFG_TYPE1_RE_AS_UR 0: Treat CFG type1 TLPs as UR for EP. Treat CFG type1 TLPs as supported for RC. 1: Treat CFG type1 TLPs as Supported for EP. Treat CFG type1 TLPs as UR for RC. When SR-IOV mode is enabled, then this bit is set to allow the filter to process Type 1 Config requests if the EP consumes more than one bus number. [18]: CX_FLT_MASK_UR_OUTSIDE_BAR 0: Treat out-of-bar TLPs as UR 1: Do not treat out-of-bar TLPs as UR [17]: CX_FLT_MASK_UR_POIS 0: Treat poisoned request TLPs as UR 1: Do not treat poisoned request TLPs as UR The native PEX controller always passes poisoned completions to your application. [16]: CX_FLT_MASK_UR_FUNC_MISMATCH 0: Treat Function MisMatched TLPs as UR 1: Do not treat Function MisMatched TLPs as UR *Note*: This register field is sticky.
15 DISABLE_FC_WD_TIMER	Disable FC Watchdog Timer Disable FC Watchdog Timer. *Note*: This register field is sticky.
14-11 —	Reserved
10-0 SKP_INT_VAL	SKP Interval Value The number of symbol times to wait between transmitting SKP ordered sets. Note that the PEX controller actually waits the number of symbol times in this register plus 1 between transmitting SKP ordered sets. Your application must program this register accordingly. For example, if 1536 were programmed into this register (in a 250 MHz PEX controller), then the PEX controller actually transmits SKP ordered sets once every 1537 symbol times. The value programmed to this register is actually clock ticks and not symbol times. In a 125 MHz PEX controller, programming the value programmed to this register should be scaled down by a factor of 2 (because one clock tick = two symbol times in this case). *Note:*This value is not used at Gen3 speed; the skip interval is hardcoded to 370 blocks. *Note*: This register field is sticky.

19.4.3.118 Gen3 Control Register (GEN3_RELATED_OFF)

19.4.3.118.1 Offset

Register	Offset
GEN3_RELATED_OFF	890h

19.4.3.118.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															GEN3_EQUALIZATION_DISABLE
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			RXEQ_PH01_EN	Reserved	EQ_EIEOS_CNT	Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.3.118.3 Fields

Field	Function
31-17 —	Reserved
16	Equalization Disable Disable equalization feature.

Table continues on the next page...

Field	Function
GEN3_EQUALIZATION_DISABLE	*Note*: This register field is sticky.
15-13 —	Reserved
12 RXEQ_PH01_EN	Rx Equalization Phase 0/Phase 1 Hold Enable When this bit is set the upstream port holds phase 0 (the downstream port holds phase 1) for 10ms. Holding phase 0 or phase 1 can be used to allow sufficient time for Rx Equalization to be performed by the PHY. *Note*: This register field is sticky.
11 —	Reserved
10 EQ_EIEOS_CNT	Equalization EIEOS Count Reset Disable Disable requesting reset of EIEOS count during equalization. *Note*: This register field is sticky.
9-0 —	Reserved

19.4.3.119 DBI Read-only Write Enable Register (MISC_CONTROL_1_OFF)

19.4.3.119.1 Offset

Register	Offset
MISC_CONTROL_1_OFF	8BCh

19.4.3.119.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															RO_WR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.119.3 Fields

Field	Function
31-1 —	Reserved
0 RO_WR_EN	Read-only write enable This field enables writing (using internal accesses only) to some read-only and hardware-initialized bits in the configuration registers. 0b - Read-only fields are read only. 1b - Enables writing to some read-only fields

19.4.3.120 Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)

19.4.3.120.1 Offset

Register	Offset
COHERENCY_CONTROL_1_OFF	8E0h

19.4.3.120.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_MEMTYPE_BOUNDARY_LOW_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_MEMTYPE_BOUNDARY_LOW_ADDR														Reserved	CFG_MEMTYPE_VALUE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.120.3 Fields

Field	Function
31-2 CFG_MEMTYPE_BOUNDARY_LOW_ADDR	Boundary lower address for memory type Bits [31:0] of dword-aligned address of the boundary for memory type. The two lower address least-significant bits must be 00. Addresses up to but not including this value are in the lower address space region; addresses equal or greater than this value are in the upper address space region. *Note*: This register field is sticky.
1 —	Reserved
0 CFG_MEMTYPE_VALUE	Memory type Sets the memory type for the lower and upper regions of the address space: *Note*: This register field is sticky. 0b - lower = CCSR; upper = Memory 1b - Reserved

19.4.3.121 Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)

19.4.3.121.1 Offset

Register	Offset
COHERENCY_CONTROL_2_OFF	8E4h

19.4.3.121.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_MEMTYPE_BOUNDARY_HIGH_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_MEMTYPE_BOUNDARY_HIGH_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.121.3 Fields

Field	Function
31-0 CFG_MEMTYPE_BOUNDARY_HIGH_ADDR	Boundary upper address for memory type Bits [63:32] of the 64-bit dword-aligned address of the boundary for memory type. *Note*: This register field is sticky.

19.4.3.122 Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)

19.4.3.122.1 Offset

Register	Offset
COHERENCY_CONTROL_3_OFF	8E8h

19.4.3.122.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.122.3 Fields

Field	Function
31-0 —	Reserved Must be 0000_0000h.

19.4.3.123 iATU Index Register (IATU_VIEWPORT_OFF)

19.4.3.123.1 Offset

Register	Offset
IATU_VIEWPORT_OFF	900h

19.4.3.123.2 Function

The iATU registers are programmed through an indirect addressing scheme (using this index register) to reduce the address footprint in the PCI Express extended configuration space.

The iATU can remap 256 outbound and 24 inbound address regions. This index register determines the memory region to be accessed.

19.4.3.123.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGION_DIR	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								REGION_INDEX							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.123.4 Fields

Field	Function
31 REGION_DIR	Region Direction Defines the region being accessed as either 0: Outbound 1: Inbound
30-8 —	Reserved
7-0 REGION_INDEX	Region Index Defines which region is being accessed when writing to the control, base, limit and target registers. Must not be set to a number greater than 255 when an outbound region is being accessed. Must not be set to a value greater than 23 when an inbound region is being accessed.

19.4.3.124 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_INBOUND_0)

19.4.3.124.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_0	904h

19.4.3.124.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							CTRL_1_FUNC_NUM					Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					Reserved		Reserved	Reserved			TYPE				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.124.3 Fields

Field	Function
31-25 —	Reserved
24-20 CTRL_1_FUNC_NUM	Function Number. **MEM-I/O:** DM/EP/SW: When the Address and BAR matching logic in the PEX controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. **CFG0/CFG1:** DM/EP/SW: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. RC: Reserved. You must set this bit to "0"
19-18 —	Reserved
17-16 —	Reserved
15-11 —	Reserved
10-9 —	Reserved

Table continues on the next page...

Field	Function
8 —	Reserved
7-5 —	Reserved
4-0 TYPE	Type When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful).

19.4.3.125 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)

19.4.3.125.1 Offset

Register	Offset
IATU_REGION_CTRL_1 _OFF_OUTBOUND_0	904h

19.4.3.125.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							CTRL_1_FUNC_NUM					Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					ATTR		TD	TC			TYPE				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.125.3 Fields

Field	Function
31-25 —	Reserved
24-20 CTRL_1_FUNC_NUM	Function Number DM/EP/SW: When the address of an outbound TLP is matched to this region and the "Function Number Translation Bypass" field in the "iATU Control 2 Register" is "0", then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the PEX controller is enabled. RC: Reserved. You must set this bit to "0"
19-18 —	Reserved
17-16 —	Reserved
15-11 —	Reserved
10-9 ATTR	Attribute (Attr) When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register.
8 TD	TLP Digest (TD) When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register.
7-5 TC	Traffic class (TC) When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register.
4-0 TYPE	Type When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register.

19.4.3.126 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)

19.4.3.126.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_0	908h

19.4.3.126.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	REGION_EN	MATCH_MODE	Reserved	Reserved	Reserved	Reserved	Reserved		Reserved		Reserved	Reserved	FUNC_NUM_MATCH_EN	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	Reserved	Reserved			BAR_NUM			Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.126.3 Fields

Field	Function
31 REGION_EN	Region Enable This bit must be set to "1" for address translation to take place.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers.

Table continues on the next page...

Field	Function
	1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header.
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-24 —	Reserved
23-22 —	Reserved
21 —	Reserved
20 —	Reserved
19 FUNC_NUM_MATCH_EN	Function Number Match Enable **RC/SW:** Reserved. You must set this bit to "0". **DM/EP:** Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed.
18 —	Reserved
17 —	Reserved
16 —	Reserved
15 —	Reserved
14 —	Reserved
13-11 —	Reserved
10-8 BAR_NUM	BAR Number

Table continues on the next page...

Field	Function
	DM/EP/SW: When the BAR number of an inbound MEM or IO TLP that is matched by the normal internal BAR address matching mechanism is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. 000b - BAR0 001b - BAR1 010b - BAR2 011b - BAR3 100b - BAR4 101b - BAR5 110b - ROM 111b - reserved IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b-101b and that BAR configured as an IO BAR. **RC:** Reserved. Must be 0.
7-0 —	Reserved

19.4.3.127 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)

19.4.3.127.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_0	908h

19.4.3.127.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGION_EN	Reserved	Reserved	CFG_SHIFT_MODE	Reserved	Reserved							Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MSG_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.127.3 Fields

Field	Function
31 REGION_EN	Region Enable This bit must be set to "1" for address translation to take place.
30 —	Reserved
29 —	Reserved
28 CFG_SHIFT_MODE	CFG Shift Mode This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address.Shifts bits [27:12] of the untranslated address to form bits [31:16] of the translated address.
27 —	Reserved
26-20 —	Reserved
19 —	Reserved
18-8 —	Reserved
7-0 MSG_CODE	Message Code Message TLPs: When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register.

19.4.3.128 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)

19.4.3.128.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_0	90Ch

19.4.3.128.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_BASE_RW				LWR_BASE_HW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.128.3 Fields

Field	Function
31-12 LWR_BASE_RW	Lower base address bits—programmable Forms bits [31:12] of the start address of the address region to be translated.
11-0 LWR_BASE_HW	Lower base address bits—hardwired Forms bits [11:0] of the start address of the address region to be translated. The start address must be aligned to a 4 KB boundary, so these bits are always 0. A write to this location is ignored by the PEX controller.

19.4.3.129 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)

19.4.3.129.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0	90Ch

19.4.3.129.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_BASE_RW				LWR_BASE_HW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.129.3 Fields

Field	Function
31-12 LWR_BASE_RW	Lower base address bits—programmable Forms bits [31:12] of the start address of the address region to be translated.
11-0 LWR_BASE_HW	Lower base address bits—hardwired Forms bits [11:0] of the start address of the address region to be translated. The start address must be aligned to a 4 KB boundary, so these bits are always 0. A write to this location is ignored by the PEX controller.

19.4.3.130 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)

19.4.3.130.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_0	910h

19.4.3.130.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.130.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Upper base address bits Forms bits [63:32] of the start (and end) address of the address region to be translated.

19.4.3.131 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)

19.4.3.131.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0	910h

19.4.3.131.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.131.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Upper base address bits Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect.

19.4.3.132 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_INBOUND_0)

19.4.3.132.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_0	914h

19.4.3.132.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_RW				LIMIT_ADDR_HW											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.132.3 Fields

Field	Function
31-12 LIMIT_ADDR_RW	Limit address bits—programmable Forms bits [31:12] of the end address of the address region to be translated.
11-0 LIMIT_ADDR_HW	Limit address bits—hardwired Forms bits [11:0] of the end address of the address region to be translated. Address regions must be aligned to a 4 KB boundary, so these bits are always 1. A write to this location is ignored by the PEX controller.

19.4.3.133 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)

19.4.3.133.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_0	914h

19.4.3.133.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_RW				LIMIT_ADDR_HW											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.133.3 Fields

Field	Function
31-12 LIMIT_ADDR_RW	Limit address bits—programmable Forms bits [31:12] of the end address of the address region to be translated.
11-0 LIMIT_ADDR_HW	Limit address bits—hardwired Forms bits [11:0] of the end address of the address region to be translated. Address regions must be aligned to a 4 KB boundary, so these bits are always 1. A write to this location is ignored by the PEX controller.

19.4.3.134 iATU Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)

19.4.3.134.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_0	918h

19.4.3.134.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_TARGET_RW				LWR_TARGET_HW											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.134.3 Fields

Field	Function
31-12 LWR_TARGET_RW	Lower target address bits—programmable Forms bits [31:12] of the of the new address of the translated region.
11-0 LWR_TARGET_HW	Lower target address bits—hardwired Forms bits [11:0] of the start address of the new address of the translated region. The start address must be aligned to a 4 KB boundary (in address match mode); and to the BAR size boundary (in BAR match mode) so these bits are always 0. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PEX controller.

19.4.3.135 iATU Outbound Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)

19.4.3.135.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0	918h

19.4.3.135.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_TARGET_RW				LWR_TARGET_HW											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.135.3 Fields

Field	Function
31-12 LWR_TARGET_RW	Lower target address bits—programmable Forms bits [31:12] of the of the new address of the translated region.
11-0 LWR_TARGET_HW	Lower target address bits—hardwired Forms bits [11:0] of the start address of the new address of the translated region. The start address must be aligned to a 4 KB boundary, so these bits are always 0. A write to this location is ignored by the PEX controller.

19.4.3.136 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0)

19.4.3.136.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0	91Ch

19.4.3.136.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.136.3 Fields

Field	Function
31-0 UPPER_TARGET_RW	Upper target address bits Forms bits [63:32] of the start address of the new address of the translated region. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect.

19.4.3.137 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)

19.4.3.137.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0	91Ch

19.4.3.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.3.137.3 Fields

Field	Function
31-0 UPPER_TARGET_RW	Upper target address bits Forms bits [63:32] of the start address of the new address of the translated region.

19.4.3.138 Base Address Register 0 Mask (BAR0_MASK)

19.4.3.138.1 Offset

Register	Offset
BAR0_MASK	1010h

19.4.3.138.2 Function

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.138.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.138.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.3.139 Base Address Register 1 Mask (BAR1_MASK)

19.4.3.139.1 Offset

Register	Offset
BAR1_MASK	1014h

19.4.3.139.2 Function

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.139.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.139.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.3.140 Base Address Register 2 Mask (BAR2_MASK)

19.4.3.140.1 Offset

Register	Offset
BAR2_MASK	1018h

19.4.3.140.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.140.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.140.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.3.141 Base Address Register 3 Mask (BAR3_MASK)

19.4.3.141.1 Offset

Register	Offset
BAR3_MASK	101Ch

19.4.3.141.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.141.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

19.4.3.141.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.3.142 Base Address Register 4 Mask (BAR4_MASK)

19.4.3.142.1 Offset

Register	Offset
BAR4_MASK	1020h

19.4.3.142.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.142.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.142.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.3.143 Base Address Register 5 Mask (BAR5_MASK)

19.4.3.143.1 Offset

Register	Offset
BAR5_MASK	1024h

19.4.3.143.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.143.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

19.4.3.143.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.3.144 Expansion ROM Base Address Register Mask (EP mode) (EXP_ROM_BAR_MASK_EP)

19.4.3.144.1 Offset

Register	Offset
EXP_ROM_BAR_MASK_EP	1030h

19.4.3.144.2 Function

NOTE

This register is supported only in EP mode; the Expansion ROM Base Address Register Mask for RC mode is located at offset 0x1038.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express Expansion ROM Base Address Register (EP-Mode) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.144.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.144.4 Fields

Field	Function
31-1 MASK	Expansion ROM BAR Mask

Table continues on the next page...

Field	Function
0 BAR_EN	Expansion ROM BAR enable. 0b - Disable Expansion ROM BAR 1b - Enable Expansion ROM BAR

19.4.3.145 Expansion ROM Base Address Register Mask (RC mode) (EXP_ROM_BAR_MASK_RC)

19.4.3.145.1 Offset

Register	Offset
EXP_ROM_BAR_MASK_RC	1038h

19.4.3.145.2 Function

NOTE

This register is supported only in RC mode; the Expansion ROM Base Address Register Mask for EP mode is located at offset 0x1030.

The Expansion ROM Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express Expansion ROM Base Address Register (RC-Mode) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.3.145.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK					Reserved										BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.3.145.4 Fields

Field	Function
31-11 MASK	Expansion ROM BAR Mask
10-1 —	Reserved
0 BAR_EN	Expansion ROM BAR enable. 0b - Disable Expansion ROM BAR 1b - Enable Expansion ROM BAR

19.4.4 Non-SR-IOV Instances (PEX2 and PEX3) register descriptions

This section describes the registers for the non-SR-IOV instances of the PEX controller (PEX2 and PEX3); see SR-IOV Instance (PEX1) register descriptions for the descriptions of the registers for the SR-IOV instance (PEX1).

19.4.4.1 PCI_Express_Configuration_Registers memory map

PEX2 base address: 350_0000h

PEX3 base address: 360_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	PCI Express Vendor ID Register (Vendor_ID_Register)	16	RO	1957h
2h	PCI Express Device ID Register (Device_ID_Register)	16	RO	80C0h
4h	PCI Express Command Register (Command_Register)	16	RW	0000h
6h	PCI Express Status Register (Status_Register)	16	W1C	0010h
8h	PCI Express Revision ID Register (Revision_ID_Register)	8	RO	10h
9h	PCI Express Class Code Register (programming interface) (Class_Code_Register_a)	8	RO	00h
Ah	PCI Express Class Code Register (sub class) (Class_Code_Register_b)	8	RO	20h
Bh	PCI Express Class Code Register (base class) (Class_Code_Register_c)	8	RO	0Bh
Ch	PCI Express Cache Line Size Register (Cache_Line_Size_Register)	8	RW	00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
Dh	PCI Express Latency Timer Register (Latency_Timer_Register)	8	RO	00h
Eh	PCI Express Header Type Register (Header_Type_Register)	8	RO	00h
10h	PCI Express Base Address Register 0 (BAR0)	32	RW	0000_0000h
14h	PCI Express Base Address Register 1 (BAR1)	32	RW	0000_0000h
18h	PCI Express Base Address Register 2 (BAR2)	32	RW	0000_000Ch
18h	PCI Express Primary Bus Number Register (Primary_Bus_Number_Register)	8	RW	00h
19h	PCI Express Secondary Bus Number Register (Secondary_Bus_Number_Register)	8	RW	00h
1Ah	PCI Express Subordinate Bus Number Register (Subordinate_Bus_Number_Register)	8	RW	00h
1Ch	PCI Express Base Address Register 3 (BAR3)	32	RW	0000_0000h
1Ch	PCI Express I/O Base Register (IO_Base_Register)	8	RW	01h
1Dh	PCI Express I/O Limit Register (IO_Limit_Register)	8	RW	01h
1Eh	PCI Express Secondary Status Register (Secondary_Status_Register)	16	W1C	0000h
20h	PCI Express Base Address Register 4 (BAR4)	32	RW	0000_000Ch
20h	PCI Express Memory Base Register (Memory_Base_Register)	16	RW	0000h
22h	PCI Express Memory Limit Register (Memory_Limit_Register)	16	RW	0000h
24h	PCI Express Base Address Register 5 (BAR5)	32	RW	0000_0000h
24h	PCI Express Prefetchable Memory Base Register (Prefetchable_Memory_Base_Register)	16	RW	0001h
26h	PCI Express Prefetchable Memory Limit Register (Prefetchable_Memory_Limit_Register)	16	RW	0001h
28h	PCI Express Prefetchable Base Upper 32 Bits Register (Prefetchable_Base_Upper_32_Bits_Register)	32	RW	0000_0000h
2Ch	PCI Express Prefetchable Limit Upper 32 Bits Register (Prefetchable_Limit_Upper_32_Bits_Register)	32	RW	0000_0000h
2Ch	PCI Express Subsystem Vendor ID Register (Subsystem_Vendor_ID_Register)	16	RO	0000h
2Eh	PCI Express Subsystem ID Register (Subsystem_ID_Register)	16	RO	0000h
30h	PCI Express Expansion ROM Base Address Register (EP-Mode) (Expansion_ROM_BAR_Type0)	32	RW	0000_0000h
30h	PCI Express I/O Base Upper 16 Bits Register (IO_Base_Upper_16_Bits_Register)	16	RO	0000h
32h	PCI Express I/O Limit Upper 16 Bits Register (IO_Limit_Upper_16_Bits_Register)	16	RO	0000h
34h	Capabilities Pointer Register (Capabilities_Pointer_Register)	8	RO	40h
38h	PCI Express Expansion ROM Base Address Register (RC-Mode) (Expansion_ROM_BAR_Type1)	32	RW	0000_0000h
3Ch	PCI Express Interrupt Line Register (Interrupt_Line_Register)	8	RW	FFh
3Dh	PCI Express Interrupt Pin Register (Interrupt_Pin_Register)	8	RO	01h
3Eh	PCI Express Bridge Control Register (Bridge_Control_Register)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3Eh	PCI Express Minimum Grant Register (Minimum_Grant_Register)	8	RO	00h
3Fh	PCI Express Maximum Latency Register (Maximum_Latency_Register)	8	RO	00h
40h	PCI Express Power Management Capability ID Register (Power_Management_Capability_ID_Register)	8	RO	01h
42h	PCI Express Power Management Capabilities Register (Power_Management_Capabilities_Register)	16	RO	7E23h
44h	PCI Express Power Management Status and Control Register (Power_Management_Status_and_Control_Register)	16	RW	0000h
47h	PCI Express Power Management Data Register (Power_Management_Data_Register)	8	RO	00h
50h	PCI Express MSI Message Capability ID Register (MSI_Message_Capability_ID_Register)	8	RO	05h
52h	PCI Express MSI Message Control Register (MSI_Message_Control_Register)	16	RW	0088h
54h	PCI Express MSI Message Address Register (MSI_Message_Address_Register)	32	RW	0000_0000h
58h	PCI Express MSI Message Upper Address Register (MSI_Message_Upper_Address_Register)	32	RW	0000_0000h
5Ch	PCI Express MSI Message Data Register (MSI_Message_Data_Register)	16	RW	0000h
70h	PCI Express Capability ID Register (Capability_ID_Register)	8	RO	10h
72h	PCI Express Capabilities Register (Capabilities_Register)	16	RO	0002h
74h	PCI Express Device Capabilities Register (Device_Capabilities_Register)	32	RO	0000_8001h
78h	PCI Express Device Control Register (Device_Control_Register)	16	RW	2810h
7Ah	PCI Express Device Status Register (Device_Status_Register)	16	W1C	0000h
7Ch	PCI Express Link Capabilities Register (Link_Capabilities_Register)	32	RO	0043_F443h
80h	PCI Express Link Control Register (Link_Control_Register)	16	RW	0008h
82h	PCI Express Link Status Register (Link_Status_Register)	16	RO	1000h
84h	PCI Express Slot Capabilities Register (Slot_Capabilities_Register)	32	RO	0000_0000h
88h	PCI Express Slot Control Register (Slot_Control_Register)	16	RW	03C0h
8Ah	PCI Express Slot Status Register (Slot_Status_Register)	16	W1C	0008h
8Ch	PCI Express Root Control Register (Root_Control_Register)	16	RW	0000h
8Eh	PCI Express Root Capabilities Register (Root_Capabilities_Register)	16	RW	0000h
90h	PCI Express Root Status Register (Root_Status_Register)	32	RW	0000_0000h
94h	PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)	32	RO	0000_001Fh
98h	PCI Express Device Control 2 Register (Device_Control_2_Register)	16	RW	0000h
9Ch	PCI Express Link Capabilities 2 Register (Link_Capabilities_2_Register)	32	RO	0000_000Eh
A0h	PCI Express Link Control 2 Register (Link_Control_2_Register)	16	RW	0003h
A2h	PCI Express Link Status 2 Register (Link_Status_2_Register)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
100h	PCI Express Advanced Error Reporting Capability ID Register (Advanced_Error_Reporting_Capability_ID_Register)	16	RO	0001h
104h	PCI Express Uncorrectable Error Status Register (Uncorrectable_Error_Status_Register)	32	W1C	0000_0000h
108h	PCI Express Uncorrectable Error Mask Register (Uncorrectable_Error_Mask_Register)	32	RW	0000_0000h
10Ch	PCI Express Uncorrectable Error Severity Register (Uncorrectable_Error_Severity_Register)	32	RW	0046_2030h
110h	PCI Express Correctable Error Status Register (Correctable_Error_Status_Register)	32	W1C	0000_0000h
114h	PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register)	32	RW	0000_2000h
118h	PCI Express Advanced Error Capabilities and Control Register (Advanced_Error_Capabilities_and_Control_Register)	32	RW	0000_00A0h
11Ch	PCI Express Header Log Register 1 (Header_Log_Register_DWORD1)	32	RO	0000_0000h
120h	PCI Express Header Log Register 2 (Header_Log_Register_DWORD2)	32	RO	0000_0000h
124h	PCI Express Header Log Register 3 (Header_Log_Register_DWORD3)	32	RO	0000_0000h
128h	PCI Express Header Log Register 4 (Header_Log_Register_DWORD4)	32	RO	0000_0000h
12Ch	PCI Express Root Error Command Register (Root_Error_Command_Register)	32	RW	0000_0000h
130h	PCI Express Root Error Status Register (Root_Error_Status_Register)	32	W1C	0000_0000h
134h	PCI Express Correctable Error Source ID Register (Correctable_Error_Source_ID_Register)	16	RO	0000h
136h	PCI Express Error Source ID Register (Error_Source_ID_Register)	16	RO	0000h
148h	Secondary PCI Express Extended Capability Header (SPCIE_CAP_HEADER_REG)	32	RO	0001_0019h
14Ch	Link Control 3 Register (LINK_CONTROL3_REG)	32	RW	0000_0000h
150h	Lane Error Status Register (LANE_ERR_STATUS_REG)	32	W1C	0000_0000h
154h - 156h	Lane Equalization Control Register (LANE0_EQUALIZATION_CONTROL - LANE1_EQUALIZATION_CONTROL)	16	RO	7F7Fh
71Ch	Symbol Timer Register and Filter Mask 1 Register (SYMBOL_TIMER_FILTER_1_OFF)	32	RW	0000_0280h
890h	Gen3 Control Register (GEN3_RELATED_OFF)	32	RW	0080_0001h
8BCh	DBI Read-only Write Enable Register (MISC_CONTROL_1_OFF)	32	RW	0000_0000h
8E0h	Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)	32	RW	0000_0000h
8E4h	Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)	32	RW	0000_0000h
8E8h	Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)	32	RU	0000_0000h
900h	iATU Index Register (IATU_VIEWPORT_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
904h	iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_INBOUND_0)	32	RW	0000_0000h
904h	iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)	32	RW	0000_0000h
908h	iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)	32	RW	0000_0000h
908h	iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)	32	RW	0000_0000h
90Ch	iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
90Ch	iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
910h	iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
910h	iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
914h	iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_INBOUND_0)	32	RW	0000_0FFFh
914h	iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0FFFh
918h	iATU Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
918h	iATU Outbound Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
91Ch	iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
91Ch	iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
1010h	Base Address Register 0 Mask (BAR0_MASK)	32	WO	00FF_FFFFh
1014h	Base Address Register 1 Mask (BAR1_MASK)	32	WO	03FF_FFFFh
1018h	Base Address Register 2 Mask (BAR2_MASK)	32	WO	FFFF_FFFFh
101Ch	Base Address Register 3 Mask (BAR3_MASK)	32	WO	0000_00FFh
1020h	Base Address Register 4 Mask (BAR4_MASK)	32	WO	FFFF_FFFFh
1024h	Base Address Register 5 Mask (BAR5_MASK)	32	WO	0000_00FFh
1030h	Expansion ROM Base Address Register Mask (EP mode) (EXP_ROM_BAR_MASK_EP)	32	WO	00FF_FFFFh
1038h	Expansion ROM Base Address Register Mask (RC mode) (EXP_ROM_BAR_MASK_RC)	32	WO	00FF_FFFFh

19.4.4.2 PCI Express Vendor ID Register (Vendor_ID_Register)

19.4.4.2.1 Offset

Register	Offset
Vendor_ID_Register	0h

19.4.4.2.2 Function

The vendor ID register is used to identify the manufacturer of the device.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.4.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Vendor_ID															
W																
Reset	0	0	0	1	1	0	0	1	0	1	0	1	0	1	1	1

19.4.4.2.4 Fields

Field	Function
15-0 Vendor_ID	Vendor ID 0x1957 (NXP)

19.4.4.3 PCI Express Device ID Register (Device_ID_Register)

19.4.4.3.1 Offset

Register	Offset
Device_ID_Register	2h

19.4.4.3.2 Function

The device ID register is used to identify the device.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.4.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Device_ID															
W																
Reset	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

19.4.4.3.4 Fields

Field	Function
15-0 Device_ID	Device ID 1000000011000000b - LS1088A with security 1000000011000001b - LS1088A without security 1000000011000010b - LS1084A with security 1000000011000011b - LS1084A without security 1000000011100000b - LS1048A with security 1000000011100001b - LS1048A without security 1000000011100010b - LS1044A with security 1000000011100011b - LS1044A without security

19.4.4.4 PCI Express Command Register (Command_Register)

19.4.4.4.1 Offset

Register	Offset
Command_Register	4h

19.4.4.4.2 Function

The command register provides control over the ability to generate and respond to PCI Express cycles.

19.4.4.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					Interrupt_Disable	Reserved	SERR	Reserved	Parity_error_response	Reserved			Bus_master	Memory_space	IO_space
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.4.4 Fields

Field	Function
15-11 —	Reserved
10 Interrupt_Disable	Interrupt disable Controls the ability to generate INTx interrupt messages. Any INTx emulation interrupts already asserted by this device must be deasserted when this bit is set. 0b - Enables INTx interrupt messages 1b - Disables INTx interrupt messages
9 —	Reserved
8 SERR	SERR# enable Controls the reporting of fatal and non-fatal errors detected by the device to the Root Complex. **NOTE:** The error control and status bits in the command and status registers control PCI-compatible error reporting. PCI Express advanced error reporting is controlled by the PCI Express device control register described in PCI Express Device Control Register (Device_Control_Register) and the advanced error reporting registers (offsets 100h through 137h). 0b - Disables reporting 1b - Enables reporting
7 —	Reserved
6 Parity_error_response	Parity error response Controls whether this PCI Express controller responds to parity errors. **NOTE:** The error control and status bits in the command and status registers control PCI-compatible error reporting. PCI Express advanced error reporting is controlled by the PCI Express device control register described in PCI Express Device Control Register (Device_Control_Register) and the advanced error reporting registers (offsets 100h through 137h). 0b - Parity errors are ignored and normal operation continues. 1b - Parity errors cause the appropriate bit in the PCI Express status register to be set. However, note that errors are reported based on the values set in the PCI Express error enable and detection registers.

Table continues on the next page...

Field	Function
5-3 —	Reserved
2 Bus_master	Bus master enable Indicates whether this PCI Express device is configured as a master. EP mode: Clearing this bit prevent the device from issuing any memory or I/O transactions. Because MSI interrupts are effectively memory writes, clearing this bit also disables the ability of the device to issue MSI interrupts. RC mode: Clearing this bit disables the ability of the device to forward memory transactions upstream. This causes any inbound memory transaction to be treated as an unsupported request. 0b - Disables the ability to generate PCI Express accesses 1b - Enables this PCI Express controller to behave as a PCI Express bus master
1 Memory_space	Memory space enable Controls whether this PCI Express device (as a target) responds to memory accesses. EP mode: Clearing this bit prevents the device from accepting any memory transaction. RC mode: This bit is ignored. It does not affect outbound memory transaction 0b - This PCI Express device does not respond to PCI Express memory space accesses. 1b - This PCI Express device responds to PCI Express memory space accesses.
0 IO_space	I/O space enable EP mode: Clearing this bit prevents the device from accepting any IO transaction. Note that this bit is a don't care in EP mode since the device does not support IO transaction. RC mode: This bit is ignored. It does not affect outbound IO transaction. 0b - This PCI Express device (as a target) does not respond to PCI Express I/O space accesses. 1b - This PCI Express device (as a target) does respond to PCI Express I/O space accesses.

19.4.4.5 PCI Express Status Register (Status_Register)

19.4.4.5.1 Offset

Register	Offset
Status_Register	6h

19.4.4.5.2 Function

The status register is used to record status information for PCI Express related events.

19.4.4.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Detected_parity_error	Signaled_system_error	Received_master_abort	Received_target_abort	Signaled_target_abort	Reserved		Master_data_parity_error_detected	Reserved			Capabilities_List	Interrupt_Status	Reserved		
W	W1C	W1C	W1C	W1C	W1C			W1C				W1C	W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

19.4.4.5.4 Fields

Field	Function
15 Detected_parity_error	Detected parity error Set whenever a device receives a poisoned TLP regardless of the state of bit 6 in the command register.[1]
14 Signaled_system_error	Signaled system error Set whenever a device sends a ERR_FATAL or ERR_NONFATAL message and the SERR enable bit in the command register is set.[1]
13 Received_master_abort	Received master abort Set whenever a requestor receives a completion with unsupported request completion status.[1]
12 Received_target_abort	Received target abort Set whenever a device receives a completion with completer abort completion status.[1]
11 Signaled_target_abort	Signaled target abort Set whenever a device completes a request using completer abort completion status.[1]
10-9 —	Reserved
8 Master_data_parity_error_detected	Master data parity error Set by the requestor (primary side for Type1 headers) when either the requestor receives a completion marked poisoned or the requestor poisons a write request. Note that the parity error enable bit (bit 6) in the command register must be set for this bit to be set.[1]

Table continues on the next page...

Field	Function
7-5 —	Reserved
4 Capabilities_List	Capabilities list All PCI Express devices are required to implement the PCI Express capability structure.
3 Interrupt_Status	Interrupt status Set when an INTx interrupt message is pending internally to the device. Note that this bit is associated with INTx messages and not Message Signaled Interrupts.
2-0 —	Reserved

1. The error control and status bits in the command and status registers control PCI-compatible error reporting. PCI Express advanced error reporting is controlled by the PCI Express device control register described in PCI Express Device Control Register (Device_Control_Register) and the advanced error reporting capability structure starting at offset 100h.

19.4.4.6 PCI Express Revision ID Register (Revision_ID_Register)

19.4.4.6.1 Offset

Register	Offset
Revision_ID_Register	8h

19.4.4.6.2 Function

The revision ID register is used to identify the revision of the device.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.4.6.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Revision_ID							
W								
Reset	0	0	0	1	0	0	0	0

19.4.4.6.4 Fields

Field	Function
7-0 Revision_ID	Revision ID Revision specific.

19.4.4.7 PCI Express Class Code Register (programming interface) (Class_Code_Register_a)

19.4.4.7.1 Offset

Register	Offset
Class_Code_Register_a	9h

19.4.4.7.2 Function

The class code register is comprised of three single-byte fields—base class (offset 0x0B), sub-class (offset 0x0A), and programming interface (offset 0x09)—that indicate the basic functionality of the function.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.4.7.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Programming_Interface							
W								
Reset	0	0	0	0	0	0	0	0

19.4.4.7.4 Fields

Field	Function
7-0 Programming_Interface	Programming_Interface

19.4.4.8 PCI Express Class Code Register (sub class) (Class_Code_Register_b)

19.4.4.8.1 Offset

Register	Offset
Class_Code_Register_b	Ah

19.4.4.8.2 Function

The class code register is comprised of three single-byte fields—base class (offset 0x0B), sub-class (offset 0x0A), and programming interface (offset 0x09)—that indicate the basic functionality of the function.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.4.8.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Sub_Class							
W								
Reset	0	0	1	0	0	0	0	0

19.4.4.8.4 Fields

Field	Function
7-0 Sub_Class	Sub-Class

19.4.4.9 PCI Express Class Code Register (base class) (Class_Code_Register_c)

19.4.4.9.1 Offset

Register	Offset
Class_Code_Register_c	Bh

19.4.4.9.2 Function

The class code register is comprised of three single-byte fields—base class (offset 0x0B), sub-class (offset 0x0A), and programming interface (offset 0x09)—that indicate the basic functionality of the function.

NOTE

This register is writeable using internal accesses, but is read-only from inbound configuration accesses by an external host.

19.4.4.9.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Base_Class							
W								
Reset	0	0	0	0	1	0	1	1

19.4.4.9.4 Fields

Field	Function
7-0 Base_Class	Base Class 0x0B-Processor

19.4.4.10 PCI Express Cache Line Size Register (Cache_Line_Size_Register)

19.4.4.10.1 Offset

Register	Offset
Cache_Line_Size_Register	Ch

19.4.4.10.2 Function

The cache line size register is provided for legacy compatibility purposes (PCI 2.3); it is not used for PCI Express device functionality.

19.4.4.10.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Cache_Line_Size							
Reset	0	0	0	0	0	0	0	0

19.4.4.10.4 Fields

Field	Function
7-0 Cache_Line_Size	Cache Line Size Represents the cache line size of the processor in terms of 32-bit words (8 32-bit words = 32 bytes). Note that for PCI Express operation this register is ignored.

19.4.4.11 PCI Express Latency Timer Register (Latency_Timer_Register)

19.4.4.11.1 Offset

Register	Offset
Latency_Timer_Register	Dh

19.4.4.11.2 Function

The latency timer register is provided for legacy compatibility purposes (PCI 2.3); it is not used for PCI Express device functionality.

19.4.4.11.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Latency_Timer							
W								
Reset	0	0	0	0	0	0	0	0

19.4.4.11.4 Fields

Field	Function
7-0 Latency_Timer	Latency_Timer Note that for PCI Express operation this register is ignored.

19.4.4.12 PCI Express Header Type Register (Header_Type_Register)

19.4.4.12.1 Offset

Register	Offset
Header_Type_Register	Eh

19.4.4.12.2 Function

The PCI Express header type register is used to identify the layout of the PCI compatible header. The reset value depends on the RC/EP mode configured at power-on reset.

19.4.4.12.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Multifunction	Header_Layout						
W								
Reset	0	0[1]	0	0	0	0	0	0

1. EP=00h; RC=01h

19.4.4.12.4 Fields

Field	Function
7 Multifunction	Multifunction Identifies whether a device supports multiple functions 0b - Single function device 1b - Multiple function device
6-0 Header_Layout	Header Layout All other encodings reserved. 0000000b - Endpoint - Type 0 layout. 0000001b - Root Complex - Type 1 layout.

19.4.4.13 PCI Express Base Address Register 0 (BAR0)

19.4.4.13.1 Offset

Register	Offset
BAR0	10h

19.4.4.13.2 Function

The PCI Express base address registers (BARs) point to the beginning of distinct address ranges which the device should claim.

19.4.4.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.13.4 Fields

Field	Function
31-12 ADDRESS	Base address Indicates the base address of the inbound memory window 0. The default size is 16 MB.
11-4 —	Reserved
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.4.14 PCI Express Base Address Register 1 (BAR1)

19.4.4.14.1 Offset

Register	Offset
BAR1	14h

19.4.4.14.2 Function

This register is present only in the Type 0 Header (EP mode).

Base address register 1 at offset 0x14 is used to define the inbound memory window in the 32-bit memory space.

19.4.4.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.14.4 Fields

Field	Function
31-12 ADDRESS	Base address Indicates the base address where the inbound memory window 1 begins. The default size of this window is 64 MB.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.4.15 PCI Express Base Address Register 2 (BAR2)

19.4.4.15.1 Offset

Register	Offset
BAR2	18h

19.4.4.15.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 2 (BAR2) and base address register 3 (BAR3) define a 64-bit inbound memory window. BAR2 defines the lower portion of the memory window; BAR3 defines the upper portion of the memory window.

19.4.4.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

19.4.4.15.4 Fields

Field	Function
31-12 ADDRESS	Base address (lower portion) Indicates the lower portion of the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space. 10b - Locate anywhere in 64-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.4.16 PCI Express Primary Bus Number Register (Primary_Bus_Number_Register)

19.4.4.16.1 Offset

Register	Offset
Primary_Bus_Number_Register	18h

19.4.4.16.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.16.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Primary_Bus_Number							
Reset	0	0	0	0	0	0	0	0

19.4.4.16.4 Fields

Field	Function
7-0 Primary_Bus_Number	Primary Bus Number Bus that is connected to the upstream interface. Note that this register is programmed during system enumeration; in RC mode this register should remain 0x00.

19.4.4.17 PCI Express Secondary Bus Number Register (Secondary_Bus_Number_Register)

19.4.4.17.1 Offset

Register	Offset
Secondary_Bus_Number_Register	19h

19.4.4.17.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.17.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Secondary_Bus_Number							
Reset	0	0	0	0	0	0	0	0

19.4.4.17.4 Fields

Field	Function
7-0 Secondary_Bus_Number	Secondary Bus Number Bus that is directly connected to the downstream interface. Note that this register is programmed during system enumeration; in RC mode, this register is typically programmed to 0x01.

19.4.4.18 PCI Express Subordinate Bus Number Register (Subordinate_Bus_Number_Register)

19.4.4.18.1 Offset

Register	Offset
Subordinate_Bus_Number_Register	1Ah

19.4.4.18.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.18.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Subordinate_Bus_Number							
Reset	0	0	0	0	0	0	0	0

19.4.4.18.4 Fields

Field	Function
7-0 Subordinate_Bus_Number	Subordinate Bus Number Highest bus number that is on the downstream interface.

19.4.4.19 PCI Express Base Address Register 3 (BAR3)

19.4.4.19.1 Offset

Register	Offset
BAR3	1Ch

19.4.4.19.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 2 (BAR2) and base address register 3 (BAR3) define a 64-bit inbound memory window. BAR2 defines the lower portion of the memory window; BAR3 defines the upper portion of the memory window.

19.4.4.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.19.4 Fields

Field	Function
31-0 ADDRESS	Base address (upper portion) Indicates the upper portion of the base address where the inbound memory window begins. If no access to local memory is to be permitted by external requesters, then all bits are programmed.

19.4.4.20 PCI Express I/O Base Register (IO_Base_Register)

19.4.4.20.1 Offset

Register	Offset
IO_Base_Register	1Ch

19.4.4.20.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.4.20.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	IO_Start_Address				Address_Decode_Type			
W								
Reset	0	0	0	0	0	0	0	1

19.4.4.20.4 Fields

Field	Function
7-4	I/O Start Address

Table continues on the next page...

Field	Function
IO_Start_Address	Specifies bits 15:12 of the I/O space start address
3-0 Address_Decode_Type	Address Decode Type Specifies the number of I/O address bits. All other settings are reserved. 0000b - 16-bit I/O address decode 0001b - 32-bit I/O address decode

19.4.4.21 PCI Express I/O Limit Register (IO_Limit_Register)

19.4.4.21.1 Offset

Register	Offset
IO_Limit_Register	1Dh

19.4.4.21.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.4.21.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	IO_Limit_Address				Address_Decode_Type			
W								
Reset	0	0	0	0	0	0	0	1

19.4.4.21.4 Fields

Field	Function
7-4 IO_Limit_Address	I/O Limit Address Specifies bits 15:12 of the I/O space ending address
3-0 Address_Decode_Type	Address Decode Type Specifies the number of I/O address bits. All other settings are reserved. 0000b - 16-bit I/O address decode 0001b - 32-bit I/O address decode

19.4.4.22 PCI Express Secondary Status Register (Secondary_Status_Register)

19.4.4.22.1 Offset

Register	Offset
Secondary_Status_Register	1Eh

19.4.4.22.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.22.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DPE	SSE	RMA	RTA	STA	Reserved		MDPE	Reserved							
W	W1C	W1C	W1C	W1C	W1C			W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.22.4 Fields

Field	Function
15	Detected parity error

Table continues on the next page...

Field	Function
DPE	This bit is set whenever the secondary side receives a poisoned TLP regardless of the state of the parity error response bit.
14 SSE	Signaled system error This bit is set when a device sends a ERR_FATAL or ERR_NONFATAL message, provided the SERR enable bit in the command register is set to enable reporting.
13 RMA	Received master abort This bit is set when the secondary side receives an unsupported request (UR) completion.
12 RTA	Received target abort This bit is set when the secondary side receives a completer abort (CA) completion.
11 STA	Signaled target abort This bit is set when the secondary side issues a CA completion.
10-9 —	Reserved
8 MDPE	Master data parity error This bit is set when the parity error response bit is set and the secondary side requester receives a poisoned completion or poisons a write request. If the parity error response bit is cleared, this bit is never set.
7-0 —	Reserved

19.4.4.23 PCI Express Base Address Register 4 (BAR4)

19.4.4.23.1 Offset

Register	Offset
BAR4	20h

19.4.4.23.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 4 (BAR4) and base address register 5 (BAR5) define a 64-bit inbound memory window. BAR4 defines the lower portion of the memory window; BAR5 defines the upper portion of the memory window.

19.4.4.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS				Reserved								PREF	TYPE		MemSp
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

19.4.4.23.4 Fields

Field	Function
31-12 ADDRESS	Base address (lower portion) Indicates the lower portion of the base address where the inbound memory window begins.
11-4 —	Reserved. The device allows a 4 Kbyte window minimum.
3 PREF	Prefetchable
2-1 TYPE	Type 00b - Locate anywhere in 32-bit address space. 10b - Locate anywhere in 64-bit address space.
0 MemSp	Memory space indicator Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0.

19.4.4.24 PCI Express Memory Base Register (Memory_Base_Register)

19.4.4.24.1 Offset

Register	Offset
Memory_Base_Register	20h

19.4.4.24.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.24.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Memory_Base												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.24.4 Fields

Field	Function
15-4 Memory_Base	Memory base address Specifies bits 31:20 of the non-prefetchable memory space start address. Typically used for specifying memory-mapped I/O space. **NOTE:** Inbound posted transactions hitting into the mem base/limit range are ignored; inbound non-posted transactions hitting into the mem base/limit range results in an unsupported request response.
3-0 —	Reserved

19.4.4.25 PCI Express Memory Limit Register (Memory_Limit_Register)

19.4.4.25.1 Offset

Register	Offset
Memory_Limit_Register	22h

19.4.4.25.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.25.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Memory_Limit												Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.25.4 Fields

Field	Function
15-4 Memory_Limit	Memory limit address Specifies bits 31:20 of the non-prefetchable memory space ending address. Typically used for specifying memory-mapped I/O space. **NOTE:** Inbound posted transactions hitting into the mem base/limit range are ignored; inbound non-posted transactions hitting into the mem base/limit range results in unsupported request response.
3-0 —	Reserved

19.4.4.26 PCI Express Base Address Register 5 (BAR5)

19.4.4.26.1 Offset

Register	Offset
BAR5	24h

19.4.4.26.2 Function

This register is present only in the Type 0 Header (EP mode).

Together, base address register 4 (BAR4) and base address register 5 (BAR5) define a 64-bit inbound memory window. BAR4 defines the lower portion of the memory window; BAR5 defines the upper portion of the memory window.

19.4.4.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.26.4 Fields

Field	Function
31-0 ADDRESS	Base address (upper portion) Indicates the upper portion of the base address where the inbound memory window begins. If no access to local memory is to be permitted by external requestors, then all bits are programmed.

19.4.4.27 PCI Express Prefetchable Memory Base Register (Prefetchable_Memory_Base_Register)

19.4.4.27.1 Offset

Register	Offset
Prefetchable_Memory_Base_Register	24h

19.4.4.27.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.27.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PF_Memory_Base												Address_Decode_Type			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.4.27.4 Fields

Field	Function
15-4 PF_Memory_Base	Prefetchable memory base address Specifies bits 31:20 of the prefetchable memory space start address.
3-0 Address_Decode_Type	Address Decode Type Specifies the number of prefetchable memory address bits. All other settings reserved. 0000b - 32-bit memory address decode 0001b - 64-bit memory address decode

19.4.4.28 PCI Express Prefetchable Memory Limit Register (Prefetchable_Memory_Limit_Register)

19.4.4.28.1 Offset

Register	Offset
Prefetchable_Memory_Limit_Register	26h

19.4.4.28.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.28.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PF_Memory_Limit												Address_Decode_Type			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.4.28.4 Fields

Field	Function
15-4 PF_Memory_Limit	Prefetchable memory limit address Specifies bits 31:20 of the prefetchable memory space ending address.
3-0 Address_Decode_Type	Address decode type Specifies the number of prefetchable memory address bits. All other settings reserved. 0000b - 32-bit memory address decode 0001b - 64-bit memory address decode

19.4.4.29 PCI Express Prefetchable Base Upper 32 Bits Register (Prefetchable_Base_Upper_32_Bits_Register)

19.4.4.29.1 Offset

Register	Offset
Prefetchable_Base_Upper_32_Bits_Register	28h

19.4.4.29.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PF_Base_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PF_Base_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.29.4 Fields

Field	Function
31-0 PF_Base_Upper_32_Bits	Prefetchable memory base address (upper portion) Specifies bits 64:32 of the prefetchable memory space start address when the address decode type field in the prefetchable memory base register is 0x01.

19.4.4.30 PCI Express Prefetchable Limit Upper 32 Bits Register (Prefetchable_Limit_Upper_32_Bits_Register)

19.4.4.30.1 Offset

Register	Offset
Prefetchable_Limit_Upper_32_Bits_Register	2Ch

19.4.4.30.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PF_Limit_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PF_Limit_Upper_32_Bits															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.30.4 Fields

Field	Function
31-0 PF_Limit_Upper_32_Bits	Prefetchable memory limit address (upper portion) Specifies bits 64-32 of the prefetchable memory space ending address when the address decode type field in the prefetchable memory limit register is 0x01.

19.4.4.31 PCI Express Subsystem Vendor ID Register (Subsystem_Vendor_ID_Register)

19.4.4.31.1 Offset

Register	Offset
Subsystem_Vendor_ID_Register	2Ch

19.4.4.31.2 Function

This register is present only in the Type 0 Header (EP mode).

The PCI Express subsystem vendor ID register is used to identify the subsystem.

19.4.4.31.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Subsystem_Vendor_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.31.4 Fields

Field	Function
15-0 Subsystem_Vendor_ID	Subsystem Vendor ID This register is present only in the Type 0 Header (EP mode). The PCI Express subsystem vendor ID register is used to identify the subsystem vendor.

19.4.4.32 PCI Express Subsystem ID Register (Subsystem_ID_Register)

19.4.4.32.1 Offset

Register	Offset
Subsystem_ID_Register	2Eh

19.4.4.32.2 Function

This register is present only in the Type 0 Header (EP mode).

The PCI Express subsystem ID register is used to identify the subsystem.

19.4.4.32.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Subsystem_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.32.4 Fields

Field	Function
15-0 Subsystem_ID	Subsystem ID

19.4.4.33 PCI Express Expansion ROM Base Address Register (EP-Mode) (Expansion_ROM_BAR_Type0)

19.4.4.33.1 Offset

Register	Offset
Expansion_ROM_BAR_Type0	30h

19.4.4.33.2 Function

The Expansion ROM Base Address register is located at offset 0x30 in the Type 0 Header (EP mode); it is located at offset 0x38 in the Type 1 Header (RC mode).

19.4.4.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ROM_Base_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROM_Base_Address					Reserved										ROMBAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.33.4 Fields

Field	Function
31-11 ROM_Base_Address	Expansion ROM base address Specifies bits 31:11 of the non-prefetchable expansion ROM space start address. Typically used for specifying memory-mapped I/O space. The default size is 16M.
10-1 —	Reserved
0 ROMBAR_EN	Expansion ROM enable This bit controls whether or not the device accepts accesses to its expansion ROM 0b - The expansion ROM address space is disabled. 1b - Address decoding is enabled.

19.4.4.34 PCI Express I/O Base Upper 16 Bits Register (IO_Base_Upper_16_Bits_Register)

19.4.4.34.1 Offset

Register	Offset
IO_Base_Upper_16_Bits_Register	30h

19.4.4.34.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.4.34.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IO_Base_Upper_16_Bits															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.34.4 Fields

Field	Function
15-0 IO_Base_Upper_16_Bits	I/O base address (upper portion) Specifies bits 31-16 of the I/O space start address when the address decode type field in the I/O base register is 0x01.

19.4.4.35 PCI Express I/O Limit Upper 16 Bits Register (IO_Limit_Upper_16_Bits_Register)

19.4.4.35.1 Offset

Register	Offset
IO_Limit_Upper_16_Bits_Register	32h

19.4.4.35.2 Function

This register is present only in the Type 1 Header (RC mode).

Note that this device does not support inbound I/O transactions.

19.4.4.35.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IO_Limit_Upper_16_Bits															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.35.4 Fields

Field	Function
15-0 IO_Limit_Upper_16_Bits	I/O limit address (upper portion) Specifies bits 31-16 of the I/O space ending address when the address decode type field in the I/O limit register is 0x01.

19.4.4.36 Capabilities Pointer Register (Capabilities_Pointer_Register)

19.4.4.36.1 Offset

Register	Offset
Capabilities_Pointer_Register	34h

19.4.4.36.2 Function

The capabilities pointer identifies additional functionality supported by the device.

19.4.4.36.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Capabilities_Pointer							
W								
Reset	0	1	0	0	0	0	0	0

19.4.4.36.4 Fields

Field	Function
7-0 Capabilities_Pointer	Capabilities Pointer The capabilities pointer provides the offset for additional PCI-compatible registers above the common 64-byte header.

19.4.4.37 PCI Express Expansion ROM Base Address Register (RC-Mode) (Expansion_ROM_BAR_Type1)

19.4.4.37.1 Offset

Register	Offset
Expansion_ROM_BAR_Type1	38h

19.4.4.37.2 Function

The Expansion ROM Base Address register is located at offset 0x38 in the Type 1 Header (RC mode); it is located at offset 0x30 in the Type 0 Header (EP mode).

19.4.4.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ROM_Base_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROM_Base_Address					Reserved										ROMBAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.37.4 Fields

Field	Function
31-11 ROM_Base_Address	Expansion ROM base address Specifies bits 31:11 of the non-prefetchable expansion ROM space start address. Typically used for specifying memory-mapped I/O space. The default size is 16M.
10-1 —	Reserved
0 ROMBAR_EN	Expansion ROM enable This bit controls whether or not the device accepts accesses to its expansion ROM 0b - The expansion ROM address space is disabled. 1b - Address decoding is enabled.

19.4.4.38 PCI Express Interrupt Line Register (Interrupt_Line_Register)

19.4.4.38.1 Offset

Register	Offset
Interrupt_Line_Register	3Ch

19.4.4.38.2 Function

The interrupt line register is used by device drivers and OS software to communicate interrupt line routing information. Values in this register are programmed by system software and are system specific.

19.4.4.38.3 Diagram

Bits	7	6	5	4	3	2	1	0
R W	Interrupt_Line							
Reset	1	1	1	1	1	1	1	1

19.4.4.38.4 Fields

Field	Function
7-0 Interrupt_Line	Interrupt line Used to communicate interrupt line routing information.

19.4.4.39 PCI Express Interrupt Pin Register (Interrupt_Pin_Register)

19.4.4.39.1 Offset

Register	Offset
Interrupt_Pin_Register	3Dh

19.4.4.39.2 Function

The interrupt pin register identifies the legacy interrupt (INTx) messages the device (or function) uses.

19.4.4.39.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	Interrupt_pin							
W								
Reset	0	0	0	0	0	0	0	1

19.4.4.39.4 Fields

Field	Function
7-0 Interrupt_pin	Interrupt pin Legacy INTx message used by this device. All other settings reserved. 00000000b - This device does not use legacy interrupt (INTx) messages. 00000001b - INTA

19.4.4.40 PCI Express Bridge Control Register (Bridge_Control_Register)

19.4.4.40.1 Offset

Register	Offset
Bridge_Control_Register	3Eh

19.4.4.40.2 Function

This register is present only in the Type 1 Header (RC mode).

19.4.4.40.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									Scnd_RST	Reserved		VGA_EN	ISA_EN	SERR_EN	PER
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.40.4 Fields

Field	Function
15-7 —	Reserved
6 Scnd_RST	Secondary bus reset
5-4 —	Reserved
3 VGA_EN	VGA enable
2 ISA_EN	ISA enable
1 SERR_EN	SERR enable This bit controls the propagation of ERR_COR, ERR_NONFATAL, and ERR_FATAL responses received on the secondary side.
0 PER	Parity error response This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register.

19.4.4.41 PCI Express Minimum Grant Register (Minimum_Grant_Register)

19.4.4.41.1 Offset

Register	Offset
Minimum_Grant_Register	3Eh

19.4.4.41.2 Function

This register is present only in the Type 0 Header (EP mode).

This register does not apply to PCI Express. It is present for legacy purposes.

19.4.4.41.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MIN_GNT							
W								
Reset	0	0	0	0	0	0	0	0

19.4.4.41.4 Fields

Field	Function
7-0 MIN_GNT	Minimum grant Does not apply for PCI Express.

19.4.4.42 PCI Express Maximum Latency Register (Maximum_Latency_Register)

19.4.4.42.1 Offset

Register	Offset
Maximum_Latency_Register	3Fh

19.4.4.42.2 Function

This register is present only in the Type 0 Header (EP mode).

This register does not apply to PCI Express. It is present for legacy purposes.

19.4.4.42.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MAX_LAT							
W								
Reset	0	0	0	0	0	0	0	0

19.4.4.42.4 Fields

Field	Function
7-0 MAX_LAT	Maximum latency Does not apply for PCI Express.

19.4.4.43 PCI Express Power Management Capability ID Register (Power_Management_Capability_ID_Register)

19.4.4.43.1 Offset

Register	Offset
Power_Management_Capability_ID_Register	40h

19.4.4.43.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Power_Mgmt_Capability_ID							
W								
Reset	0	0	0	0	0	0	0	1

19.4.4.43.3 Fields

Field	Function
7-0 Power_Mgmt_Capability_ID	CAP_ID Power Management = 0x01

19.4.4.44 PCI Express Power Management Capabilities Register (Power_Management_Capabilities_Register)

19.4.4.44.1 Offset

Register	Offset
Power_Management_Capabilities_Register	42h

19.4.4.44.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PME_Support					D2	D1	AUX_Curr			DSI	Reserved	PME_CLK	Version		
W																
Reset	0	1	1	1	1	1	1	0	0	0	1	0	0	0	1	1

19.4.4.44.3 Fields

Field	Function
15-11 PME_Support	PME support Indicates the power states that this device supports
10 D2	D2 support
9 D1	D1 support
8-6 AUX_Curr	AUX Current
5 DSI	Device Specific Initialization
4 —	Reserved
3 PME_CLK	PME clock Does not apply to PCI Express.
2-0 Version	Version

19.4.4.45 PCI Express Power Management Status and Control Register (Power_Management_Status_and_Control_Register)

19.4.4.45.1 Offset

Register	Offset
Power_Management_Status_and_Control_Register	44h

19.4.4.45.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PME_STAT	Data_Scale		Data_Select				PME_EN	Reserved						Power_State	
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.45.3 Fields

Field	Function
15 PME_STAT	PME Status
14-13 Data_Scale	Data Scale Obtained directly from the PCI Express base specification.
12-9 Data_Select	Data Select Obtained directly from the PCI Express base specification.
8 PME_EN	PME_En PME Enable. **NOTE:** Bitfield access is sticky.
7-2 —	Reserved
1-0 Power_State	Power State Indicates the current power state of the function. 00b - D0 01b - D1

Field	Function
	10b - D2 11b - D3

19.4.4.46 PCI Express Power Management Data Register (Power_Management_Data_Register)

19.4.4.46.1 Offset

Register	Offset
Power_Management_Data_Register	47h

19.4.4.46.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	Data							
W								
Reset	0	0	0	0	0	0	0	0

19.4.4.46.3 Fields

Field	Function
7-0 Data	Data Obtained from the PCI Express base specification.

19.4.4.47 PCI Express MSI Message Capability ID Register (MSI_Message_Capability_ID_Register)

19.4.4.47.1 Offset

Register	Offset
MSI_Message_Capability_ID_Register	50h

19.4.4.47.2 Function

This register is supported only for EP mode.

19.4.4.47.3 Diagram

Bits	7	6	5	4	3	2	1	0
R	MSI_Message_Capability_ID							
W								
Reset	0	0	0	0	0	1	0	1

19.4.4.47.4 Fields

Field	Function
7-0 MSI_Message_Capability_ID	CAP_ID MSI Message = 0x05

19.4.4.48 PCI Express MSI Message Control Register (MSI_Message_Control_Register)

19.4.4.48.1 Offset

Register	Offset
MSI_Message_Control_Register	52h

19.4.4.48.2 Function

This register is supported only for EP mode.

19.4.4.48.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AC_64	MME			MMC			MSIE
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

19.4.4.48.4 Fields

Field	Function
15-8 —	Reserved
7 AC_64	64-bit address capable
6-4 MME	Multiple message enable
3-1 MMC	Multiple message capable
0 MSIE	MSI enable

19.4.4.49 PCI Express MSI Message Address Register (MSI_Message_Address_Register)

19.4.4.49.1 Offset

Register	Offset
MSI_Message_Address_Register	54h

19.4.4.49.2 Function

This register is supported only for EP mode.

19.4.4.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Message_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Message_Address														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.49.4 Fields

Field	Function
31-2 Message_Address	Message Address System-specified message address
1-0 —	Reserved. Always returns 00 on reads; write operations have no effect.

19.4.4.50 PCI Express MSI Message Upper Address Register (MSI_Message_Upper_Address_Register)

19.4.4.50.1 Offset

Register	Offset
MSI_Message_Upper_Address_Register	58h

19.4.4.50.2 Function

This register is supported only for EP mode.

19.4.4.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Message_Upper_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Message_Upper_Address															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.50.4 Fields

Field	Function
31-0 Message_Upper_Address	Message Address (upper portion) System-specified message upper address

19.4.4.51 PCI Express MSI Message Data Register (MSI_Message_Data_Register)

19.4.4.51.1 Offset

Register	Offset
MSI_Message_Data_Register	5Ch

19.4.4.51.2 Function

This register is supported only for EP mode.

19.4.4.51.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Message_Data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.51.4 Fields

Field	Function
15-0 Message_Data	Data System-specified message data.

19.4.4.52 PCI Express Capability ID Register (Capability_ID_Register)

19.4.4.52.1 Offset

Register	Offset
Capability_ID_Register	70h

19.4.4.52.2 Diagram

Bits	7	6	5	4	3	2	1	0
R	PCI_Express_Capability_ID							
W								
Reset	0	0	0	1	0	0	0	0

19.4.4.52.3 Fields

Field	Function
7-0 PCI_Express_Capability_ID	Capability ID PCI Express = 0x10

19.4.4.53 PCI Express Capabilities Register (Capabilities_Register)

19.4.4.53.1 Offset

Register	Offset
Capabilities_Register	72h

19.4.4.53.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		Interrupt_Message_Number					Slot	Device_Port_Type				Capability_Version			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

19.4.4.53.3 Fields

Field	Function
15-14 —	Reserved
13-9 Interrupt_Message_Number	Interrupt Message Number

Table continues on the next page...

Field	Function
	If this function is allocated more than one MSI interrupt number, then this register is required to contain the offset between the base Message Data and the MSI Message that is generated when any of the status bits in either the Slot Status register or the Root Port Status register, of this capability structure, are set.
8 Slot	Slot Implemented (RC mode only)
7-4 Device_Port_Type	Device/Port Type 0000b - (EP mode) 0100b - (RC mode)
3-0 Capability_Version	Capability Version Indicates the defined PCI Express capability structure version number.

19.4.4.54 PCI Express Device Capabilities Register (Device_Capabilities_Register)

19.4.4.54.1 Offset

Register	Offset
Device_Capabilities_Register	74h

19.4.4.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			FLRC	CSPLS		CSPLV								Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RBER	Reserved			EP_L1_LAT			EP_L0s_LAT			ET	PHAN_FCT		MAX_PL_SIZE_SUP		
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.4.54.3 Fields

Field	Function
31-29 —	Reserved
28 FLRC	Function Level Reset Capability For RC mode and non-SR-IOV capable EP mode, the reset value for this bit is 0.
27-26 CSPLS	Captured Slot Power Limit Scale
25-18 CSPLV	Captured Slot Power Limit Value For RC mode, the reset value for this field is 00h; for EP mode, the value is determined by received Set_Slot_Power_Limit messages.
17-16 —	Reserved
15 RBER	Role-Based Error Reporting
14-12 —	Reserved. System software must ignore the value read from these bits. System software is permitted to write any value to these bits.
11-9 EP_L1_LAT	Endpoint L1 Acceptable Latency
8-6 EP_L0s_LAT	Endpoint L0s Acceptable Latency
5 ET	Extended Tag Field Supported
4-3 PHAN_FCT	Phantom Functions Supported
2-0 MAX_PL_SIZE_SUP	Max_Payload_Size Supported Maximum payload size supported. 001 = 256-bytes

19.4.4.55 PCI Express Device Control Register (Device_Control_Register)

19.4.4.55.1 Offset

Register	Offset
Device_Control_Register	78h

19.4.4.55.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IFLR	MAX_READ_SIZE			ENS	APE	PFE	ETE	MAX_PAYLOAD_SIZE			RO	URR	FER	NFER	CER
Reset	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0

19.4.4.55.3 Fields

Field	Function
15 IFLR	Initiate Function Level Reset
14-12 MAX_READ_SIZE	Maximum_Read_Request_Size
11 ENS	Enable No Snoop
10 APE	AUX Power PM Enable
9 PFE	Phantom Functions Enable
8 ETE	Extended Tag Field Enable
7-5 MAX_PAYLOAD_SIZE	Max_Payload_Size Maximum payload size
4 RO	Enable Relaxed Ordering
3 URR	Unsupported Request Reporting Enable
2 FER	Fatal Error Reporting Enable
1 NFER	Non-Fatal Error Reporting Enable
0 CER	Correctable Error Reporting Enable

19.4.4.56 PCI Express Device Status Register (Device_Status_Register)

19.4.4.56.1 Offset

Register	Offset
Device_Status_Register	7Ah

19.4.4.56.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										TP	APD	URD	FED	NFED	CED
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.56.3 Fields

Field	Function
15-6 —	Reserved
5 TP	Transactions Pending
4 APD	AUX Power Detected
3 URD	Unsupported Request Detected
2 FED	Fatal Error Detected
1 NFED	Non-Fatal Error Detected
0 CED	Correctable Error Detected

19.4.4.57 PCI Express Link Capabilities Register (Link_Capabilities_Register)

19.4.4.57.1 Offset

Register	Offset
Link_Capabilities_Register	7Ch

19.4.4.57.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Port_Number								Reserved	AOC	LBWN	DLLARC	SD_ERR_RPT_CAP	CPM	L1_EX_LAT	
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	L1_EX_LAT	L0s_EX_LAT			ASPM		MAX_LINK_W						MAX_LINK_SP			
W																
Reset	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	1

19.4.4.57.3 Fields

Field	Function
31-24 Port_Number	Port Number This field indicates the PCI Express Port number for the given PCI Express Link
23 —	Reserved
22	ASPM Optionality Compliance

Table continues on the next page...

Field	Function
AOC	Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests.
21 LBWN	Link Bandwidth Notification Capability In EP-mode, this bit is hardwired to 0. In RC-mode it is hardwired to 1.
20 DLLARC	Data Link Layer Active Reporting Capable Set to 1 when in RC. Set to 0 when in EP.
19 SD_ERR_RPT_CAP	Surprise Down Error Reporting Capable
18 CPM	Clock Power Management
17-15 L1_EX_LAT	L1 Exit Latency
14-12 L0s_EX_LAT	L0s Exit Latency
11-10 ASPM	Active State Power Management (ASPM) Support
9-4 MAX_LINK_W	Maximum Link Width
3-0 MAX_LINK_SP	Maximum Link Speed 0001b - 2.5 GT/s link 0010b - 5.0 GT/s 0011b - 8.0 GT/s

19.4.4.58 PCI Express Link Control Register (Link_Control_Register)

19.4.4.58.1 Offset

Register	Offset
Link_Control_Register	80h

19.4.4.58.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				LABIE	LBMIE	HW_AUTO_WIDTH_DIS	ECPM	EXT_SYNC	CCC	RL	LD	RCB	Reserved	ASPM_CTL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

19.4.4.58.3 Fields

Field	Function
15-12 —	Reserved
11 LABIE	Link Autonomous Bandwidth Interrupt Enable
10 LBMIE	Link Bandwidth Management Interrupt Enable
9 HW_AUTO_WIDTH_DIS	Hardware Autonomous Width Disable
8 ECPM	Enable Clock Power Management
7 EXT_SYNC	Extended Synch
6 CCC	Common Clock Configuration
5 RL	Retrain Link (Reserved for EP devices). In RC mode, setting this bit initiates link retraining by directing the Physical Layer LTSSM to the Recovery state; reads of this bit always return 0.
4 LD	Link Disable (Reserved for EP devices)
3 RCB	Read Completion Boundary (RCB)
2 —	Reserved
1-0 ASPM_CTL	Active State Power Management (ASPM) Control

19.4.4.59 PCI Express Link Status Register (Link_Status_Register)

19.4.4.59.1 Offset

Register	Offset
Link_Status_Register	82h

19.4.4.59.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LABS	LBMS	Reserved	SCC	LT	Reserved	NEG_LINK_W						LINK_SP			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.59.3 Fields

Field	Function
15 LABS	Link Autonomous Bandwidth Status **NOTE:** This bit is write-1-clear in RC mode; it is read-only in EP mode
14 LBMS	Link Bandwidth Management Status **NOTE:** This bit is write-1-clear in RC mode; it is read-only in EP mode.
13 —	Reserved
12 SCC	Slot Clock Configuration
11 LT	Link Training
10 —	Reserved.
9-4 NEG_LINK_W	Negotiated link width All other encodings are reserved. The value in this field is undefined when the link is not up. 000001b - x1 000010b - x2

Table continues on the next page...

Field	Function
3-0 LINK_SP	Current Link Speed 0001b - 2.5 GT/s 0010b - 5.0 GT/s 0011b - 8.0 GT/s

19.4.4.60 PCI Express Slot Capabilities Register (Slot_Capabilities_Register)

19.4.4.60.1 Offset

Register	Offset
Slot_Capabilities_Register	84h

19.4.4.60.2 Function

This register is supported only for RC mode.

19.4.4.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Physical_Slot_Number													NOCMDCPLSUP	EMIP	SPLS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPLS	SPLV								HPD	HPS	PIP	AIP	MRLSP	PCP	ABP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.60.4 Fields

Field	Function
31-19 Physical_Slot_Number	Physical Slot Number This field indicates the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is globally unique within the chassis. This field must be initialized to 0 for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port.
18 NOCMDCPLSUP	No Command Completed Support
17 EMIP	Electromechanical Interlock Present
16-15 SPLS	Slot Power Limit Scale
14-7 SPLV	Slot Power Limit Value
6 HPD	Hot-Plug Capable **Note:** This chip does not support hot-plug capabilities.
5 HPS	Hot-Plug Surprise **Note:** This chip does not support hot-plug capabilities.
4 PIP	Power Indicator Present **Note:** This chip does not support hot-plug capabilities.
3 AIP	Attention Indicator Present **Note:** This chip does not support hot-plug capabilities.
2 MRLSP	MRL Sensor Present **Note:** This chip does not support hot-plug capabilities.
1 PCP	Power Controller Present **Note:** This chip does not support hot-plug capabilities.
0 ABP	Attention Button Present **Note:** This chip does not support hot-plug capabilities.

19.4.4.61 PCI Express Slot Control Register (Slot_Control_Register)

19.4.4.61.1 Offset

Register	Offset
Slot_Control_Register	88h

19.4.4.61.2 Function

This register is supported only for RC mode.

19.4.4.61.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			DLLSTCHGEN	EMICTL	PCC	PIC		AIC		HPIE	CCIE	PDCE	MRLSCE	PFDE	ABPE
Reset	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

19.4.4.61.4 Fields

Field	Function
15-13 —	Reserved
12 DLLSTCHGEN	Data Link Layer State Changed Enable **Note:** This chip does not support hot-plug capabilities.
11 EMICTL	Electromechanical Interlock Control **Note:** This chip does not support hot-plug capabilities.
10 PCC	Power Controller Control **Note:** This chip does not support hot-plug capabilities.
9-8 PIC	Power Indicator Control **Note:** This chip does not support hot-plug capabilities.
7-6 AIC	Attention Indicator Control **Note:** This chip does not support hot-plug capabilities.
5 HPIE	Hot-Plug Interrupt Enable **Note:** This chip does not support hot-plug capabilities.
4 CCIE	Command Completed Interrupt Enable **Note:** This chip does not support hot-plug capabilities.
3 PDCE	Presence Detect Changed Enable **Note:** This chip does not support hot-plug capabilities.
2 MRLSCE	MRL Sensor Changed Enable **Note:** This chip does not support hot-plug capabilities.
1 PFDE	Power Fault Detected Enable **Note:** This chip does not support hot-plug capabilities.
0 ABPE	Attention Button Pressed Enable **Note:** This chip does not support hot-plug capabilities.

19.4.4.62 PCI Express Slot Status Register (Slot_Status_Register)

19.4.4.62.1 Offset

Register	Offset
Slot_Status_Register	8Ah

19.4.4.62.2 Function

This register is supported only for RC mode.

19.4.4.62.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							DLLSTCHG	EM_IL_ST	PDS	MRLSS	CC	PDC	MRLSC	PFD	ABP
W								W1C				W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

19.4.4.62.4 Fields

Field	Function
15-9 —	Reserved
8 DLLSTCHG	Data Link Layer State Changed
7 EM_IL_ST	Electromechanical Interlock Status
6 PDS	Presence Detect State This bit indicates the presence of an adapter in the slot, reflected by the logical OR of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slotâs corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism.

Table continues on the next page...

Field	Function
	This register must be implemented on all Downstream Ports that implement slots. For Downstream Ports not connected to slots (where the Slot Implemented bit of the PCI Express Capabilities register is 0b), this bit must return 1b. Defined encodings are: 0b - Slot Empty 1b - Card Present in slot
5 MRLSS	MRL Sensor State 0b - MRL closed 1b - MRL open
4 CC	Command Completed
3 PDC	Presence Detect Changed
2 MRLSC	MRL Sensor Changed
1 PFD	Power Fault Detected
0 ABP	Attention Button Pressed

19.4.4.63 PCI Express Root Control Register (Root_Control_Register)

19.4.4.63.1 Offset

Register	Offset
Root_Control_Register	8Ch

19.4.4.63.2 Function

This register is supported only for RC mode.

19.4.4.63.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CRSSWVE	PMEIE	SEFEE	SENFEE	SECEE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.63.4 Fields

Field	Function
15-5 —	Reserved
4 CRSSWVE	CRS Software Visibility Enable
3 PMEIE	PME Interrupt Enable
2 SEFEE	System Error on Fatal Error Enable
1 SENFEE	System Error on Non-Fatal Error Enable
0 SECEE	System Error on Correctable Error Enable

19.4.4.64 PCI Express Root Capabilities Register (Root_Capabilities_Register)

19.4.4.64.1 Offset

Register	Offset
Root_Capabilities_Register	8Eh

19.4.4.64.2 Function

This register is supported only for RC mode.

19.4.4.64.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															CRSSWV
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.64.4 Fields

Field	Function
15-1 —	Reserved
0 CRSSWV	CRS Software Visibility

19.4.4.65 PCI Express Root Status Register (Root_Status_Register)

19.4.4.65.1 Offset

Register	Offset
Root_Status_Register	90h

19.4.4.65.2 Function

This register is supported only for RC mode.

19.4.4.65.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														PMEP	PMES
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PME_requester_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.65.4 Fields

Field	Function
31-18 —	Reserved
17 PMEP	PME Pending
16 PMES	PME Status
15-0 PME_requester_ID	PME Requester ID

19.4.4.66 PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)

19.4.4.66.1 Offset

Register	Offset
Device_Capabilities_2_Register	94h

19.4.4.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										ARI_FWD	CPL_TO_DS	CPL_TO_RS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

19.4.4.66.3 Fields

Field	Function
31-6 —	Reserved
5 ARI_FWD	ARI Forwarding Supported
4 CPL_TO_DS	Completion Timeout Disable Supported
3-0 CPL_TO_RS	Completion Timeout Ranges Supported

19.4.4.67 PCI Express Device Control 2 Register (Device_Control_2_Register)

19.4.4.67.1 Offset

Register	Offset
Device_Control_2_Register	98h

19.4.4.67.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						IDO_CPL_EN	IDO_REQ_EN	Reserved		ARIF E	CPL_TOD	CPL_TO_VAL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.67.3 Fields

Field	Function
15-10 —	Reserved
9	IDO Completion Enable

Table continues on the next page...

Field	Function
IDO_CPL_EN	
8 IDO_REQ_EN	IDO Request Enable
7-6 —	Reserved
5 ARIFE	ARI Forwarding Enable Must be set when RC is communicating with ARI devices. When set will allow RC to issue configuration type 0 cycles with device number != 0.
4 CPL_TOD	Completion Timeout Disable
3-0 CPL_TO_VAL	Completion Timeout Value

19.4.4.68 PCI Express Link Capabilities 2 Register (Link_Capabilities_2_Register)

19.4.4.68.1 Offset

Register	Offset
Link_Capabilities_2_Register	9Ch

19.4.4.68.2 Function

The PCI Express link capability 2 register is shown below.

19.4.4.68.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							Crosslink_Supported	Support_Link_Speed_Vector							Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

19.4.4.68.4 Fields

Field	Function
31-9 —	Reserved
8 Crosslink_Supported	Crosslink Supported
7-1 Support_Link_Speed_Vector	Supported Link Speeds Vector This field indicates the supported Link speed(s) of the associated Port. For each bit, a value of 1 indicates that the corresponding Link speed is supported; otherwise, the Link speed is not supported. Bit definitions are: Bit 1 2.5 GT/s Bit 2 5.0 GT/s Bit 3 8.0 GT/s Bits 7:4 Reserved
0 —	Reserved

19.4.4.69 PCI Express Link Control 2 Register (Link_Control_2_Register)

19.4.4.69.1 Offset

Register	Offset
Link_Control_2_Register	A0h

19.4.4.69.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CDE				CSOS	EMC	TxM			SDE	HWASD	EC	T_LS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

19.4.4.69.3 Fields

Field	Function
15-12 CDE	Compliance Preset/De-emphasis
11 CSOS	Compliance SOS
10 EMC	Enter Modified Compliance
9-7 TxM	Transmit Margin This bit is applicable only for the Gen2 speed.
6 SDE	Selectable De-emphasis
5 HWASD	Hardware Autonomous Speed Disable
4 EC	Enter Compliance
3-0 T_LS	Target Link Speed

19.4.4.70 PCI Express Link Status 2 Register (Link_Status_2_Register)

19.4.4.70.1 Offset

Register	Offset
Link_Status_2_Register	A2h

19.4.4.70.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										LER	EP3S	EP2S	EP1S	EC	DE_LVL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.70.3 Fields

Field	Function
15-6 —	Reserved
5 LER	Link Equalization Request
4 EP3S	Equalization Phase 3 Successful
3 EP2S	Equalization Phase 2 Successful
2 EP1S	Equalization Phase 1 Successful
1 EC	Equalization Complete
0 DE_LVL	Current De-emphasis Level

19.4.4.71 PCI Express Advanced Error Reporting Capability ID Register (Advanced_Error_Reporting_Capability_ID_Register)

19.4.4.71.1 Offset

Register	Offset
Advanced_Error_Reporting_Capability_ID_Register	100h

19.4.4.71.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Capability_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.4.71.3 Fields

Field	Function
15-0 Capability_ID	PCI Express Extended Capability ID 0x0001 = Advanced error reporting capability

19.4.4.72 PCI Express Uncorrectable Error Status Register (Uncorrectable_Error_Status_Register)

19.4.4.72.1 Offset

Register	Offset
Uncorrectable_Error_Status_Register	104h

19.4.4.72.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved											URE	ECRCE	MTLP	RXO	UC
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CA	CTO	FCPE	PTLP	Reserved							DLPE	Reserved			
W	W1C	W1C	W1C	W1C								W1C				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.72.3 Fields

Field	Function
31-21 —	Reserved
20 URE	Unsupported request error status.
19 ECRCE	ECRC error status.
18 MTLP	Malformed TLP status.
17 RXO	Receiver overflow status.
16 UC	Unexpected completion status.
15 CA	Completer abort status.
14 CTO	Completion timeout status. Note that a completion timeout error is a fatal error. If a completion timeout error is detected, the system has become unstable. Hot reset is recommended to restore stability of the system.
13 FCPE	Flow control protocol error status.
12 PTLP	Poisoned TLP status.
11-5 —	Reserved

Table continues on the next page...

Field	Function
4 DLPE	Data link protocol error status.
3-0 —	Reserved

19.4.4.73 PCI Express Uncorrectable Error Mask Register (Uncorrectable_Error_Mask_Register)

19.4.4.73.1 Offset

Register	Offset
Uncorrectable_Error_Mask_Register	108h

19.4.4.73.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved											UREM	ECRCEM	MTLPM	RXOM	UCM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAM	CTOM	FCPEM	PTLPM	Reserved							DLPEM	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.73.3 Fields

Field	Function
31-21 —	Reserved
20 UREM	Unsupported request error mask.

Table continues on the next page...

Field	Function
19 ECRCEM	ECRC error mask.
18 MTLPM	Malformed TLP mask.
17 RXOM	Receiver overflow mask.
16 UCM	Unexpected completion mask.
15 CAM	Completer abort mask.
14 CTOM	Completion timeout mask.
13 FCPEM	Flow control protocol error mask.
12 PTLPM	Poisoned TLP mask.
11-5 —	Reserved
4 DLPEM	Data link protocol error mask.
3-0 —	Reserved

19.4.4.74 PCI Express Uncorrectable Error Severity Register (Uncorrectable_Error_Severity_Register)

19.4.4.74.1 Offset

Register	Offset
Uncorrectable_Error_Severity_Register	10Ch

19.4.4.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved											URES	ECRCES	MTLPS	RXOS	UCS
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	CAS	CTOS	FCPES	PTLPS	Reserved							DLPES	Reserved			
Reset	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0

19.4.4.74.3 Fields

Field	Function
31-21 —	Reserved
20 URES	Unsupported request error severity.
19 ECRCES	ECRC error severity.
18 MTLPS	Malformed TLP severity.
17 RXOS	Receiver overflow severity.
16 UCS	Unexpected completion severity.
15 CAS	Completer abort severity.
14 CTOS	Completion timeout severity.
13 FCPES	Flow control protocol error severity.
12 PTLPS	Poisoned TLP severity.
11-5 —	Reserved
4	Data link protocol error severity.

Table continues on the next page...

Field	Function
DLPES	
3-0 —	Reserved

19.4.4.75 PCI Express Correctable Error Status Register (Correctable_Error_Status_Register)

19.4.4.75.1 Offset

Register	Offset
Correctable_Error_Status _Register	110h

19.4.4.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		ADVNFE	RTTO	Reserved			RNR	BDLLP	BTLP	Reserved					RXE
W			W1C	W1C				W1C	W1C	W1C						W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.75.3 Fields

Field	Function
31-14 —	Reserved
13 ADVNFE	Advisory Non-Fatal Error Status indicates the occurrence of the advisory error, and the Advisory Non-Fatal Error Mask bit in the PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register) is checked to determine whether to proceed further with logging and signaling

Table continues on the next page...

Field	Function
12 RTTO	Replay timer timeout status
11-9 —	Reserved
8 RNR	REPLAY_NUM Rollover status
7 BDLLP	Bad DLLP status
6 BTLP	Bad TLP status
5-1 —	Reserved
0 RXE	Receiver error status

19.4.4.76 PCI Express Correctable Error Mask Register (Correctable_Error_Mask_Register)

19.4.4.76.1 Offset

Register	Offset
Correctable_Error_Mask_Register	114h

19.4.4.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ADVNFEM	RTTOM	Reserved			RNRM	BDLLPM	BTLPM	Reserved					RXEM
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.76.3 Fields

Field	Function
31-14 —	Reserved
13 ADVNFEM	Advisory non-fatal error mask. This bit is Set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting
12 RTTOM	Replay timer timeout mask
11-9 —	Reserved
8 RNRM	REPLAY_NUM Rollover mask
7 BDLLPM	Bad DLLP mask
6 BTLPM	Bad TLP mask
5-1 —	Reserved
0 RXEM	Receiver error mask

19.4.4.77 PCI Express Advanced Error Capabilities and Control Register (Advanced_Error_Capabilities_and_Control_Register)

19.4.4.77.1 Offset

Register	Offset
Advanced_Error_Capabilities_and_Control_Register	118h

19.4.4.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							ECRCCE	ECRCCC	ECRCGE	ECRCGC	First_Error_Pointer				
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

19.4.4.77.3 Fields

Field	Function
31-9 —	Reserved
8 ECRCCE	ECRC checking enable.
7 ECRCCC	ECRC checking capable.
6 ECRCGE	ECRC generation enable.
5 ECRCGC	ECRC generation capable.
4-0 First_Error_Pointer	The First Error Pointer is a read-only register that identifies the bit position of the first error reported in the Uncorrectable Error Status register.

19.4.4.78 PCI Express Header Log Register 1 (Header_Log_Register_DWORD1)

19.4.4.78.1 Offset

Register	Offset
Header_Log_Register_DWORD1	11Ch

19.4.4.78.2 Function

The first DWORD of the PCI Express header log register is shown in the figure below.

19.4.4.78.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_0								Byte_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_2								Byte_3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.78.4 Fields

Field	Function
31-24 Byte_0	Byte n of the TLP header associated with the error.
23-16 Byte_1	Byte n of the TLP header associated with the error.
15-8 Byte_2	Byte n of the TLP header associated with the error.
7-0 Byte_3	Byte n of the TLP header associated with the error.

19.4.4.79 PCI Express Header Log Register 2 (Header_Log_Register_DWORD2)

19.4.4.79.1 Offset

Register	Offset
Header_Log_Register_DWORD2	120h

19.4.4.79.2 Function

The second DWORD of the PCI Express header log register is shown in the figure below.

19.4.4.79.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_4								Byte_5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_6								Byte_7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.79.4 Fields

Field	Function
31-24 Byte_4	Byte n of the TLP header associated with the error.
23-16 Byte_5	Byte n of the TLP header associated with the error.
15-8 Byte_6	Byte n of the TLP header associated with the error.
7-0 Byte_7	Byte n of the TLP header associated with the error.

19.4.4.80 PCI Express Header Log Register 3 (Header_Log_Register_DWORD3)

19.4.4.80.1 Offset

Register	Offset
Header_Log_Register_DWORD3	124h

19.4.4.80.2 Function

The third DWORD of the PCI Express header log register is shown in the figure below.

19.4.4.80.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_8								Byte_9							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_A								Byte_B							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.80.4 Fields

Field	Function
31-24 Byte_8	Byte n of the TLP header associated with the error.
23-16 Byte_9	Byte n of the TLP header associated with the error.
15-8 Byte_A	Byte n of the TLP header associated with the error.
7-0 Byte_B	Byte n of the TLP header associated with the error.

19.4.4.81 PCI Express Header Log Register 4 (Header_Log_Register_DWORD4)

19.4.4.81.1 Offset

Register	Offset
Header_Log_Register_DWORD4	128h

19.4.4.81.2 Function

The fourth DWORD of the PCI Express header log register is shown in the figure below.

19.4.4.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Byte_C								Byte_D							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Byte_E								Byte_F							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.81.4 Fields

Field	Function
31-24 Byte_C	Byte n of the TLP header associated with the error.
23-16 Byte_D	Byte n of the TLP header associated with the error.
15-8 Byte_E	Byte n of the TLP header associated with the error.
7-0 Byte_F	Byte n of the TLP header associated with the error.

19.4.4.82 PCI Express Root Error Command Register (Root_Error_Command_Register)

19.4.4.82.1 Offset

Register	Offset
Root_Error_Command_Register	12Ch

19.4.4.82.2 Function

This register is supported only for RC mode.

19.4.4.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													FERE	NFERE	CERE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.82.4 Fields

Field	Function
31-3 —	Reserved
2 FERE	Fatal error reporting enable.
1 NFERE	Non-fatal error reporting enable
0 CERE	Correctable error reporting enable

19.4.4.83 PCI Express Root Error Status Register (Root_Error_Status_Register)

19.4.4.83.1 Offset

Register	Offset
Root_Error_Status_Register	130h

19.4.4.83.2 Function

This register is supported only for RC mode.

19.4.4.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	AEIMN					Reserved										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved									FEMR	NFEMR	FUF	MEFNFR	EFNFR	MECR	ECR
W										W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.83.4 Fields

Field	Function
31-27 AEIMN	Advanced error interrupt message number.
26-7 —	Reserved
6 FEMR	Fatal error messages received.
5 NFEMR	Non-fatal error messages received.

Table continues on the next page...

Field	Function
4 FUF	First uncorrectable fatal.
3 MEFNFR	Multiple ERR_FATAL/NONFATAL received.
2 EFNFR	ERR_FATAL/NONFATAL received.
1 MECR	Multiple ERR_COR received.
0 ECR	ERR_COR received.

19.4.4.84 PCI Express Correctable Error Source ID Register (Correctable_Error_Source_ID_Register)

19.4.4.84.1 Offset

Register	Offset
Correctable_Error_Source_ID_Register	134h

19.4.4.84.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_COR_Source_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.84.3 Fields

Field	Function
15-0 ERR_COR_Source_ID	Loaded with the requester ID indicated in the received ERR_COR Message when the ERR_COR Received register is not already set. Default value of this field is 0.

19.4.4.85 PCI Express Error Source ID Register (Error_Source_ID_Register)

19.4.4.85.1 Offset

Register	Offset
Error_Source_ID_Register	136h

19.4.4.85.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Error_Source_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.85.3 Fields

Field	Function
15-0 Error_Source_ID	ERR_FATAL/NONFATAL source ID

19.4.4.86 Secondary PCI Express Extended Capability Header (SPCIE_CAP_HEADER_REG)

19.4.4.86.1 Offset

Register	Offset
SPCIE_CAP_HEADER_REG	148h

19.4.4.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NEXT_OFFSET												CAP_VERSION			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EXTENDED_CAP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1

19.4.4.86.3 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset This field is a pointer to the next capability structure. **NOTE:** The reset value of this field depends on the instantiation of the module. See PCI Express configuration registers for the specific values.
19-16 CAP_VERSION	Capability Version Indicates the version of the Capability structure present.
15-0 EXTENDED_CAP_ID	PCI Express Extended Capability ID 0019h = Secondary PCI Express extended capability

19.4.4.87 Link Control 3 Register (LINK_CONTROL3_REG)

19.4.4.87.1 Offset

Register	Offset
LINK_CONTROL3_REG	14Ch

19.4.4.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														EQ_REQ_INT_EN	PERFORM_EQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.87.3 Fields

Field	Function
31-2 —	Reserved
1 EQ_REQ_INT_EN	Link Equalization Request Interrupt Enable This bit is RW for Downstream Ports only.
0 PERFORM_EQ	Perform Equalization This bit is RW for Downstream Ports only.

19.4.4.88 Lane Error Status Register (LANE_ERR_STATUS_REG)

19.4.4.88.1 Offset

Register	Offset
LANE_ERR_STATUS_REG	150h

19.4.4.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														LANE_ERR_STATUS	
W															W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.88.3 Fields

Field	Function
31-2 —	Reserved
1-0 LANE_ERR_STATUS	Lane Error Status Each bit indicates if the corresponding Lane detected a Lane-based error. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Field supported in	Field not supported in
PEX2_LANE_ERR_STATUS_REG[0]	PEX2_LANE_ERR_STATUS_REG[1]
PEX3_LANE_ERR_STATUS_REG	—

19.4.4.89 Lane Equalization Control Register (LANE0_EQUALIZATION_CONTROL - LANE1_EQUALIZATION_CONTROL)

19.4.4.89.1 Offset

Register	Offset
LANE0_EQUALIZATION _CONTROL	154h
LANE1_EQUALIZATION _CONTROL	156h

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
PEX2_L ANE0_ EQUALI ZATI ON_C ONTR OL	PEX2_L ANE1_ EQUALI ZATI ON_C ONTR OL
PEX3_L ANE0_ EQUALI ZATI ON_C ONTR OL– LANE1_ EQUALI ZATI ON_C ONTR OL	—

19.4.4.89.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	USP_RX_PRESET_HINT			USP_TX_PRESET				Reserved	DSP_RX_PRESET_HINT			DSP_TX_PRESET			
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

19.4.4.89.3 Fields

Field	Function
15 —	Reserved
14-12 USP_RX_PRESET_HINT	Upstream Port Receiver Preset Hint
11-8 USP_TX_PRESET	Upstream Port Transmitter Preset
7 —	Reserved
6-4 DSP_RX_PRESET_HINT	Downstream Port Receiver Preset Hint
3-0 DSP_TX_PRESET	Downstream Port Transmitter Preset

19.4.4.90 Symbol Timer Register and Filter Mask 1 Register (SYMBOL_TIMER_FILTER_1_OFF)

19.4.4.90.1 Offset

Register	Offset
SYMBOL_TIMER_FILTER_1_OFF	71Ch

19.4.4.90.2 Function

The Filter Mask 1 Register modifies the receive filtering and error handling rules. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule

19.4.4.90.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASK_RADM_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DISABLE_FC_WD_TIMER	Reserved				SKP_INT_VAL										
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

19.4.4.90.4 Fields

Field	Function
31-16	Filter Mask 1
MASK_RADM_1	The Filter Mask 1 Register modifies the receive filtering and error handling rules. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. [31]: CX_FLT_MASK_RC_CFG_DISCARD 0: For RC filter to not allow CFG transaction being received 1: For RC filter to allow CFG transaction being received [30]: CX_FLT_MASK_RC_IO_DISCARD 0: For RC filter to not allow IO transaction being received 1: For RC filter to allow IO transaction being received

Table continues on the next page...

Field	Function
	[29]: CX_FLT_MASK_MSG_DROP 0: Drop MSG TLP (except for Vendor MSG). 1: Do not Drop MSG (except for Vendor MSG). Send message TLPs to your application. This bit only controls message TLPs other than Vendor MSGs. Vendor MSGs are controlled by Filter Mask Register 2, bits [1:0]. **NOTE:** The forwarding of message TLPs may be undesirable. If so, write a 0 to this bit to disable forwarding of message TLPs. [28]: CX_FLT_MASK_CPL_ECRC_DISCARD Only used when completion queue is advertized with infinite credits and is in store-and-forward mode. 0: Discard completions with ECRC errors 1: Allow completions with ECRC errors to be passed up Reserved field for SW. [27]: CX_FLT_MASK_ECRC_DISCARD 0: Discard TLPs with ECRC errors 1: Allow TLPs with ECRC errors to be passed up [26]: CX_FLT_MASK_CPL_LEN_MATCH 0: Enforce length match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err 1: MASK length match for completions [25]: CX_FLT_MASK_CPL_ATTR_MATCH 0: Enforce attribute match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask attribute match for completions [24]: CX_FLT_MASK_CPL_TC_MATCH 0: Enforce Traffic Class match for completions; a violation results in a malformed TLP error, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask Traffic Class match for completions [23]: CX_FLT_MASK_CPL_FUNC_MATCH 0: Enforce function match for completions; a violation results in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask function match for completions [22]: CX_FLT_MASK_CPL_REQID_MATCH 0: Enforce Req. Id match for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask Req. Id match for completions [21]: CX_FLT_MASK_CPL_TAGERR_MATCH 0: Enforce Tag Error Rules for completions; a violation result in cpl_abort, and possibly AER of unexp_cpl_err, cpl_rcvd_ur, cpl_rcvd_ca 1: Mask Tag Error Rules for completions [20]: CX_FLT_MASK_LOCKED_RD_AS_UR 0: Treat locked Read TLPs as UR for EP. Treat locked Read TLPs as supported for RC.

Table continues on the next page...

Field	Function
	1: Treat locked Read TLPs as Supported for EP. Treat locked Read TLPs as UR for RC. [19]: CX_FLT_MASK_CFG_TYPE1_RE_AS_UR 0: Treat CFG type1 TLPs as UR for EP. Treat CFG type1 TLPs as supported for RC. 1: Treat CFG type1 TLPs as Supported for EP. Treat CFG type1 TLPs as UR for RC. [18]: CX_FLT_MASK_UR_OUTSIDE_BAR 0: Treat out-of-bar TLPs as UR 1: Do not treat out-of-bar TLPs as UR [17]: CX_FLT_MASK_UR_POIS 0: Treat poisoned request TLPs as UR 1: Do not treat poisoned request TLPs as UR The native PEX controller always passes poisoned completions to your application. [16]: CX_FLT_MASK_UR_FUNC_MISMATCH 0: Treat Function MisMatched TLPs as UR 1: Do not treat Function MisMatched TLPs as UR *Note*: This register field is sticky.
15 DISABLE_FC_WD_TIMER	Disable FC Watchdog Timer Disable FC Watchdog Timer. *Note*: This register field is sticky.
14-11 —	Reserved
10-0 SKP_INT_VAL	SKP Interval Value The number of symbol times to wait between transmitting SKP ordered sets. Note that the PEX controller actually waits the number of symbol times in this register plus 1 between transmitting SKP ordered sets. Your application must program this register accordingly. For example, if 1536 were programmed into this register (in a 250 MHz PEX controller), then the PEX controller actually transmits SKP ordered sets once every 1537 symbol times. The value programmed to this register is actually clock ticks and not symbol times. In a 125 MHz PEX controller, programming the value programmed to this register should be scaled down by a factor of 2 (because one clock tick = two symbol times in this case). *Note:*This value is not used at Gen3 speed; the skip interval is hardcoded to 370 blocks. *Note*: This register field is sticky.

19.4.4.91 Gen3 Control Register (GEN3_RELATED_OFF)

19.4.4.91.1 Offset

Register	Offset
GEN3_RELATED_OFF	890h

19.4.4.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R / W	Reserved															GEN3_EQUALIZATION_DISABLE
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	Reserved			RXEQ_PH01_EN	Reserved	EQ_EIEOS_CNT	Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

19.4.4.91.3 Fields

Field	Function
31-17 —	Reserved
16 GEN3_EQUALIZATION_DISABLE	Equalization Disable Disable equalization feature. *Note*: This register field is sticky.
15-13 —	Reserved
12 RXEQ_PH01_EN	Rx Equalization Phase 0/Phase 1 Hold Enable When this bit is set the upstream port holds phase 0 (the downstream port holds phase 1) for 10ms. Holding phase 0 or phase 1 can be used to allow sufficient time for Rx Equalization to be performed by the PHY. *Note*: This register field is sticky.
11 —	Reserved
10	Equalization EIEOS Count Reset Disable

Table continues on the next page...

Field	Function
EQ_EIEOS_CNT	Disable requesting reset of EIEOS count during equalization. *Note*: This register field is sticky.
9-0 —	Reserved

19.4.4.92 DBI Read-only Write Enable Register (MISC_CONTROL_1_OFF)

19.4.4.92.1 Offset

Register	Offset
MISC_CONTROL_1_OFF	8BCh

19.4.4.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															RO_WR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.92.3 Fields

Field	Function
31-1 —	Reserved
0 RO_WR_EN	Read-only write enable This field enables writing (using internal accesses only) to some read-only and hardware-initialized bits in the configuration registers.

Field	Function
	0b - Read-only fields are read only. 1b - Enables writing to some read-only fields

19.4.4.93 Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)

19.4.4.93.1 Offset

Register	Offset
COHERENCY_CONTROL_1_OFF	8E0h

19.4.4.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_MEMTYPE_BOUNDARY_LOW_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_MEMTYPE_BOUNDARY_LOW_ADDR														Reserved	CFG_MEMTYPE_VALUE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.93.3 Fields

Field	Function
31-2 CFG_MEMTYPE_BOUNDARY_LOW_ADDR	Boundary lower address for memory type Bits [31:0] of dword-aligned address of the boundary for memory type. The two lower address least-significant bits must be 00. Addresses up to but not including this value are in the lower address space region; addresses equal or greater than this value are in the upper address space region. *Note*: This register field is sticky.
1 —	Reserved
0 CFG_MEMTYPE_VALUE	Memory type Sets the memory type for the lower and upper regions of the address space: *Note*: This register field is sticky. 0b - lower = CCSR; upper = Memory 1b - Reserved

19.4.4.94 Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)

19.4.4.94.1 Offset

Register	Offset
COHERENCY_CONTROL_2_OFF	8E4h

19.4.4.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_MEMTYPE_BOUNDARY_HIGH_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_MEMTYPE_BOUNDARY_HIGH_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.94.3 Fields

Field	Function
31-0 CFG_MEMTYPE_BOUNDARY_HIGH_ADDR	Boundary upper address for memory type Bits [63:32] of the 64-bit dword-aligned address of the boundary for memory type. *Note*: This register field is sticky.

19.4.4.95 Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)

19.4.4.95.1 Offset

Register	Offset
COHERENCY_CONTROL_3_OFF	8E8h

19.4.4.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.95.3 Fields

Field	Function
31-0 —	Reserved Must be 0000_0000h.

19.4.4.96 iATU Index Register (IATU_VIEWPORT_OFF)

19.4.4.96.1 Offset

Register	Offset
IATU_VIEWPORT_OFF	900h

19.4.4.96.2 Function

The iATU registers are programmed through an indirect addressing scheme (using this index register) to reduce the address footprint in the PCI Express extended configuration space.

The iATU can remap 6 outbound and 6 inbound address regions. This index register determines the memory region to be accessed.

19.4.4.96.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGION_DIR	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								REGION_INDEX							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.96.4 Fields

Field	Function
31 REGION_DIR	Region Direction Defines the region being accessed as either 0: Outbound 1: Inbound
30-8 —	Reserved

Table continues on the next page...

Field	Function
7-0 REGION_INDEX	Region Index Defines which region is being accessed when writing to the control, base, limit and target registers. Must not be set to a number greater than 5 when an outbound region is being accessed. Must not be set to a value greater than 5 when an inbound region is being accessed.

19.4.4.97 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_INBOUND_0)

19.4.4.97.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_0	904h

19.4.4.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							CTRL_1_FUNC_NUM					Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					Reserved		Reserved	Reserved			TYPE				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.97.3 Fields

Field	Function
31-25	Reserved

Table continues on the next page...

Field	Function
—	
24-20 CTRL_1_FUNC_NUM	Function Number. **MEM-I/O:** DM/EP/SW: When the Address and BAR matching logic in the PEX controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. **CFG0/CFG1:** DM/EP/SW: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. RC: Reserved. You must set this bit to "0"
19-18 —	Reserved
17-16 —	Reserved
15-11 —	Reserved
10-9 —	Reserved
8 —	Reserved
7-5 —	Reserved
4-0 TYPE	Type When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful).

19.4.4.98 iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)

19.4.4.98.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_0	904h

19.4.4.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved							Reserved					Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					ATTR		TD	TC			TYPE				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.98.3 Fields

Field	Function
31-25 —	Reserved
24-20 —	Reserved
19-18 —	Reserved
17-16 —	Reserved
15-11 —	Reserved
10-9 ATTR	Attribute (Attr) When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register.
8 TD	TLP Digest (TD) When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register.
7-5 TC	Traffic class (TC) When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register.
4-0 TYPE	Type When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register.

19.4.4.99 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)

19.4.4.99.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_0	908h

19.4.4.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	REGION_EN	MATCH_MODE	Reserved	Reserved	Reserved	Reserved	Reserved		Reserved		Reserved	Reserved	FUNC_NUM_MATCH_EN	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	Reserved	Reserved			BAR_NUM			Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.99.3 Fields

Field	Function
31 REGION_EN	Region Enable This bit must be set to "1" for address translation to take place.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup.

Table continues on the next page...

Field	Function
	1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. 1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header.
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25-24 —	Reserved
23-22 —	Reserved
21 —	Reserved
20 —	Reserved
19 FUNC_NUM_MATCH_EN	Function Number Match Enable **RC/SW:** Reserved. You must set this bit to "0". **DM/EP:** Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed.
18 —	Reserved
17 —	Reserved
16	Reserved

Table continues on the next page...

Field	Function
—	
15 —	Reserved
14 —	Reserved
13-11 —	Reserved
10-8 BAR_NUM	BAR Number **DM/EP/SW:** When the BAR number of an inbound MEM or IO TLP that is matched by the normal internal BAR address matching mechanism is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. 000b - BAR0 001b - BAR1 010b - BAR2 011b - BAR3 100b - BAR4 101b - BAR5 110b - ROM 111b - reserved IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b-101b and that BAR configured as an IO BAR. **RC:** Reserved. Must be 0.
7-0 —	Reserved

19.4.4.100 iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)

19.4.4.100.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_0	908h

19.4.4.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGION_EN	Reserved	Reserved	CFG_SHIFT_MODE	Reserved	Reserved							Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MSG_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.100.3 Fields

Field	Function
31 REGION_EN	Region Enable This bit must be set to "1" for address translation to take place.
30 —	Reserved
29 —	Reserved
28 CFG_SHIFT_MODE	CFG Shift Mode This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address.Shifts bits [27:12] of the untranslated address to form bits [31:16] of the translated address.
27 —	Reserved
26-20 —	Reserved
19 —	Reserved
18-8 —	Reserved
7-0 MSG_CODE	Message Code Message TLPs: When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register.

19.4.4.101 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)

19.4.4.101.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_0	90Ch

19.4.4.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_BASE_RW				LWR_BASE_HW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.101.3 Fields

Field	Function
31-12 LWR_BASE_RW	Lower base address bits—programmable Forms bits [31:12] of the start address of the address region to be translated.
11-0 LWR_BASE_HW	Lower base address bits—hardwired Forms bits [11:0] of the start address of the address region to be translated. The start address must be aligned to a 4 KB boundary, so these bits are always 0. A write to this location is ignored by the PEX controller.

19.4.4.102 iATU Lower Base Address Register (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)

19.4.4.102.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0	90Ch

19.4.4.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_BASE_RW				LWR_BASE_HW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.102.3 Fields

Field	Function
31-12 LWR_BASE_RW	Lower base address bits—programmable Forms bits [31:12] of the start address of the address region to be translated.
11-0 LWR_BASE_HW	Lower base address bits—hardwired Forms bits [11:0] of the start address of the address region to be translated. The start address must be aligned to a 4 KB boundary, so these bits are always 0. A write to this location is ignored by the PEX controller.

19.4.4.103 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)

19.4.4.103.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_0	910h

19.4.4.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.103.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Upper base address bits Forms bits [63:32] of the start (and end) address of the address region to be translated.

19.4.4.104 iATU Upper Base Address Register (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)

19.4.4.104.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0	910h

19.4.4.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.104.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Upper base address bits Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect.

19.4.4.105 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_INBOUND_0)

19.4.4.105.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_0	914h

19.4.4.105.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_RW				LIMIT_ADDR_HW											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.105.3 Fields

Field	Function
31-12 LIMIT_ADDR_RW	Limit address bits—programmable Forms bits [31:12] of the end address of the address region to be translated.
11-0 LIMIT_ADDR_HW	Limit address bits—hardwired Forms bits [11:0] of the end address of the address region to be translated. Address regions must be aligned to a 4 KB boundary, so these bits are always 1. A write to this location is ignored by the PEX controller.

19.4.4.106 iATU Limit Address Register (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)

19.4.4.106.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_0	914h

19.4.4.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_RW				LIMIT_ADDR_HW											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.106.3 Fields

Field	Function
31-12 LIMIT_ADDR_RW	Limit address bits—programmable Forms bits [31:12] of the end address of the address region to be translated.
11-0 LIMIT_ADDR_HW	Limit address bits—hardwired Forms bits [11:0] of the end address of the address region to be translated. Address regions must be aligned to a 4 KB boundary, so these bits are always 1. A write to this location is ignored by the PEX controller.

19.4.4.107 iATU Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)

19.4.4.107.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_0	918h

19.4.4.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_TARGET_RW				LWR_TARGET_HW											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.107.3 Fields

Field	Function
31-12 LWR_TARGET_RW	Lower target address bits—programmable Forms bits [31:12] of the of the new address of the translated region.
11-0 LWR_TARGET_HW	Lower target address bits—hardwired Forms bits [11:0] of the start address of the new address of the translated region. The start address must be aligned to a 4 KB boundary (in address match mode); and to the BAR size boundary (in BAR match mode) so these bits are always 0. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PEX controller.

19.4.4.108 iATU Outbound Region#N Lower Offset Address Register (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)

19.4.4.108.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0	918h

19.4.4.108.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_TARGET_RW				LWR_TARGET_HW											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.108.3 Fields

Field	Function
31-12 LWR_TARGET_RW	Lower target address bits—programmable Forms bits [31:12] of the of the new address of the translated region.
11-0 LWR_TARGET_HW	Lower target address bits—hardwired Forms bits [11:0] of the start address of the new address of the translated region. The start address must be aligned to a 4 KB boundary, so these bits are always 0. A write to this location is ignored by the PEX controller.

19.4.4.109 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0)

19.4.4.109.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_INBOUND_0	91Ch

19.4.4.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.109.3 Fields

Field	Function
31-0 UPPER_TARGET_RW	Upper target address bits Forms bits [63:32] of the start address of the new address of the translated region. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect.

19.4.4.110 iATU Upper Target Address Register (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)

19.4.4.110.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0	91Ch

19.4.4.110.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.4.4.110.3 Fields

Field	Function
31-0 UPPER_TARGET_RW	Upper target address bits Forms bits [63:32] of the start address of the new address of the translated region.

19.4.4.111 Base Address Register 0 Mask (BAR0_MASK)

19.4.4.111.1 Offset

Register	Offset
BAR0_MASK	1010h

19.4.4.111.2 Function

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.111.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.111.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.4.112 Base Address Register 1 Mask (BAR1_MASK)

19.4.4.112.1 Offset

Register	Offset
BAR1_MASK	1014h

19.4.4.112.2 Function

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.112.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.112.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.4.113 Base Address Register 2 Mask (BAR2_MASK)

19.4.4.113.1 Offset

Register	Offset
BAR2_MASK	1018h

19.4.4.113.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.113.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.113.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.4.114 Base Address Register 3 Mask (BAR3_MASK)

19.4.4.114.1 Offset

Register	Offset
BAR3_MASK	101Ch

19.4.4.114.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.114.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

19.4.4.114.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.4.115 Base Address Register 4 Mask (BAR4_MASK)

19.4.4.115.1 Offset

Register	Offset
BAR4_MASK	1020h

19.4.4.115.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.115.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.115.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.4.116 Base Address Register 5 Mask (BAR5_MASK)

19.4.4.116.1 Offset

Register	Offset
BAR5_MASK	1024h

19.4.4.116.2 Function

NOTE

This register is supported only in EP mode; it does not exist in RC mode.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express base address register (BAR) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.116.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

19.4.4.116.4 Fields

Field	Function
31-1 MASK	Mask
0 BAR_EN	BAR enable. 0b - Disable BAR 1b - Enable BAR

19.4.4.117 Expansion ROM Base Address Register Mask (EP mode) (EXP_ROM_BAR_MASK_EP)

19.4.4.117.1 Offset

Register	Offset
EXP_ROM_BAR_MASK_EP	1030h

19.4.4.117.2 Function

NOTE

This register is supported only in EP mode; the Expansion ROM Base Address Register Mask for RC mode is located at offset 0x1038.

The Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express Expansion ROM Base Address Register (EP-Mode) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.117.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK															BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.117.4 Fields

Field	Function
31-1 MASK	Expansion ROM BAR Mask

Table continues on the next page...

Field	Function
0 BAR_EN	Expansion ROM BAR enable. 0b - Disable Expansion ROM BAR 1b - Enable Expansion ROM BAR

19.4.4.118 Expansion ROM Base Address Register Mask (RC mode) (EXP_ROM_BAR_MASK_RC)

19.4.4.118.1 Offset

Register	Offset
EXP_ROM_BAR_MASK_RC	1038h

19.4.4.118.2 Function

NOTE

This register is supported only in RC mode; the Expansion ROM Base Address Register Mask for EP mode is located at offset 0x1030.

The Expansion ROM Base Address Register Mask is used to control size of memory requested by the corresponding PCI Express Expansion ROM Base Address Register (RC-Mode) during initialization. When a mask bit is set to 1, the corresponding address bit in the BAR is masked and cannot be written, returning a 0 when read.

19.4.4.118.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MASK															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	MASK					Reserved										BAR_EN
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.4.4.118.4 Fields

Field	Function
31-11 MASK	Expansion ROM BAR Mask
10-1 —	Reserved
0 BAR_EN	Expansion ROM BAR enable. 0b - Disable Expansion ROM BAR 1b - Enable Expansion ROM BAR

19.5 PEX_LUT memory map/registers overview

PEX_LUT provides registers for:

- mapping AMQ attributes to request IDs.
- generating INTx and PME messages
- controlling errors (disable, detect, and interrupt enable)
- accessing VF configuration registers (SR-IOV only)
- generating a soft-reset
- determining the current LTSSM state

19.5.1 SR-IOV PEX Look Up Table (PEX1_LUT) register descriptions

This section describes the registers for the SR-IOV instance of the PEX Look Up Table (PEX1_LUT); see Non-SR-IOV PEX Look Up Table (PEX2_LUT and PEX3_LUT) register descriptions for the descriptions of the registers for the non-SR-IOV instances (PEX2_LUT and PEX3_LUT).

19.5.1.1 PEX_LUT memory map

PEX1_LUT base address: 348_0000h

Offset	Register	Width (In bits)	Access	Reset value
20h	PEX LUT Status Register (PEXLSR)	32	W1C	0000_0000h
24h	PEX LUT Control Register (PEXLCR)	32	RW	0000_0000h
800h	PEX LUT Entry a Upper Data Register (PEXL0UDR)	32	RW	0000_FFFFh
804h	PEX LUT Entry a Lower Data Register (PEXL0LDR)	32	RW	8000_0000h
808h	PEX LUT Entry a Upper Data Register (PEXL1UDR)	32	RW	0000_0000h
80Ch	PEX LUT Entry a Lower Data Register (PEXL1LDR)	32	RW	0000_0000h
810h	PEX LUT Entry a Upper Data Register (PEXL2UDR)	32	RW	0000_0000h
814h	PEX LUT Entry a Lower Data Register (PEXL2LDR)	32	RW	0000_0000h
818h	PEX LUT Entry a Upper Data Register (PEXL3UDR)	32	RW	0000_0000h
81Ch	PEX LUT Entry a Lower Data Register (PEXL3LDR)	32	RW	0000_0000h
820h	PEX LUT Entry a Upper Data Register (PEXL4UDR)	32	RW	0000_0000h
824h	PEX LUT Entry a Lower Data Register (PEXL4LDR)	32	RW	0000_0000h
828h	PEX LUT Entry a Upper Data Register (PEXL5UDR)	32	RW	0000_0000h
82Ch	PEX LUT Entry a Lower Data Register (PEXL5LDR)	32	RW	0000_0000h
830h	PEX LUT Entry a Upper Data Register (PEXL6UDR)	32	RW	0000_0000h
834h	PEX LUT Entry a Lower Data Register (PEXL6LDR)	32	RW	0000_0000h
838h	PEX LUT Entry a Upper Data Register (PEXL7UDR)	32	RW	0000_0000h
83Ch	PEX LUT Entry a Lower Data Register (PEXL7LDR)	32	RW	0000_0000h
840h	PEX LUT Entry a Upper Data Register (PEXL8UDR)	32	RW	0000_0000h
844h	PEX LUT Entry a Lower Data Register (PEXL8LDR)	32	RW	0000_0000h
848h	PEX LUT Entry a Upper Data Register (PEXL9UDR)	32	RW	0000_0000h
84Ch	PEX LUT Entry a Lower Data Register (PEXL9LDR)	32	RW	0000_0000h
850h	PEX LUT Entry a Upper Data Register (PEXL10UDR)	32	RW	0000_0000h
854h	PEX LUT Entry a Lower Data Register (PEXL10LDR)	32	RW	0000_0000h
858h	PEX LUT Entry a Upper Data Register (PEXL11UDR)	32	RW	0000_0000h
85Ch	PEX LUT Entry a Lower Data Register (PEXL11LDR)	32	RW	0000_0000h
860h	PEX LUT Entry a Upper Data Register (PEXL12UDR)	32	RW	0000_0000h
864h	PEX LUT Entry a Lower Data Register (PEXL12LDR)	32	RW	0000_0000h
868h	PEX LUT Entry a Upper Data Register (PEXL13UDR)	32	RW	0000_0000h
86Ch	PEX LUT Entry a Lower Data Register (PEXL13LDR)	32	RW	0000_0000h
870h	PEX LUT Entry a Upper Data Register (PEXL14UDR)	32	RW	0000_0000h
874h	PEX LUT Entry a Lower Data Register (PEXL14LDR)	32	RW	0000_0000h
878h	PEX LUT Entry a Upper Data Register (PEXL15UDR)	32	RW	0000_0000h
87Ch	PEX LUT Entry a Lower Data Register (PEXL15LDR)	32	RW	0000_0000h
880h	PEX LUT Entry a Upper Data Register (PEXL16UDR)	32	RW	0000_0000h
884h	PEX LUT Entry a Lower Data Register (PEXL16LDR)	32	RW	0000_0000h
888h	PEX LUT Entry a Upper Data Register (PEXL17UDR)	32	RW	0000_0000h
88Ch	PEX LUT Entry a Lower Data Register (PEXL17LDR)	32	RW	0000_0000h
890h	PEX LUT Entry a Upper Data Register (PEXL18UDR)	32	RW	0000_0000h
894h	PEX LUT Entry a Lower Data Register (PEXL18LDR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
898h	PEX LUT Entry a Upper Data Register (PEXL19UDR)	32	RW	0000_0000h
89Ch	PEX LUT Entry a Lower Data Register (PEXL19LDR)	32	RW	0000_0000h
8A0h	PEX LUT Entry a Upper Data Register (PEXL20UDR)	32	RW	0000_0000h
8A4h	PEX LUT Entry a Lower Data Register (PEXL20LDR)	32	RW	0000_0000h
8A8h	PEX LUT Entry a Upper Data Register (PEXL21UDR)	32	RW	0000_0000h
8ACh	PEX LUT Entry a Lower Data Register (PEXL21LDR)	32	RW	0000_0000h
8B0h	PEX LUT Entry a Upper Data Register (PEXL22UDR)	32	RW	0000_0000h
8B4h	PEX LUT Entry a Lower Data Register (PEXL22LDR)	32	RW	0000_0000h
8B8h	PEX LUT Entry a Upper Data Register (PEXL23UDR)	32	RW	0000_0000h
8BCh	PEX LUT Entry a Lower Data Register (PEXL23LDR)	32	RW	0000_0000h
8C0h	PEX LUT Entry a Upper Data Register (PEXL24UDR)	32	RW	0000_0000h
8C4h	PEX LUT Entry a Lower Data Register (PEXL24LDR)	32	RW	0000_0000h
8C8h	PEX LUT Entry a Upper Data Register (PEXL25UDR)	32	RW	0000_0000h
8CCh	PEX LUT Entry a Lower Data Register (PEXL25LDR)	32	RW	0000_0000h
8D0h	PEX LUT Entry a Upper Data Register (PEXL26UDR)	32	RW	0000_0000h
8D4h	PEX LUT Entry a Lower Data Register (PEXL26LDR)	32	RW	0000_0000h
8D8h	PEX LUT Entry a Upper Data Register (PEXL27UDR)	32	RW	0000_0000h
8DCh	PEX LUT Entry a Lower Data Register (PEXL27LDR)	32	RW	0000_0000h
8E0h	PEX LUT Entry a Upper Data Register (PEXL28UDR)	32	RW	0000_0000h
8E4h	PEX LUT Entry a Lower Data Register (PEXL28LDR)	32	RW	0000_0000h
8E8h	PEX LUT Entry a Upper Data Register (PEXL29UDR)	32	RW	0000_0000h
8ECh	PEX LUT Entry a Lower Data Register (PEXL29LDR)	32	RW	0000_0000h
8F0h	PEX LUT Entry a Upper Data Register (PEXL30UDR)	32	RW	0000_0000h
8F4h	PEX LUT Entry a Lower Data Register (PEXL30LDR)	32	RW	0000_0000h
8F8h	PEX LUT Entry a Upper Data Register (PEXL31UDR)	32	RW	0000_0000h
8FCh	PEX LUT Entry a Lower Data Register (PEXL31LDR)	32	RW	0000_0000h
4_0014h	PEX PFa Config Register (PEX_PF0_CONFIG)	32	RW	0000_0000h
4_0018h	PEX PFa Interrupt status Register (PEX_PF0_INT_STAT)	32	RO	0000_0000h
4_0020h	PEX PFa PCIE pme and message detect register (PEX_PF0_PME_MES_DR)	32	W1C	0000_0000h
4_0024h	PEX PFa PCIE pme and message disable register (PEX_PF0_PME_MES_DISR)	32	RW	0000_0000h
4_0028h	PEX PFa PCIE pme and message interrupt enable register (PEX_PF0_PME_MES_IER)	32	RW	0000_0000h
4_002Ch	PEX PFa PCIE message command register (PEX_PF0_MCR)	32	RW	0000_0000h
4_0140h	PEX PFa Route By Port Address Upper register (PEX_PF0_RBP_ADDR_U)	32	RW	0000_0000h
4_0200h	PEX PFa PCIE error detect register (PEX_PF0_ERR_DR)	32	W1C	0000_0000h
4_0208h	PEX PFa PCIE error interrupt enable register (PEX_PF0_ERR_EN)	32	RW	0000_0000h
4_0210h	PEX PFa PCIE error disable register (PEX_PF0_ERR_DISR)	32	RW	0000_0000h
4_07F8h	PEX PFa VF control register (PEX_PF0_VF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4_07FCh	PEX PF0 Debug register (PEX_PF0_DBG)	32	RW	0000_0000h
5_0014h	PEX PFa Config Register (PEX_PF1_CONFIG)	32	RW	0000_0000h
5_0018h	PEX PFa Interrupt status Register (PEX_PF1_INT_STAT)	32	RO	0000_0000h
5_0020h	PEX PFa PCIE pme and message detect register (PEX_PF1_PME_MES_DR)	32	W1C	0000_0000h
5_0024h	PEX PFa PCIE pme and message disable register (PEX_PF1_PME_MES_DISR)	32	RW	0000_0000h
5_0028h	PEX PFa PCIE pme and message interrupt enable register (PEX_PF1_PME_MES_IER)	32	RW	0000_0000h
5_002Ch	PEX PFa PCIE message command register (PEX_PF1_MCR)	32	RW	0000_0000h
5_0140h	PEX PFa Route By Port Address Upper register (PEX_PF1_RBP_ADDR_U)	32	RW	0000_0000h
5_0200h	PEX PFa PCIE error detect register (PEX_PF1_ERR_DR)	32	W1C	0000_0000h
5_0208h	PEX PFa PCIE error interrupt enable register (PEX_PF1_ERR_EN)	32	RW	0000_0000h
5_0210h	PEX PFa PCIE error disable register (PEX_PF1_ERR_DISR)	32	RW	0000_0000h
5_07F8h	PEX PFa VF control register (PEX_PF1_VF)	32	RW	0000_0000h

19.5.1.2 PEX LUT Status Register (PEXLSR)

19.5.1.2.1 Offset

Register	Offset
PEXLSR	20h

19.5.1.2.2 Function

The PEX LUT Status Register (PEXLSR) logs the status of a lookup request, through the PEXLSR[LUTM] field. When no match is detected for all the lookup table entries, the lookup table miss LUTM field will be set. PEXLSR[LUTM] can be cleared by writing a 1 to it.

19.5.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LUTM	MLUTM	Reserved													
W	W1C	W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CREQID															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.2.4 Fields

Field	Function
31 LUTM	LUTM: Lookup table miss. This field is set when the AXI REQID does not match any of the lookup table entries. This field remains set from the time of an LUT miss until cleared by writing 1 to it.
30 MLUTM	MLUTM: Multiple lookup table miss. This field is set when more than one lookup failed. This field is set from the time of the second LUT miss until cleared by writing 1 to it. This bit is also set if a single miss occurs on both write and read channels simultaneously.
29-16 —	Reserved
15-0 CREQID	CREQID: Captured REQID. This field is the request ID of the first lookup table miss. If a simultaneous LUT miss is detected on both write and read channels, capture priority is given to the write channel REQID and multiple miss is reported in the MLUTM bit. This field remains unchanged from the time of an LUT miss until cleared by writing all 1s to it.

19.5.1.3 PEX LUT Control Register (PEXLCR)

19.5.1.3.1 Offset

Register	Offset
PEXLCR	24h

19.5.1.3.2 Function

The PEX LUT Control Register (PEXLCR) represents the Address Management Qualifier (AMQ) attributes, PL, BMT, 14'h0, ICID[14:0], assigned to the REQID translation when there is a lookup table miss.

19.5.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DPL	DBMT	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	DICID														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.3.4 Fields

Field	Function
31 DPL	DPL: Default privilege Level. This field is the default AMQ privilege level attribute for a lookup table miss.
30 DBMT	DBMT: Default bypass memory translation. This field is the default AMQ bypass memory translation attribute for a lookup table miss.
29-15 —	Reserved
14-0 DICID	DICID: Default isolation context ID. This field is the default AMQ isolation context ID attribute for a lookup table miss

19.5.1.4 PEX LUT Entry a Upper Data Register (PEXL0UDR)

19.5.1.4.1 Offset

Register	Offset
PEXL0UDR	800h

19.5.1.4.2 Function

The PEX LUT Entry *a* Upper Data Register (PEXL*a*UDR) represents the upper data fields of entry *a* within the PEX lookup table.

19.5.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REQID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.5.1.4.4 Fields

Field	Function
31-16 REQID	REQID: The Request ID of entry *a* of the PEX lookup table. This field is qualified by the MASK field PEXL*a*UDR[MASK] of the same PEX lookup table entry when attempting to match.
15-0 MASK	MASK: The mask field of entry *a* of the PEX lookup table. This field is qualified with the REQID field of the same entry when attempting to match. It is used to bitwise disable matches against PEXL*a*UDR[REQID] and a received packet's REQID. Note: The reset value is 0xFFFF for entry 0 and 0x0000 for all other entries of the PEX lookup table.

19.5.1.5 PEX LUT Entry a Lower Data Register (PEXL0LDR)

19.5.1.5.1 Offset

Register	Offset
PEXL0LDR	804h

19.5.1.5.2 Function

The PEX LUT Entry *a* Lower Data Register (PEXL*a*LDR) represents the lower data fields of entry *a* within the PEX lookup table.

19.5.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	PL	BMT	Reserved												
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ICID														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.5.4 Fields

Field	Function
31 EN	Enable: The enable field of entry *a* within the PEX lookup table. This field indicates whether the entry participates in matching. Note: The reset value is 0x1 for entry 0 and 0x0 for all other entries of the PEX lookup table. 0b - This entry does not participate in matching. 1b - This entry participates in matching.
30 PL	Priviledge Level: The priviledge level field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
29 BMT	Bypass memory translation: The bypass memory translation of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
28-15 —	Reserved
14-0 ICID	Isolation context ID: The isolation context ID field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.

19.5.1.6 PEX LUT Entry a Upper Data Register (PEXL1UDR - PEXL31UDR)

19.5.1.6.1 Offset

For a = 1 to 31:

Register	Offset
PEXLaUDR	800h + (a × 8h)

19.5.1.6.2 Function

The PEX LUT Entry *a* Upper Data Register (PEXL*a*UDR) represents the upper data fields of entry *a* within the PEX lookup table.

19.5.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REQID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.6.4 Fields

Field	Function
31-16 REQID	REQID: The Request ID of entry *a* of the PEX lookup table. This field is qualified by the MASK field PEXL*a*UDR[MASK] of the same PEX lookup table entry when attempting to match.
15-0 MASK	MASK: The mask field of entry *a* of the PEX lookup table. This field is qualified with the REQID field of the same entry when attempting to match. It is used to bitwise disable matches against PEXL*a*UDR[REQID] and a received packet's REQID. Note: The reset value is 0xFFFF for entry 0 and 0x0000 for all other entries of the PEX lookup table.

19.5.1.7 PEX LUT Entry a Lower Data Register (PEXL1LDR - PEXL31LDR)

19.5.1.7.1 Offset

For a = 1 to 31:

Register	Offset
PEXLaLDR	804h + (a × 8h)

19.5.1.7.2 Function

The PEX LUT Entry *a* Lower Data Register (PEXL*a*LDR) represents the lower data fields of entry *a* within the PEX lookup table.

19.5.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	PL	BMT	Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ICID														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.7.4 Fields

Field	Function
31 EN	Enable: The enable field of entry *a* within the PEX lookup table. This field indicates whether the entry participates in matching. Note: The reset value is 0x1 for entry 0 and 0x0 for all other entries of the PEX lookup table. 0b - This entry does not participate in matching. 1b - This entry participates in matching.
30 PL	Priviledge Level: The priviledge level field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
29 BMT	Bypass memory translation: The bypass memory translation of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
28-15 —	Reserved
14-0 ICID	Isolation context ID: The isolation context ID field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.

19.5.1.8 PEX PFa Config Register (PEX_PF0_CONFIG - PEX_PF1_CONFIG)

19.5.1.8.1 Offset

Register	Offset
PEX_PF0_CONFIG	4_0014h
PEX_PF1_CONFIG	5_0014h

19.5.1.8.2 Function

The PEX PF*a* Config register is used for setting the config ready control of the PCIE device.

19.5.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															CFG_READY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.8.4 Fields

Field	Function
31-1 —	Reserved
0 CFG_READY	Config Ready: This bit is cleared by default to enable EP application software configure the PCIe controller. Any config attempts from the Root Complex will be returned with config retry status (CRS). After application software finishes configuring the PCIe controller it sets the CFG_READY bit to enable accesses from the Root Complex.

19.5.1.9 PEX PFa Interrupt status Register (PEX_PF0_INT_STAT - PEX_PF1_INT_STAT)

19.5.1.9.1 Offset

Register	Offset
PEX_PF0_INT_STAT	4_0018h
PEX_PF1_INT_STAT	5_0018h

19.5.1.9.2 Function

The PEX PF*a* Interrupt status register is used to indicate that an interrupt is pending.

19.5.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INTM	INTE	Reserved													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.9.4 Fields

Field	Function
31-16 —	Reserved
15 INTM	Per PF dependent message interrupt is pending.
14 INTE	Per PF dependent error interrupt is pending.
13-0 —	Reserved

19.5.1.10 PEX PFa PCIE pme and message detect register (PEX_PF0_PME_MES_DR - PEX_PF1_PME_MES_DR)

19.5.1.10.1 Offset

Register	Offset
PEX_PF0_PME_MES_DR	4_0020h
PEX_PF1_PME_MES_DR	5_0020h

19.5.1.10.2 Function

The PEX PF*a* PCIE pme and message detect register is used to detect various power management and error events.

19.5.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PTO	Reserved	ENL23	EXL23	Reserved	HRD	LDD	Reserved	LUD	Reserved						
W	W1C		W1C	W1C		W1C	W1C		W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.10.4 Fields

Field	Function
31-16 —	Reserved
15 PTO	PTO: Indicates that PME turn off was detected. 1'b1: PME turn off was detected. 1'b0: PME turn off was not detected.

Table continues on the next page...

Field	Function
14 —	Reserved
13 ENL23	ENL23: Indicates that PCIe core entered L2/L3 ready state. 1'b1: Entered L2/3 ready state was detected. 1'b0: Entered L2/3 ready state was not detected.
12 EXL23	EXL23: Indicates that PCIe core Exited L2/L3 ready state. 1'b1: Exit L2/3 ready state was detected. 1'b0: Exit L2/3 ready state was not detected.
11 —	Reserved
10 HRD	HRD: Indicates a hot reset was detected. 1'b1: hot reset was detected. 1'b0: hot reset was not detected.
9 LDD	LDD : Indicates a link down was detected. 1'b1: link down was detected. 1'b0: link down was not detected.
8 —	Reserved
7 LUD	LUD : Indicates a link up was detected. 1'b1: link up was detected. 1'b0: link up was not detected.
6-0 —	Reserved

19.5.1.11 PEX PFa PCIE pme and message disable register (PEX_PF0_PME_MES_DISR - PEX_PF1_PME_MES_DISR)

19.5.1.11.1 Offset

Register	Offset
PEX_PF0_PME_MES_DISR	4_0024h
PEX_PF1_PME_MES_DISR	5_0024h

19.5.1.11.2 Function

The PEX PF*a* PCIE pme and message disable register is used to disable the detection of various power management and error events.

19.5.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PTOD	Reserved	ENL23D	EXL23D	Reserved	HRDD	LDDD	Reserved	LUDD	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.11.4 Fields

Field	Function
31-16 —	Reserved
15 PTOD	PTO: PME turn off detect disabled. 1'b1: PME turn off detect disabled. 1'b0: PME turn off detect enabled.
14 —	Reserved
13 ENL23D	ENL23: Entered L2/L3 ready state detect disabled. 1'b1: Entered L2/L3 ready state detect disabled. 1'b0: Entered L2/L3 ready state detect enabled.
12 EXL23D	EXL23: Exited L2/L3 ready state detect disable. 1'b1: Exit L2/L3 ready state detect disabled. 1'b0: Exit L2/L3 ready state detect enabled.
11 —	Reserved
10 HRDD	HRDD: Hot reset detect disable. 1'b1: hot reset detect disabled. 1'b0: hot reset detect enabled.
9 LDDD	LDDD : Link down detect disable. 1'b1: link down detect disabled. 1'b0: link down detect enabled.
8 —	Reserved
7 LUDD	LUDD : link up detect disable. 1'b1: link up detect disabled. 1'b0: link up detect enabled.
6-0 —	Reserved

19.5.1.12 PEX PFa PCIE pme and message interrupt enable register (PEX_PF0_PME_MES_IER - PEX_PF1_PME_MES_IER)

19.5.1.12.1 Offset

Register	Offset
PEX_PF0_PME_MES_IER	4_0028h
PEX_PF1_PME_MES_IER	5_0028h

19.5.1.12.2 Function

The PEX PF*a* PCIE pme and message disable register is used to enable interrupt generation for the detection of various power management and error events.

19.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PTOIE	Reserved	ENL23IE	EXL23IE	Reserved	HRDIE	LDDIE	Reserved	LUDIE	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.12.4 Fields

Field	Function
31-16 —	Reserved
15 PTOIE	PTO: PME turn off detect interrupt enable. 1'b1: PME turn off detect interrupt enabled. 1'b0: PME turn off detect interrupt disabled.
14 —	Reserved

Table continues on the next page...

Field	Function
13 ENL23IE	ENL23IE: Entered L2/L3 ready state detect interrupt enabled. 1'b1: Entered L2/L3 ready state detect interrupt enabled. 1'b0: Entered L2/L3 ready state detect interrupt disabled.
12 EXL23IE	EXL23IE: Exited L2/L3 ready state detect interrupt enable. 1'b1: Exit L2/L3 ready state detect interrupt enabled. 1'b0: Exit L2/L3 ready state detect interrupt disabled.
11 —	Reserved
10 HRDIE	HRDD: Hot reset detect interrupt enable. 1'b1: hot reset detect interrupt enabled. 1'b0: hot reset detect interrupt disabled.
9 LDDIE	LDDIE : Link down detect interrupt enable. 1'b1: link down detect interrupt enabled. 1'b0: link down detect interrupt disabled.
8 —	Reserved
7 LUDIE	LUDIE : link up detect interrupt enable. 1'b1: link up detect interrupt enabled. 1'b0: link up detect interrupt disabled.
6-0 —	Reserved

19.5.1.13 PEX PFa PCIE message command register (PEX_PF0_MCR - PEX_PF1_MCR)

19.5.1.13.1 Offset

Register	Offset
PEX_PF0_MCR	4_002Ch
PEX_PF1_MCR	5_002Ch

19.5.1.13.2 Function

The PEX PF*a* PCIE message command register enables the generation of interrupt and power management messages per PF.

19.5.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											INTX	Reserved	SPME S	EXL2 S	PTOMR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.13.4 Fields

Field	Function
31-5 —	Reserved
4 INTX	INTX: Assert/de-assert intx command. When sys_int goes from low to high, the core generates an Assert_INTx Message. When sys_int goes from high to low, the core generates a Deassert_INTx Message.The Interrupt Pin register for the corresponding function determines which INTx Message the core generates (INTA, INTB, INTC, or INTD).
3 —	Reserved
2 SPMES	SPME: Send PM_PME command. When set to 1 , a PM PME command is generated to wake up the the PCIE power management controller from a D1, D2 or D3 power state. The bit is self clearing. Once the bit is set , S/W needs to wait for the bit to self clear before sending a new command..
1 EXL2S	EXL2S : Exit L2 state command. When set to 1 , an L2 exit command is generated. The bit is self clearing. Once the bit is set , S/W needs to wait for the bit to self clear before sending a new command.
0 PTOMR	PTOMR : Generate PME turn off message (RC only). When set to 1, a pme turn off message is generated from the RC. The bit is self clearing. Once the bit is set , S/W needs to wait for the bit to self clear before sending a new command.

19.5.1.14 PEX PFa Route By Port Address Upper register (PEX_PF0_RBP_ADDR_U - PEX_PF1_RBP_ADDR_U)

19.5.1.14.1 Offset

Register	Offset
PEX_PF0_RBP_ADDR_U	4_0140h
PEX_PF1_RBP_ADDR_U	5_0140h

19.5.1.14.2 Function

The PEX PF*a* Route By Port Address Upper register provides the upper 16 bit out of the 64 bit address of the PCIE slave device used for the outbound transactions.

19.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR_U															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.14.4 Fields

Field	Function
31-16 ADDR_U	ADDR_U : addr[63:48] . upper 16 bit of the PCIE slave device used for the outbound transactions.
15-0 —	Reserved

19.5.1.15 PEX PFa PCIE error detect register (PEX_PF0_ERR_DR - PEX_PF1_ERR_DR)

19.5.1.15.1 Offset

Register	Offset
PEX_PF0_ERR_DR	4_0200h
PEX_PF1_ERR_DR	5_0200h

19.5.1.15.2 Function

The PEX PF*a* error detect register is used for capturing various error events .

19.5.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ME	Reserved							PCT	Reserved	PCAC	Reserved	CDNSC	Reserved		
W	W1C								W1C		W1C		W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													URE P	Reserved	
W														W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.15.4 Fields

Field	Function
31 ME	ME: Indicates Multiple errors of same type. If any of the detectable errors in PEX_ERR_DET register is detected more than one time the ME bit will be set. 1'b1: Multiple errors of same type was detected. 1'b0: Multiple errors of same type was not detected.
30-24 —	Reserved
23 PCT	PCT: Indicates completion timeout. 1'b1: Completion timeout was detected. 1'b0: Completion timeout was not detected.
22 —	Reserved
21 PCAC	PCAC: Completer abort was detected. 1'b1: Completer abort was detected. 1'b0: Completer abort was not detected.

Table continues on the next page...

Field	Function
20 —	Reserved
19 CDNSC	CDNSC : Completion with data not succsessful was detected. 1'b1: Completion with data not succsessful was detected. 1'b0: Completion with data not succsessful was not detected.
18-3 —	Reserved
2 UREP	UREP : Indicates an unsupported request completion was detected. 1'b1: Unsupported request in EP mode was detected. 1'b0: Unsupported request in EP mode not detected.
1-0 —	Reserved

19.5.1.16 PEX PFa PCIE error interrupt enable register (PEX_PF0_ERR_EN - PEX_PF1_ERR_EN)

19.5.1.16.1 Offset

Register	Offset
PEX_PF0_ERR_EN	4_0208h
PEX_PF1_ERR_EN	5_0208h

19.5.1.16.2 Function

The PEX PF*a* error interrupt enable register. Enables/disables interrupt for errors detected in PEX_ERR_DET.

19.5.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								PCTIE	Reserved	PCACIE	Reserved	CDNSCIE	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													UREPIE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.16.4 Fields

Field	Function
31-24 —	Reserved
23 PCTIE	PCTIE: completion timeout interrupt enable. 1'b1: Completion timeout interrupt enabled. 1'b0: Completion timeout interrupt disabled.
22 —	Reserved
21 PCACIE	PCACIE: Completer abort interrupt enable. 1'b1: Completer abort interrupt enabled. 1'b0: Completer abort interrupt disabled.
20 —	Reserved
19 CDNSCIE	CDNSCIE : Completion with data not succsessful interrupt enable. 1'b1: Completion with data not succsessful interrupt enabled. 1'b0: Completion with data not succsessful was not detected.
18-3 —	Reserved
2 UREPIE	UREPIE : Unsupported request in EP mode interrupt enable. 1'b1: Unsupported request in EP mode interrupt enabled. 1'b0: Unsupported request in EP mode interrupt disabled.
1-0 —	Reserved

19.5.1.17 PEX PFa PCIE error disable register (PEX_PF0_ERR_DISR - PEX_PF1_ERR_DISR)

19.5.1.17.1 Offset

Register	Offset
PEX_PF0_ERR_DISR	4_0210h
PEX_PF1_ERR_DISR	5_0210h

19.5.1.17.2 Function

The PEX PF*a* error disable register. disables detection of errors in PEX_ERR_DET.

19.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MED	Reserved							PCTD	Reserved	PCACD	Reserved	CDNSCD	Reserved		
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													UREP D	Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.17.4 Fields

Field	Function
31 MED	ME: Multiple errors of same type detection disable. 1'b1: Multiple errors of same type detection disabled. 1'b0: Multiple errors of same type detection enabled.
30-24 —	Reserved
23 PCTD	PCTIE: completion detection disable. 1'b1: Completion timeout detection disabled. 1'b0: Completion timeout detection enabled.
22 —	Reserved

Table continues on the next page...

Field	Function
21 PCACD	PCACD: Completer abort detection disable. 1'b1: Completer abort detection disabled. 1'b0: Completer abort detection enabled.
20 —	Reserved
19 CDNSCD	CDNSCD : Completion with data not succsessful detection disable. 1'b1: Completion with data not succsessful detection disabled. 1'b0: Completion with data not succsessful detection enabled.
18-3 —	Reserved
2 UREPD	UREPD : Unsupported request in EP mode detection disable. 1'b1: Unsupported request in EP mode detection disabled. 1'b0: Unsupported request in EP mode detection enabled.
1-0 —	Reserved

19.5.1.18 PEX PFa VF control register (PEX_PF0_VF - PEX_PF1_VF)

19.5.1.18.1 Offset

Register	Offset
PEX_PF0_VF	4_07F8h
PEX_PF1_VF	5_07F8h

19.5.1.18.2 Function

The PEX PF*a* VF control register enables configuring the SR-IOV VF settings for each PF.

19.5.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				VF						VF_ACT	Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.18.4 Fields

Field	Function
31-28 —	Reserved
27-22 VF	Virtual Function number VF+1 selects the virtual function number of the transaction.
21 VF_ACT	VF Active This field controls access to the VF register space. 0b - De-activates VF register access 1b - Activates VF register access
20-0 —	Reserved

19.5.1.19 PEX PF0 Debug register (PEX_PF0_DBG)

19.5.1.19.1 Offset

Register	Offset
PEX_PF0_DBG	4_07FCh

19.5.1.19.2 Function

By setting the WE field, the user has access to the SR field. Once set, SR will enable soft reset to the PEX module. LTSSM is a read only field that reports the LTSSM status from the PEX module.

19.5.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WE	SR	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										LTSSM					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.1.19.4 Fields

Field	Function
31 WE	Write Enable This bit when set allows the user to have write access to the PEXDBG[SR] for both setting and clearing the soft reset on the PEX module.
30 SR	Soft Reset When SR is set to 1, the PEX module enters soft reset. The PEX mode remains in soft reset while SR is set to 1. When SR is cleared, the PEX module exits soft reset. If SR is to be set and cleared multiple times back to back, software must wait a few seconds before changing the bit value to allow each new state to take effect.
29-6 —	Reserved
5-0 LTSSM	Link Training Status State Machine (LTSSM) status This field is read only and captures the PEX block LTSSM state at each clock cycle. These bits indicate the link training status. This field is useful for debugging link training failures. **NOTE:** The status code changes while reacting to the link status. Therefore, software may need to read LTSSM multiple times to ensure the value has stabilized. 000000b - DETECT_QUIET 000001b - DETECT_ACTIVE 000010b - POLL_ACTIVE 000011b - POLL_COMPLIANCE 000100b - POLL_CONFIG 000101b - PRE_DETECT_QUIET 000110b - DETECT_WAIT 000111b - CFG_LINKWD_START 001000b - CFG_LINKWD_ACEPT

Field	Function
	001001b - CFG_LANENUM_WAIT 001010b - CFG_LANENUM_ACEPT 001011b - CFG_COMPLETE 001100b - CFG_IDLE 001101b - RCVRY_LOCK 001110b - RCVRY_SPEED 001111b - RCVRY_RCVRCFG 010000b - RCVRY_IDLE 010001b - L0 010010b - L0S 010011b - L123_SEND_EIDLE 010100b - L1_IDLE 010101b - L2_IDLE 010110b - L2_WAKE 010111b - DISABLED_ENTRY 011000b - DISABLED_IDLE 011001b - DISABLED 011010b - LPBK_ENTRY 011011b - LPBK_ACTIVE 011100b - LPBK_EXIT 011101b - LPBK_EXIT_TIMEOUT 011110b - HOT_RESET_ENTRY 011111b - HOT_RESET 100000b - RCVRY_EQ0 100001b - RCVRY_EQ1 100010b - RCVRY_EQ2 100011b - RCVRY_EQ3

19.5.2 Non-SR-IOV PEX Look Up Table (PEX2_LUT and PEX3_LUT) register descriptions

This section describes the registers for the non-SR-IOV instances (PEX2_LUT and PEX3_LUT); see SR-IOV PEX Look Up Table (PEX1_LUT) register descriptions for the descriptions of the registers for the SR-IOV instance of the PEX Look Up Table (PEX1_LUT).

19.5.2.1 PEX_LUT memory map

PEX2_LUT base address: 358_0000h

PEX3_LUT base address: 368_0000h

Offset	Register	Width (In bits)	Access	Reset value
20h	PEX LUT Status Register (PEXLSR)	32	W1C	0000_0000h
24h	PEX LUT Control Register (PEXLCR)	32	RW	0000_0000h
800h	PEX LUT Entry a Upper Data Register (PEXL0UDR)	32	RW	0000_FFFFh
804h	PEX LUT Entry a Lower Data Register (PEXL0LDR)	32	RW	8000_0000h
808h	PEX LUT Entry a Upper Data Register (PEXL1UDR)	32	RW	0000_0000h
80Ch	PEX LUT Entry a Lower Data Register (PEXL1LDR)	32	RW	0000_0000h
810h	PEX LUT Entry a Upper Data Register (PEXL2UDR)	32	RW	0000_0000h
814h	PEX LUT Entry a Lower Data Register (PEXL2LDR)	32	RW	0000_0000h
818h	PEX LUT Entry a Upper Data Register (PEXL3UDR)	32	RW	0000_0000h
81Ch	PEX LUT Entry a Lower Data Register (PEXL3LDR)	32	RW	0000_0000h
820h	PEX LUT Entry a Upper Data Register (PEXL4UDR)	32	RW	0000_0000h
824h	PEX LUT Entry a Lower Data Register (PEXL4LDR)	32	RW	0000_0000h
828h	PEX LUT Entry a Upper Data Register (PEXL5UDR)	32	RW	0000_0000h
82Ch	PEX LUT Entry a Lower Data Register (PEXL5LDR)	32	RW	0000_0000h
830h	PEX LUT Entry a Upper Data Register (PEXL6UDR)	32	RW	0000_0000h
834h	PEX LUT Entry a Lower Data Register (PEXL6LDR)	32	RW	0000_0000h
838h	PEX LUT Entry a Upper Data Register (PEXL7UDR)	32	RW	0000_0000h
83Ch	PEX LUT Entry a Lower Data Register (PEXL7LDR)	32	RW	0000_0000h
840h	PEX LUT Entry a Upper Data Register (PEXL8UDR)	32	RW	0000_0000h
844h	PEX LUT Entry a Lower Data Register (PEXL8LDR)	32	RW	0000_0000h
848h	PEX LUT Entry a Upper Data Register (PEXL9UDR)	32	RW	0000_0000h
84Ch	PEX LUT Entry a Lower Data Register (PEXL9LDR)	32	RW	0000_0000h
850h	PEX LUT Entry a Upper Data Register (PEXL10UDR)	32	RW	0000_0000h
854h	PEX LUT Entry a Lower Data Register (PEXL10LDR)	32	RW	0000_0000h
858h	PEX LUT Entry a Upper Data Register (PEXL11UDR)	32	RW	0000_0000h
85Ch	PEX LUT Entry a Lower Data Register (PEXL11LDR)	32	RW	0000_0000h
860h	PEX LUT Entry a Upper Data Register (PEXL12UDR)	32	RW	0000_0000h
864h	PEX LUT Entry a Lower Data Register (PEXL12LDR)	32	RW	0000_0000h
868h	PEX LUT Entry a Upper Data Register (PEXL13UDR)	32	RW	0000_0000h
86Ch	PEX LUT Entry a Lower Data Register (PEXL13LDR)	32	RW	0000_0000h
870h	PEX LUT Entry a Upper Data Register (PEXL14UDR)	32	RW	0000_0000h
874h	PEX LUT Entry a Lower Data Register (PEXL14LDR)	32	RW	0000_0000h
878h	PEX LUT Entry a Upper Data Register (PEXL15UDR)	32	RW	0000_0000h
87Ch	PEX LUT Entry a Lower Data Register (PEXL15LDR)	32	RW	0000_0000h
880h	PEX LUT Entry a Upper Data Register (PEXL16UDR)	32	RW	0000_0000h
884h	PEX LUT Entry a Lower Data Register (PEXL16LDR)	32	RW	0000_0000h
888h	PEX LUT Entry a Upper Data Register (PEXL17UDR)	32	RW	0000_0000h
88Ch	PEX LUT Entry a Lower Data Register (PEXL17LDR)	32	RW	0000_0000h
890h	PEX LUT Entry a Upper Data Register (PEXL18UDR)	32	RW	0000_0000h
894h	PEX LUT Entry a Lower Data Register (PEXL18LDR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
898h	PEX LUT Entry a Upper Data Register (PEXL19UDR)	32	RW	0000_0000h
89Ch	PEX LUT Entry a Lower Data Register (PEXL19LDR)	32	RW	0000_0000h
8A0h	PEX LUT Entry a Upper Data Register (PEXL20UDR)	32	RW	0000_0000h
8A4h	PEX LUT Entry a Lower Data Register (PEXL20LDR)	32	RW	0000_0000h
8A8h	PEX LUT Entry a Upper Data Register (PEXL21UDR)	32	RW	0000_0000h
8ACh	PEX LUT Entry a Lower Data Register (PEXL21LDR)	32	RW	0000_0000h
8B0h	PEX LUT Entry a Upper Data Register (PEXL22UDR)	32	RW	0000_0000h
8B4h	PEX LUT Entry a Lower Data Register (PEXL22LDR)	32	RW	0000_0000h
8B8h	PEX LUT Entry a Upper Data Register (PEXL23UDR)	32	RW	0000_0000h
8BCh	PEX LUT Entry a Lower Data Register (PEXL23LDR)	32	RW	0000_0000h
8C0h	PEX LUT Entry a Upper Data Register (PEXL24UDR)	32	RW	0000_0000h
8C4h	PEX LUT Entry a Lower Data Register (PEXL24LDR)	32	RW	0000_0000h
8C8h	PEX LUT Entry a Upper Data Register (PEXL25UDR)	32	RW	0000_0000h
8CCh	PEX LUT Entry a Lower Data Register (PEXL25LDR)	32	RW	0000_0000h
8D0h	PEX LUT Entry a Upper Data Register (PEXL26UDR)	32	RW	0000_0000h
8D4h	PEX LUT Entry a Lower Data Register (PEXL26LDR)	32	RW	0000_0000h
8D8h	PEX LUT Entry a Upper Data Register (PEXL27UDR)	32	RW	0000_0000h
8DCh	PEX LUT Entry a Lower Data Register (PEXL27LDR)	32	RW	0000_0000h
8E0h	PEX LUT Entry a Upper Data Register (PEXL28UDR)	32	RW	0000_0000h
8E4h	PEX LUT Entry a Lower Data Register (PEXL28LDR)	32	RW	0000_0000h
8E8h	PEX LUT Entry a Upper Data Register (PEXL29UDR)	32	RW	0000_0000h
8ECh	PEX LUT Entry a Lower Data Register (PEXL29LDR)	32	RW	0000_0000h
8F0h	PEX LUT Entry a Upper Data Register (PEXL30UDR)	32	RW	0000_0000h
8F4h	PEX LUT Entry a Lower Data Register (PEXL30LDR)	32	RW	0000_0000h
8F8h	PEX LUT Entry a Upper Data Register (PEXL31UDR)	32	RW	0000_0000h
8FCh	PEX LUT Entry a Lower Data Register (PEXL31LDR)	32	RW	0000_0000h
4_0014h	PEX PFa Config Register (PEX_PF0_CONFIG)	32	RW	0000_0000h
4_0018h	PEX PFa Interrupt status Register (PEX_PF0_INT_STAT)	32	RO	0000_0000h
4_0020h	PEX PFa PCIE pme and message detect register (PEX_PF0_PME_MES_DR)	32	W1C	0000_0000h
4_0024h	PEX PFa PCIE pme and message disable register (PEX_PF0_PME_MES_DISR)	32	RW	0000_0000h
4_0028h	PEX PFa PCIE pme and message interrupt enable register (PEX_PF0_PME_MES_IER)	32	RW	0000_0000h
4_002Ch	PEX PFa PCIE message command register (PEX_PF0_MCR)	32	RW	0000_0000h
4_0140h	PEX PFa Route By Port Address Upper register (PEX_PF0_RBP_ADDR_U)	32	RW	0000_0000h
4_0200h	PEX PFa PCIE error detect register (PEX_PF0_ERR_DR)	32	W1C	0000_0000h
4_0208h	PEX PFa PCIE error interrupt enable register (PEX_PF0_ERR_EN)	32	RW	0000_0000h
4_0210h	PEX PFa PCIE error disable register (PEX_PF0_ERR_DISR)	32	RW	0000_0000h
4_07FCh	PEX PF0 Debug register (PEX_PF0_DBG)	32	RW	0000_0000h

19.5.2.2 PEX LUT Status Register (PEXLSR)

19.5.2.2.1 Offset

Register	Offset
PEXLSR	20h

19.5.2.2.2 Function

The PEX LUT Status Register (PEXLSR) logs the status of a lookup request, through the PEXLSR[LUTM] field. When no match is detected for all the lookup table entries, the lookup table miss LUTM field will be set. PEXLSR[LUTM] can be cleared by writing a 1 to it.

19.5.2.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LUTM	MLUTM	Reserved													
W	W1C	W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CREQID															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.2.4 Fields

Field	Function
31 LUTM	LUTM: Lookup table miss. This field is set when the AXI REQID does not match any of the lookup table entries. This field remains set from the time of an LUT miss until cleared by writing 1 to it.
30 MLUTM	MLUTM: Multiple lookup table miss. This field is set when more than one lookup failed. This field is set from the time of the second LUT miss until cleared by writing 1 to it. This bit is also set if a single miss occurs on both write and read channels simultaneously.
29-16	Reserved

Table continues on the next page...

Field	Function
—	
15-0 CREQID	CREQID: Captured REQID. This field is the request ID of the first lookup table miss. If a simultaneous LUT miss is detected on both write and read channels, capture priority is given to the write channel REQID and multiple miss is reported in the MLUTM bit. This field remains unchanged from the time of an LUT miss until cleared by writing all 1s to it.

19.5.2.3 PEX LUT Control Register (PEXLCR)

19.5.2.3.1 Offset

Register	Offset
PEXLCR	24h

19.5.2.3.2 Function

The PEX LUT Control Register (PEXLCR) represents the Address Management Qualifier (AMQ) attributes, PL, BMT, 14'h0, ICID[14:0], assigned to the REQID translation when there is a lookup table miss.

19.5.2.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DPL	DBMT	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	DICID														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.3.4 Fields

Field	Function
31 DPL	DPL: Default privilege Level. This field is the default AMQ privilege level attribute for a lookup table miss.
30 DBMT	DBMT: Default bypass memory translation. This field is the default AMQ bypass memory translation attribute for a lookup table miss.
29-15 —	Reserved
14-0 DICID	DICID: Default isolation context ID. This field is the default AMQ isolation context ID attribute for a lookup table miss

19.5.2.4 PEX LUT Entry a Upper Data Register (PEXL0UDR)

19.5.2.4.1 Offset

Register	Offset
PEXL0UDR	800h

19.5.2.4.2 Function

The PEX LUT Entry *a* Upper Data Register (PEXL*a*UDR) represents the upper data fields of entry *a* within the PEX lookup table.

19.5.2.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REQID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19.5.2.4.4 Fields

Field	Function
31-16 REQID	REQID: The Request ID of entry *a* of the PEX lookup table. This field is qualified by the MASK field PEXL*a*UDR[MASK] of the same PEX lookup table entry when attempting to match.
15-0 MASK	MASK: The mask field of entry *a* of the PEX lookup table. This field is qualified with the REQID field of the same entry when attempting to match. It is used to bitwise disable matches against PEXL*a*UDR[REQID] and a received packet's REQID. Note: The reset value is 0xFFFF for entry 0 and 0x0000 for all other entries of the PEX lookup table.

19.5.2.5 PEX LUT Entry a Lower Data Register (PEXL0LDR)

19.5.2.5.1 Offset

Register	Offset
PEXL0LDR	804h

19.5.2.5.2 Function

The PEX LUT Entry *a* Lower Data Register (PEXL*a*LDR) represents the lower data fields of entry *a* within the PEX lookup table.

19.5.2.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	PL	BMT	Reserved												
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ICID														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.5.4 Fields

Field	Function
31 EN	Enable: The enable field of entry *a* within the PEX lookup table. This field indicates whether the entry participates in matching. Note: The reset value is 0x1 for entry 0 and 0x0 for all other entries of the PEX lookup table. 0b - This entry does not participate in matching. 1b - This entry participates in matching.
30 PL	Priviledge Level: The priviledge level field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
29 BMT	Bypass memory translation: The bypass memory translation of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
28-15 —	Reserved
14-0 ICID	Isolation context ID: The isolation context ID field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.

19.5.2.6 PEX LUT Entry a Upper Data Register (PEXL1UDR - PEXL31UDR)

19.5.2.6.1 Offset

For a = 1 to 31:

Register	Offset
PEXLaUDR	800h + (a × 8h)

19.5.2.6.2 Function

The PEX LUT Entry *a* Upper Data Register (PEXL*a*UDR) represents the upper data fields of entry *a* within the PEX lookup table.

19.5.2.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REQID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.6.4 Fields

Field	Function
31-16 REQID	REQID: The Request ID of entry *a* of the PEX lookup table. This field is qualified by the MASK field PEXL*a*UDR[MASK] of the same PEX lookup table entry when attempting to match.
15-0 MASK	MASK: The mask field of entry *a* of the PEX lookup table. This field is qualified with the REQID field of the same entry when attempting to match. It is used to bitwise disable matches against PEXL*a*UDR[REQID] and a received packet's REQID. Note: The reset value is 0xFFFF for entry 0 and 0x0000 for all other entries of the PEX lookup table.

19.5.2.7 PEX LUT Entry a Lower Data Register (PEXL1LDR - PEXL31LDR)

19.5.2.7.1 Offset

For a = 1 to 31:

Register	Offset
PEXLaLDR	804h + (a × 8h)

19.5.2.7.2 Function

The PEX LUT Entry *a* Lower Data Register (PEXL*a*LDR) represents the lower data fields of entry *a* within the PEX lookup table.

19.5.2.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	PL	BMT	Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ICID														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.7.4 Fields

Field	Function
31 EN	Enable: The enable field of entry *a* within the PEX lookup table. This field indicates whether the entry participates in matching. Note: The reset value is 0x1 for entry 0 and 0x0 for all other entries of the PEX lookup table. 0b - This entry does not participate in matching. 1b - This entry participates in matching.
30 PL	Priviledge Level: The priviledge level field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
29 BMT	Bypass memory translation: The bypass memory translation of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.
28-15 —	Reserved
14-0 ICID	Isolation context ID: The isolation context ID field of entry *a* within the PEX lookup table. If this entry matches, this field will be added to the outgoing AXI request sideband.

19.5.2.8 PEX PFa Config Register (PEX_PF0_CONFIG)

19.5.2.8.1 Offset

Register	Offset
PEX_PF0_CONFIG	4_0014h

19.5.2.8.2 Function

The PEX PF*a* Config register is used for setting the config ready control of the PCIE device.

19.5.2.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															CFG_READY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.8.4 Fields

Field	Function
31-1 —	Reserved
0 CFG_READY	Config Ready: This bit is cleared by default to enable EP application software configure the PCIe controller. Any config attempts from the Root Complex will be returned with config retry status (CRS). After application software finishes configuring the PCIe controller it sets the CFG_READY bit to enable accesses from the Root Complex.

19.5.2.9 PEX PFa Interrupt status Register (PEX_PF0_INT_STAT)

19.5.2.9.1 Offset

Register	Offset
PEX_PF0_INT_STAT	4_0018h

19.5.2.9.2 Function

The PEX PF*a* Interrupt status register is used to indicate that an interrupt is pending.

19.5.2.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INTM	INTE	Reserved													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.9.4 Fields

Field	Function
31-16 —	Reserved
15 INTM	Per PF dependent message interrupt is pending.
14 INTE	Per PF dependent error interrupt is pending.
13-0 —	Reserved

19.5.2.10 PEX PFa PCIE pme and message detect register (PEX_PF0_PME_MES_DR)

19.5.2.10.1 Offset

Register	Offset
PEX_PF0_PME_MES_DR	4_0020h

19.5.2.10.2 Function

The PEX PF*a* PCIE pme and message detect register is used to detect various power management and error events.

19.5.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PTO	Reserved	ENL23	EXL23	Reserved	HRD	LDD	Reserved	LUD	Reserved						
W	W1C		W1C	W1C		W1C	W1C		W1C							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.10.4 Fields

Field	Function
31-16 —	Reserved
15 PTO	PTO: Indicates that PME turn off was detected. 1'b1: PME turn off was detected. 1'b0: PME turn off was not detected.
14 —	Reserved
13 ENL23	ENL23: Indicates that PCIe core entered L2/L3 ready state. 1'b1: Entered L2/3 ready state was detected. 1'b0: Entered L2/3 ready state was not detected.
12 EXL23	EXL23: Indicates that PCIe core Exited L2/L3 ready state. 1'b1: Exit L2/3 ready state was detected. 1'b0: Exit L2/3 ready state was not detected.
11 —	Reserved
10 HRD	HRD: Indicates a hot reset was detected. 1'b1: hot reset was detected. 1'b0: hot reset was not detected.
9 LDD	LDD : Indicates a link down was detected. 1'b1: link down was detected. 1'b0: link down was not detected.
8 —	Reserved

Table continues on the next page...

Field	Function
7 LUD	LUD : Indicates a link up was detected. 1'b1: link up was detected. 1'b0: link up was not detected.
6-0 —	Reserved

19.5.2.11 PEX PFa PCIE pme and message disable register (PEX_PF0_PME_MES_DISR)

19.5.2.11.1 Offset

Register	Offset
PEX_PF0_PME_MES_DISR	4_0024h

19.5.2.11.2 Function

The PEX PF*a* PCIE pme and message disable register is used to disable the detection of various power management and error events.

19.5.2.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PTOD	Reserved	ENL23D	EXL23D	Reserved	HRDD	LDDD	Reserved	LUDD	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.11.4 Fields

Field	Function
31-16	Reserved

Table continues on the next page...

Field	Function
—	
15 PTOD	PTO: PME turn off detect disabled. 1'b1: PME turn off detect disabled. 1'b0: PME turn off detect enabled.
14 —	Reserved
13 ENL23D	ENL23: Entered L2/L3 ready state detect disabled. 1'b1: Entered L2/L3 ready state detect disabled. 1'b0: Entered L2/L3 ready state detect enabled.
12 EXL23D	EXL23: Exited L2/L3 ready state detect disable. 1'b1: Exit L2/L3 ready state detect disabled. 1'b0: Exit L2/L3 ready state detect enabled.
11 —	Reserved
10 HRDD	HRDD: Hot reset detect disable. 1'b1: hot reset detect disabled. 1'b0: hot reset detect enabled.
9 LDDD	LDDD : Link down detect disable. 1'b1: link down detect disabled. 1'b0: link down detect enabled.
8 —	Reserved
7 LUDD	LUDD : link up detect disable. 1'b1: link up detect disabled. 1'b0: link up detect enabled.
6-0 —	Reserved

19.5.2.12 PEX PFa PCIE pme and message interrupt enable register (PEX_PF0_PME_MES_IER)

19.5.2.12.1 Offset

Register	Offset
PEX_PF0_PME_MES_IER	4_0028h

19.5.2.12.2 Function

The PEX PF*a* PCIE pme and message disable register is used to enable interrupt generation for the detection of various power management and error events.

19.5.2.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PTOIE	Reserved	ENL23IE	EXL23IE	Reserved	HRDIE	LDDIE	Reserved	LUDIE	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.12.4 Fields

Field	Function
31-16 —	Reserved
15 PTOIE	PTO: PME turn off detect interrupt enable. 1'b1: PME turn off detect interrupt enabled. 1'b0: PME turn off detect interrupt disabled.
14 —	Reserved
13 ENL23IE	ENL23IE: Entered L2/L3 ready state detect interrupt enabled. 1'b1: Entered L2/L3 ready state detect interrupt enabled. 1'b0: Entered L2/L3 ready state detect interrupt disabled.
12 EXL23IE	EXL23IE: Exited L2/L3 ready state detect interrupt enable. 1'b1: Exit L2/L3 ready state detect interrupt enabled. 1'b0: Exit L2/L3 ready state detect interrupt disabled.
11 —	Reserved
10 HRDIE	HRDD: Hot reset detect interrupt enable. 1'b1: hot reset detect interrupt enabled. 1'b0: hot reset detect interrupt disabled.
9 LDDIE	LDDIE : Link down detect interrupt enable. 1'b1: link down detect interrupt enabled. 1'b0: link down detect interrupt disabled.
8 —	Reserved
7 LUDIE	LUDIE : link up detect interrupt enable. 1'b1: link up detect interrupt enabled. 1'b0: link up detect interrupt disabled.
6-0 —	Reserved

19.5.2.13 PEX PFa PCIE message command register (PEX_PF0_MCR)

19.5.2.13.1 Offset

Register	Offset
PEX_PF0_MCR	4_002Ch

19.5.2.13.2 Function

The PEX PF*a* PCIE message command register enables the generation of interrupt and power management messages per PF.

19.5.2.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											INTX	Reserved	SPMES	EXL2S	PTOMR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.13.4 Fields

Field	Function
31-5 —	Reserved
4 INTX	INTX: Assert/de-assert intx command. When sys_int goes from low to high, the core generates an Assert_INTx Message. When sys_int goes from high to low, the core generates a Deassert_INTx Message.The Interrupt Pin register for the corresponding function determines which INTx Message the core generates (INTA).
3 —	Reserved
2 SPMES	SPME: Send PM_PME command. When set to 1 , a PM PME command is generated to wake up the the PCIE power management controller from a D1, D2 or D3 power state. The bit is self clearing. Once the bit is set , S/W needs to wait for the bit to self clear before sending a new command..

Table continues on the next page...

Field	Function
1 EXL2S	EXL2S : Exit L2 state command. When set to 1 , an L2 exit command is generated. The bit is self clearing. Once the bit is set , S/W needs to wait for the bit to self clear before sending a new command.
0 PTOMR	PTOMR : Generate PME turn off message (RC only). When set to 1, a pme turn off message is generated from the RC. The bit is self clearing. Once the bit is set , S/W needs to wait for the bit to self clear before sending a new command.

19.5.2.14 PEX PFa Route By Port Address Upper register (PEX_PF0_RBP_ADDR_U)

19.5.2.14.1 Offset

Register	Offset
PEX_PF0_RBP_ADDR_U	4_0140h

19.5.2.14.2 Function

The PEX PF*a* Route By Port Address Upper register provides the upper 16 bit out of the 64 bit address of the PCIE slave device used for the outbound transactions.

19.5.2.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR_U															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.14.4 Fields

Field	Function
31-16	ADDR_U : addr[63:48] . upper 16 bit of the PCIE slave device used for the outbound transactions.

Table continues on the next page...

Field	Function
ADDR_U	
15-0 —	Reserved

19.5.2.15 PEX PFa PCIE error detect register (PEX_PF0_ERR_DR)

19.5.2.15.1 Offset

Register	Offset
PEX_PF0_ERR_DR	4_0200h

19.5.2.15.2 Function

The PEX PF*a* error detect register is used for capturing various error events .

19.5.2.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ME	Reserved							PCT	Reserved	PCAC	Reserved	CDNSC	Reserved		
W	W1C								W1C		W1C		W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													URE P	Reserved	
W														W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.15.4 Fields

Field	Function
31 ME	ME: Indicates Multiple errors of same type. If any of the detectable errors in PEX_ERR_DET register is detected more than one time the ME bit will be set. 1'b1: Multiple errors of same type was detected. 1'b0: Multiple errors of same type was not detected.
30-24 —	Reserved
23 PCT	PCT: Indicates completion timeout. 1'b1: Completion timeout was detected. 1'b0: Completion timeout was not detected.
22 —	Reserved
21 PCAC	PCAC: Completer abort was detected. 1'b1: Completer abort was detected. 1'b0: Completer abort was not detected.
20 —	Reserved
19 CDNSC	CDNSC : Completion with data not succsessful was detected. 1'b1: Completion with data not succsessful was detected. 1'b0: Completion with data not succsessful was not detected.
18-3 —	Reserved
2 UREP	UREP : Indicates an unsupported request completion was detected. 1'b1: Unsupported request in EP mode was detected. 1'b0: Unsupported request in EP mode not detected.
1-0 —	Reserved

19.5.2.16 PEX PFa PCIE error interrupt enable register (PEX_PF0_ERR_EN)

19.5.2.16.1 Offset

Register	Offset
PEX_PF0_ERR_EN	4_0208h

19.5.2.16.2 Function

The PEX PF*a* error interrupt enable register. Enables/disables interrupt for errors detected in PEX_ERR_DET.

19.5.2.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								PCTIE	Reserved	PCACIE	Reserved	CDNSCIE	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													UREPIE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.16.4 Fields

Field	Function
31-24 —	Reserved
23 PCTIE	PCTIE: completion timeout interrupt enable. 1'b1: Completion timeout interrupt enabled. 1'b0: Completion timeout interrupt disabled.
22 —	Reserved
21 PCACIE	PCACIE: Completer abort interrupt enable. 1'b1: Completer abort interrupt enabled. 1'b0: Completer abort interrupt disabled.
20 —	Reserved
19 CDNSCIE	CDNSCIE : Completion with data not succsessful interrupt enable. 1'b1: Completion with data not succsessful interrupt enabled. 1'b0: Completion with data not succsessful was not detected.
18-3 —	Reserved
2 UREPIE	UREPIE : Unsupported request in EP mode interrupt enable. 1'b1: Unsupported request in EP mode interrupt enabled. 1'b0: Unsupported request in EP mode interrupt disabled.
1-0 —	Reserved

19.5.2.17 PEX PFa PCIE error disable register (PEX_PF0_ERR_DISR)

19.5.2.17.1 Offset

Register	Offset
PEX_PF0_ERR_DISR	4_0210h

19.5.2.17.2 Function

The PEX PF*a* error disable register. disables detection of errors in PEX_ERR_DET.

19.5.2.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MED	Reserved							PCTD	Reserved	PCACD	Reserved	CDNSCD	Reserved		
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													UREP D	Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.17.4 Fields

Field	Function
31 MED	ME: Multiple errors of same type detection disable. 1'b1: Multiple errors of same type detection disabled. 1'b0: Multiple errors of same type detection enabled.
30-24 —	Reserved
23 PCTD	PCTIE: completion detection disable. 1'b1: Completion timeout detection disabled. 1'b0: Completion timeout detection enabled.
22 —	Reserved
21 PCACD	PCACD: Completer abort detection disable. 1'b1: Completer abort detection disabled. 1'b0: Completer abort detection enabled.
20 —	Reserved

Table continues on the next page...

Field	Function
19 CDNSCD	CDNSCD : Completion with data not succsessful detection disable. 1'b1: Completion with data not succsessful detection disabled. 1'b0: Completion with data not succsessful detection enabled.
18-3 —	Reserved
2 UREPD	UREPD : Unsupported request in EP mode detection disable. 1'b1: Unsupported request in EP mode detection disabled. 1'b0: Unsupported request in EP mode detection enabled.
1-0 —	Reserved

19.5.2.18 PEX PF0 Debug register (PEX_PF0_DBG)

19.5.2.18.1 Offset

Register	Offset
PEX_PF0_DBG	4_07FCh

19.5.2.18.2 Function

By setting the WE field, the user has access to the SR field. Once set, SR will enable soft reset to the PEX module. LTSSM is a read only field that reports the LTSSM status from the PEX module.

19.5.2.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WE	SR	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										LTSSM					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

19.5.2.18.4 Fields

Field	Function
31 WE	Write Enable This bit when set allows the user to have write access to the PEXDBG[SR] for both setting and clearing the soft reset on the PEX module.
30 SR	Soft Reset When SR is set to 1, the PEX module enters soft reset. The PEX mode remains in soft reset while SR is set to 1. When SR is cleared, the PEX module exits soft reset. If SR is to be set and cleared multiple times back to back, software must wait a few seconds before changing the bit value to allow each new state to take effect.
29-6 —	Reserved
5-0 LTSSM	Link Training Status State Machine (LTSSM) status This field is read only and captures the PEX block LTSSM state at each clock cycle. These bits indicate the link training status. This field is useful for debugging link training failures. **NOTE:** The status code changes while reacting to the link status. Therefore, software may need to read LTSSM multiple times to ensure the value has stabilized. 000000b - DETECT_QUIET 000001b - DETECT_ACTIVE 000010b - POLL_ACTIVE 000011b - POLL_COMPLIANCE 000100b - POLL_CONFIG 000101b - PRE_DETECT_QUIET 000110b - DETECT_WAIT 000111b - CFG_LINKWD_START 001000b - CFG_LINKWD_ACEPT 001001b - CFG_LANENUM_WAIT 001010b - CFG_LANENUM_ACEPT 001011b - CFG_COMPLETE 001100b - CFG_IDLE 001101b - RCVRY_LOCK 001110b - RCVRY_SPEED 001111b - RCVRY_RCVRCFG 010000b - RCVRY_IDLE 010001b - L0 010010b - L0S 010011b - L123_SEND_EIDLE 010100b - L1_IDLE 010101b - L2_IDLE 010110b - L2_WAKE 010111b - DISABLED_ENTRY 011000b - DISABLED_IDLE 011001b - DISABLED 011010b - LPBK_ENTRY 011011b - LPBK_ACTIVE 011100b - LPBK_EXIT 011101b - LPBK_EXIT_TIMEOUT 011110b - HOT_RESET_ENTRY 011111b - HOT_RESET 100000b - RCVRY_EQ0 100001b - RCVRY_EQ1 100010b - RCVRY_EQ2 100011b - RCVRY_EQ3

19.6 Functional Description

The PCI Express protocol relies on a requestor/completer relationship where one device requests that some desired action be performed by some target device and the target device completes the task and responds. Usually the requests and responses occur through a network of links, but to the requester and to the completer, the intermediate components are transparent.

Figure 19-2. Requester/Completer Relationship

Each PCI Express device is divided into two halves—transmit (TX) and receive (RX), and each of these halves is further divided into three layers—transaction, data link, and physical, as shown in the following figure.

Figure 19-3. PCI Express High-Level Layering

Packets are formed in the transaction layer (TLPs) and data link layer (DLLPs), and each subsequent layer adds the necessary encodings and framing, as shown in the following figure. As packets are received, they are decoded and processed by the same layers but in reverse order, so they may be processed by the layer or by the device application software.

Figure 19-4. PCI Express Packet Flow

19.6.1 Architecture

This section describes implementation details of the PCI Express controller.

19.6.1.1 PCI Express Transactions

The following table contains the list of transactions that the PCI Express controller supports as an initiator and a target.

Table 19-5. PCI Express Transactions

PCI Express Transaction	Supported as an Initiator	Supported as a Target	Definition
Mrd	Yes	Yes	Memory Read Request
MRdLk	No	No	Memory Read Lock. As a target, CplLk with UR status is returned.
MWr	Yes	Yes	Memory Write Request to memory-mapped PCI-Express space
IORd	Yes (RC only)	No	I/O Read request. As a target, Cpl with UR status is returned.
IOWr	Yes (RC only)	No	I/O Write Request. As a target, Cpl with UR status is returned.
CfgRd0	Yes (RC only)	Yes (EP only)	Configuration Read Type 0.
CfgWr0	Yes (RC only)	Yes (EP only)	Configuration Write Type 0.
CfgRd1	Yes (RC only)	No	Configuration Read Type 1. As a target, Cpl with UR status is returned.
CfgWr1	Yes (RC only)	No	Configuration Write Type 1. As a target, Cpl with UR status is returned.
Msg	Yes	Yes	Message Request. Message without data is not forwarded to memory.
MsgD	Yes (RC only)	Yes (EP only)	Message Request with Data payload.
Cpl	Yes	Yes	Completion without Data

Table continues on the next page...

Table 19-5. PCI Express Transactions (continued)

PCI Express Transaction	Supported as an Initiator	Supported as a Target	Definition
CplD	Yes	Yes	Completion with Data
CplLk	No	Yes	Completion for Locked Memory Read without Data. The only time that CplLk is returned with UR status is when the controller receives a MRdLk command.
CplDLk	No	No	Completion for Locked Memory Read with Data

19.6.1.2 Lane Reversal

PCI Express lane reversal is supported as part of the PCI Express training operations where the lanes are reversed between the two chips at the end of a PCI Express link. This can occur because of chip packaging, pinning or board placement. Lane reversal is performed automatically as part of the lane-numbering negotiation after the link-width negotiation is complete.

The following table describes the supported configurations for the x4 controller.

Table 19-6. x4 controller lane reversal

Link configuration	Lane 0	Lane 1	Lane 2	Lane 3
x4 link without lane reversal	0	1	2	3
x2 link without lane reversal	0	1	—	—
x4 link with lane reversal	3	2	1	0
NOTE: The numbers shown in this table (0-3) are the lane numbers assigned to each lane as a result of link initialization and configuration.				

NOTE

The maximum number of lanes used in the lane reversal case cannot exceed the maximum number of lanes allocated for the PCI Express interface in use as determined by the SerDes options RCW settings (SRDS_PRTCL_Sn).

19.6.1.3 Transaction ordering rules

In general, transactions are serviced in the order that they are received.

However, transactions can be reordered as they are sent due to a stalled condition such as a full internal buffer. The following table describes the transaction ordering rules for this device.

Table 19-7. PCI Express controller TLP transaction ordering rules

Row pass column?		Posted request	Non-posted request		Completion	
		Memory write or message request (col 2)	Read request (Col 3)	I/O or Configuration write request (col 4)	Read completion (Col 5)	I/O or configuration write completion (col 6)
Posted request	Memory write or message request (row A)	a) No[1] b) No[1]	Yes	Yes	a) Yes[2] b) N/A[3]	a) Yes[2] b) N/A[3]
Non-posted request	Read request (row B)	No	No	No	a) No[4] b) Yes[5]	a) No[4] b)Yes[5]
	I/O or configuration write request (Row C)	No	No	No	a) No[4] b) Yes[5]	a) No[4] b) Yes[5]
Completion	Read completion (row D)	a) No[6] b) Yes[7]	Yes	Yes	a) No[8] b) No[8]	No
	I/O or configuration write completion (row E)	a) No[6] b) Yes[7]	Yes	Yes	No	No

1. Regardless of the setting of the relaxed ordering (RO) bit, a posted request cannot bypass another posted request.
2. Regardless of the setting of the relaxed ordering bit, a posted request can always bypass a completion.
3. N/A indicates that the original rules at these entries defined by the *PCI Express Base Specification* do not apply to RC or EP.
4. A non-posted request cannot bypass a completion if the relaxed ordering bit is cleared (that is, RO = 0).
5. A non-posted request can bypass a completion if the relaxed ordering bit is set (that is, RO = 1).
6. A read completion, I/O write completion, or configuration write completion cannot bypass a posted request if the relaxed ordering bit is cleared (that is, RO = 0).
7. A read completion, I/O write completion, or configuration write completion can bypass a posted request if the relaxed ordering bit is set (that is, RO = 1).
8. Regardless of the setting of the relaxed ordering bit, a read completion cannot bypass another read completion.

In general, the following points summarize the ordering rules for sending the next outstanding request:

- A posted request can bypass all other transactions except another posted request.
- A non-posted request cannot bypass posted or other non-posted requests, but it can bypass a completion if the relaxed ordering (RO) bit is set.(See Table 19-7).
- A completion can bypass posted requests if the relaxed ordering (RO) bit is set and can bypass non-posted transactions. However, a completion cannot bypass other completions.

19.6.1.4 Internal Address Translation Unit

The core uses the iATU to replace the TLP address and TLP header fields in the current TLP request header. This section presents the following topics:

- iATU Overview
- Programming the iATU
- Outbound iATU Operation
- Inbound iATU Operation

19.6.1.4.1 iATU Overview

Address translation is used for mapping different address ranges to different memory spaces supported by your application. A typical example maps the internal platform memory space to PCI memory space. The iATU supports type translation. Without address translation, your application address is passed to/from the PCIe TLPs directly through the internal platform . You can program the iATU to implement your own address translation scheme without the need for additional external hardware.

Inbound Features:

- Match mode operation for MEM TLPs. No translation for completions. Selectable BAR Match mode operation for MEM TLPs.

  NOTE

  Address match mode can only be used in RC mode; BAR match mode must be used in EP mode.

  - TLPs destined for configuration registers in an upstream port are not translated.
  - TLPs that are not error-free (ECRC, malformed and so on) are not translated.
- Programmable TLP header field matching.
  - TYPE
- Function number (physical and virtual).
- 24 address regions programmable for location and size.
- Programmable enable/disable per region.
- Automatic FMT field translation between three DWORDs and four DWORDs for 64-bit addresses.
- ECAM Configuration Shift mode to allow a 256 MB CFG1 space to be located anywhere in the 64-bit address space.
- Supports regions from 4 kB to 4 GB in size.

Outbound Features:

- Address Match mode operation for MEM and I/O, CFG, and MSG TLPs. No translation for completions.
- Supports type translation through TLP type header field replacement for MEM or I/O types to MSG/CFG types.
  - Includes posted to non-posted translation (for example, MWr to CfgWr0)
  - No translation from completions
- Programmable TLP header field replacement.
  - TYPE/TD/TC/ATTR/MSG-Code
  - Function number (physical and virtual)
- 256 address regions programmable for location and size.
- Programmable enable/disable per region.
- Automatic FMT field translation between three DWORDs and four DWORDs for 64-bit addresses.
- Configuration Shift mode. Optimizes the memory footprint of CFG accesses destined for the internal platform interface in a multifunction device.
- Response code which defines the completion status to return for accesses matching a region.
- Supports regions from 4 kB to 4 GB in size.

NOTE

The default behavior of the ATU when there is no address match in the outbound direction or no TLP attribute match in the inbound direction, is to pass the transaction through.

19.6.1.4.2 Programming the iATU

You can access the iATU registers through the DBI interface or through PCIe CFG accesses. The following registers are used for programming the iATU.

Table 19-8. iATU Register Map

Byte Offset	Description
0x900	iATU Index Register
0x904	iATU Region Control 1 Register
0x908	iATU Region Control 2 Register
0x90C	iATU Region Lower Base Address Register
0x910	iATU Region Upper Base Address Register
0x914	iATU Region Limit Address Register
0x918	iATU Region Lower Target Address Register
0x91C	iATU Region Upper Target Address Register
0x920	iATU Region Control 3 Register

The iATU registers are programmed through an indirect addressing scheme (using an index register) to reduce the address footprint in the PCI Express extended configuration space. The index register has a “Region Direction” bit to determine whether an inbound or outbound region is being accessed and a “Region Index” field to determine which region to program/read when accessing the other address translation registers.

19.6.1.4.3 Outbound iATU Operation

This section describes the processing of outbound requests by the iATU. This section presents the following topics:

- Overview (Address Match Mode)
- RID BDF Number Replacement
- iATU Outbound MSG Handling
- CFG Handling
- CFG Shift Feature
- FMT Translation
- No Address Match Result
- Writing to a MRdLk Region
- Outbound Programming Example

19.6.1.4.3.1 Overview (Address Match Mode)

The address field of each request MEM and I/O TLP is checked to see if it falls into any of the enabled address regions defined by the “Start” and “End” addresses as defined in Figure 19-5.

NOTE

When the “Region Enable” bit of the Region Control 2 register is “0”, then that region is not used for address matching.

When an address match is found, then the TLP address field is modified as follows: Address = Address - Base Address + Target Address and the TYPE, TD, TC, Function Number, and ATTR header fields are replaced with the corresponding fields in the “iATU Control 1 Register” (IATU_REGION_CTRL_1_OFF_OUTBOUND_0). When your application internal platform address field matches more than one of the 256 address regions, then the first (lowest of the numbers from 0 to 255) enabled region to be matched is used. See No Address Match Result for details on what happens when there is no address match. This operational mode (called “Address Match Mode“) is always used for outbound translation.

Figure 19-5. iATU Address Region Mapping: Outbound and Inbound (Address Match Mode), 64-bit Address

When the PCIe core is operating with 32-bit addresses, then the operation is defined as in figure below.

Figure 19-6. iATU Address Region Mapping: Outbound and Inbound (Address Match Mode), 32-bit Address

The upper 32 bits of the Target Address register always forms the upper 32 bits of the translated address because:

- The maximum region size is 4 GB.
- A region must not cross a 4 GB boundary.

In the figure, CX_ATU_MIN_REGION_SIZE specifies the minimum size of an address translation region and is 4 KB.

19.6.1.4.3.2 RID BDF Number Replacement

When there is a successful address match on an outbound TLP, then the function number used in generating the function part of the requester ID field of the TLP is taken from the 3-bit “Function Number” field of the “iATU Control 1 Register” (IATU_REGION_CTRL_1_OFF_OUTBOUND_0). The value in this field must be 0x0 unless multifunction operation is enabled (the number of PFs is greater than 1).

NOTE

The requester ID uses the 8:5:3 bit PCI Bus.Device.Function (BDF) format.

19.6.1.4.3.3 iATU Outbound MSG Handling

NOTE

The iATU only supports generating outbound messages with data (MsgD); messages without data are not supported. The MWr transactions to be translated must have a non-zero effective length.

The iATU supports TYPE translation/conversion of MEM and I/O TLPs to MsgD TLPs. This supports applications that are unable to directly generate MsgD TLPs. When there is a successful address match on an outbound MEM TLP, and the translated TLP type field is MSG (that is, the type field of the “iATU Control 1 Register” IATU_REGION_CTRL_1_OFF_OUTBOUND_0 is “10xxx”), then the message code field of the TLP is set to the value in the “Message Code” field of the “iATU Control 2 Register” (IATU_REGION_CTRL_2_OFF_OUTBOUND_0).

19.6.1.4.3.4 CFG Handling

The iATU supports translation of I/O and MEM TLPs to CFG TLPs. The 16-bit Bus.Device.Function (BDF) is derived from bits [31:16] of the “iATU Lower Target Address Register” (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0). You can shift/remap the BDF within the PCIe address space using the CFG Shift Feature. Address translation of CFG TLPs is also possible with the iATU.

19.6.1.4.3.5 CFG Shift Feature

This expander feature can be enabled by setting the “CFG Shift” bit of the “iATU Control 2 Register” (IATU_REGION_CTRL_2_OFF_OUTBOUND_0). This shifts/maps the BDF, which is bits [27:12] of the target address, up to bits [31:16] of the translated

address. This allows all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped into any 256 MB region of the PCIe address space. This scheme also supports the Enhanced Configuration Address Mapping (ECAM) mechanism from Section 7.2.2 of the PCI Express Base 3.0 Specification, revision 1.0. ECAM supports the mapping (MEM to CFG translation) from memory address space to PCIe configuration space address as defined in Table 19-9. ECAM maps bits [27:12] of the untranslated MEM TLP to become the BDF of the resulting CFG TLP.

Table 19-9. ECAM Mapping

Memory Address Bits	PCIe Configuration Space
27:20	Bus Number
19:15	Device Number
14:12	Function Number
11:8	Extended Register Number
7:2	Register Number

19.6.1.4.3.6 FMT Translation

The iATU automatically sets the TLP format field for three DWORDs when it detects all zeroes in the upper 32 bits of the translated address. Otherwise, it sets it to four DWORDs when it detects a 64-bit address (that is, when there is a “1” in the upper 32 bits of the translated address). When the original address and the translated address are of different format, the iATU ensures that the TLP header size matches the translated address format.

19.6.1.4.3.7 No Address Match Result

When there is no address match then the address is untranslated but the TLP header information comes from the relevant fields on the internal platform interface.

19.6.1.4.3.8 Writing to a MRdLk Region

When there is a successful address match for an outbound write and the type header field in the “iATU Control 1 Register” is “ 00001” indicating a locked MEM transfer, then the core sets the type field to “0000” (MEM).

19.6.1.4.3.9 Outbound Programming Example

Define outbound region 1 as a 64 kB I/O region from 0x8000_0000_D000_0000 to 0x8000_0000_D000_FFFF, to be mapped to 0x00010000 in the PCIe I/O space.

1. Setup the Index Register.

   Write 0x00000001 to Address { 0x700 + 0x200 } to set outbound region 1 as the current region

2. Setup the Region Base and Limit Address Registers.

   Write 0xd0000000 to Address {0x700 +0x20C} to set the Lower Base Address.

   Write 0x80000000 to Address {0x700 +0x210} to set the Upper Base Address.

   Write 0xd000ffff to Address {0x700 +0x214} to set the Limit Address.

3. Setup the Target Address Registers.

   Write 0x00010000 to Address {0x700 +0x218} to set the Lower Target Address.

   Write 0x00000000 to Address {0x700 +0x21C} to set the Upper Target Address.

4. Configure the region through the Region Control 1 Register.

   Write 0x00000002 to Address {0x700 +0x204} to define the type of the region to be I/O.

5. Enable the region.

   Write 0x80000000 to Address {0x700 +0x208} to enable the region.

19.6.1.4.4 Inbound iATU Operation

This section discusses how the iATU processes inbound requests. The topics are:

- Overview
- MEM Match Modes
- CFG Handling (Upstream Port)
- FMT Translation
- Inbound Programming Example

19.6.1.4.4.1 Overview

NOTE

- The main difference between inbound and outbound iATU operation is that the TLP type is never changed in the inbound direction. Instead, the type field is used for more

precise matching. Other fields can also be optionally used to further refine the matching process.

- Another difference is that for MEM TLPs, you can select between address matching (as used in outbound operation) or BAR matching. Normally, an Endpoint uses BAR match mode, and a Root Complex uses address mode as a Root Complex normally does not implement BARs.

The following translation rules and limitations apply:

- When there is no match, then the address is untranslated. In addition
- TLPs destined for the configuration registers in an upstream port are not translated.
- TLPs that are not error-free (ECRC, malformed and so on) are not translated.
- Address translation of all TLP types (MEM, CFG, and MSG) except completion is supported in Address Match mode. In BAR Match mode, only translation of MEM is supported.

  The setting of the “Match Mode” field in the “iATU Control 2 Register” (IATU_REGION_CTRL_2_OFF_ OUTBOUND_0) determines how iATU inbound matching is done for each TLP type.

Table 19-10. Determination of Match Mode By TLP Type and “Match Mode” Field of iATU Control 2 Register

	Match Mode = 0	Match Mode = 1
MEM	Address Match Mode	BAR Match Mode
CFG0	Routing ID Match Mode	Accept Mode
MSG/MSGD	Address Match Mode	Vendor ID Match Mode

19.6.1.4.4.2 MEM Match Modes

Inbound address translation for MEM TLPs operates in one of two matching modes as determined by the “Inbound Match Mode” field in the “iATU Control 2 Register”.

19.6.1.4.4.2.1 Address Match Mode

The operation is similar to Outbound iATU Operation. The address field of each request TLP is checked to see if it falls into any of the enabled address regions as shown in Figure 19-5 .

NOTE

When the “Region Enable” bit of the “Region Control 2 Register” is”0”, then that region is not used for address matching.

NOTE

The target (translation) address and base address of the inbound iATU window must be aligned based on the window size programmed in the corresponding BAR MASK register.

When an address match is found then the TLP address field is modified as follows:

Address = Address - Base Address + Target Address

When the TLP address field matches more than one of the 24 address regions, then the first (lowest of the numbers from 0 to 23) enabled region to be matched is used.

19.6.1.4.4.2.2 BAR Match Mode

Looking for an address match is a two-step process.

1. The standard internal PCI Express BAR Matching Mechanism checks if the address field of any MEM request TLP falls into any address region defined by the enabled BAR addresses and masks.
2. When a matched BAR is found, then the iATU compares the BAR ID to the “BAR Number” field in the “iATU Control 2 Register” for all enabled regions.Figure 19-7 and Figure 19-8 provide more details on inbound translation in BAR Match Mode.

NOTE

BAR Match Mode can only be used for MEM transactions

When the PCIe core is operating with 64-bit BARs, the operation is defined as in figure below.

Figure 19-7. iATU Address Region Mapping: Inbound (BAR Match Mode), 64-bit BAR

In address match mode, when the address range does not match one of its BAR ranges in an upstream port, then the device rejects the request with unsupported request (UR) completion status and no translation occurs.

When the PCIe core is operating with 32-bit BARs, the operation is defined as in figure below.

Figure 19-8. iATU Address Region Mapping: Inbound (BAR Match Mode), 32-bit BAR

19.6.1.4.4.3 CFG Handling (Upstream Port)

The PEX controller normally routes CFG TLPs to the configuration registers without translating them. The iATU only translates CFG0 TLPs that the PEX controller has routed to the internal platform. Inbound address translation for CFG0 TLPs operates in one of two matching modes as determined by the “Inbound CFG0 Match Mode” field in the “iATU Control 2 Register”.

Routing ID Match Mode: The operation is similar to Outbound iATU Operation. The routing ID of the inbound CFG0 TLP must fall within the Base and Limit of the defined iATU region for matching to proceed. The iATU interprets the routing ID (Bytes 8-11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM and I/O transactions.

Accept Mode: The PEX controller always accepts CFG0 TLPs even when the CFG bus number does not match the current bus number of the device. This mode follows that behavior. The routing ID of received CFG0 TLPs are ignored when determining a match.

CFG1 Transactions: For CfgRd1/CfgWr1 transactions, the base and limit addresses could enclose the entire 32-bit 4 GB memory space with the routing ID forming the upper 16 bits. The target address maps these CFG transactions to anywhere in application address space.

19.6.1.4.4.4 FMT Translation

The iATU automatically sets the TLP format field for three DWORDs when it detects all zeroes in the upper 32 bits of the translated address. Otherwise it sets it to four DWORDs when it detects a 64-bit address (when there is a “1” in the upper 32 bits of the translated address). When the original address and the translated address are of a different format then the iATU ensures that the TLP header size matches the translated address format.

19.6.1.4.4.5 Inbound Programming Example

19.6.1.4.4.5.1 BAR match mode

Define inbound region 2 as MEM region matching BAR4 (BAR match mode) mapping to 0x8000_0000_2000_0000 in your application memory space

1. Setup the Index Register.

   Write 0x80000002 to Address { 0x700 + 0x200 } to set inbound region 2 as the current region.

2. Setup the Target Address Registers.

Write 0x20000000 to Address {0x700 + 0x218} to set the Lower Target Address.

Write 0x80000000 to Address {0x700 + 0x21C} to set the Upper Target Address.

3. Configure the region through the Region Control 1 Register.

   Write 0x00000000 to Address {0x700 + 0x204} to define the type of the region to be MEM.

4. Enable the region for BAR Match Mode.

   Write 0xC0000400 to Address {0x700 + 0x208} to enable the region for BAR match mode for BAR4.

19.6.1.4.4.5.2 Address match mode

Define inbound Region 0 as: MEM region matching TLPs with addresses in the range 0x00010000 to 0x0005ffff mapped to 0x1000_0000_2000_0000 - 0x1000_0000_2004_ffff in your application memory space

1. Setup the Index Register.

   Write 0x80000000 to Address { 0x700 + 0x200 } to set inbound region 0 as the current region

2. Setup the Region Base and Limit Address Registers.

   Write 0x00010000 to Address {0x700 + 0x20C} to set the Lower Base Address.

   Write 0x00000000 to Address {0x700 + 0x210} to set the Upper Base Address.

   Write 0x0005ffff to Address {0x700 + 0x214} to set the Limit Address

3. Setup the Target Address Registers.

   Write 0x20000000 to Address {0x700 + 0x218} to set the Lower Target Address.

   Write 0x10000000 to Address {0x700 + 0x21C} to set the Upper Target Address.

4. Configure the region through the Region Control 1 Register.

   Write 0x00000000 to Address {0x700 + 0x204} to define the type of the region to be MEM.

5. Enable the region.

   Write 0x80000000 to Address {0x700 + 0x208} to enable the region in address match mode.

NOTE

- EP port: Defined MEM regions must be inside an enabled BAR range.
- RC port: Defined MEM regions must either match a BAR or be outside of the base and limit ranges defined for the port in the Type 1 configuration header.

19.6.1.5 Memory Space Addressing

A PCI Express memory transaction can address a 32- or 64-bit memory space. As an initiator, the controller is capable of sending 32- or 64-bit memory packets. Any transaction from the internal platform that (after passing through the translation mechanism) has a translated address greater than 4G is sent as a 64-bit memory packet. Otherwise, a 32-bit memory packet is sent. As a target device, the controller is capable of decoding 32- or 64-bit memory packets. This is done through two 32-bit inbound windows and two 64-bit inbound windows. All inbound addresses are translated to 40-bit internal platform addresses.

Table 19-11. PCI Express Memory Transactions

Transaction	Type[4:0]	FMT[1]	FMT[0]
MRd (32-bit address)	00000	0	0
MRd (64-bit address)	00000	0	1
MWr (32-bit address)	00000	1	0
MWr (64-bit address)	00000	1	1

19.6.1.6 I/O Space Addressing

As an initiator, the controller can send I/O transactions in RC mode only. This can be done by programming one of the outbound translation window's attribute to send I/O transactions. The controller does not support I/O transactions as a target.

Table 19-12. PCI Express I/O Transactions

Transaction	Type	FMT[1]
IORd	00010	0
IOWr	00010	1

The PCI Express Base Specification restricts I/O space to 4 GB (32-bit I/O address).

19.6.1.7 Configuration Space Addressing

The controller can generate outbound configuration transactions in RC mode only. The controller does not support inbound configuration transactions in RC mode.

The controller accepts inbound configuration transactions in EP mode only. The controller does not support generating outbound configuration transactions in EP mode.

Table 19-13. PCI Express Configuration Transactions

Transaction	Type	FMT[1]
CfgRd0	00100	0
CfgWr0	00100	1
CfgRd1	00101	0
CfgWr1	00101	1

Note that all configuration transactions sent on PCI Express require a response, regardless whether they are read or a write configuration transactions.

19.6.1.8 Messages

This section describes outbound message generation and inbound message reception in RC and EP modes.

19.6.1.8.1 Outbound Message Generation

Software can generate the following outbound message transactions:

Outbound Message	Generated by
Set_Slot_Power_Limit	Outbound iATU translation to MsgD with the appropriate encoding.
Set_Slot_Power_Limit	Writing to the Slot Power Limit Scale and Slot Power Limit Value fields of the Slot Capabilities Register in the PCI Express Capabilities Structure.

19.6.1.8.2 Inbound Messages

The following table provides a complete list of supported inbound messages in RC mode.

Table 19-14. PCI Express RC Inbound Message Handling

Name	Code[7:0]	Routing[2:0]	Action
INTx Interrupt Signaling			
Assert_INTA	0010 0000	100	Assert INTA virtual wire. Sent to GIC.
Assert_INTB	0010 0001	100	Assert INTB virtual wire. Sent to GIC.
Assert_INTC	0010 0010	100	Assert INTC virtual wire. Sent to GIC.
Assert_INTD	0010 0011	100	Assert INTD virtual wire. Sent to GIC.
Deassert_INTA	0010 0100	100	De-assert INTA virtual wire. Sent to GIC.
Deassert_INTB	0010 0101	100	De-assert INTB virtual wire. Sent to GIC.
Deassert_INTC	0010 0110	100	De-assert INTC virtual wire. Sent to GIC.
Deassert_INTD	0010 0111	100	De-assert INTD virtual wire. Sent to GIC.
Power Management			
PM_PME	0001 1000	000	Generate interrupt to GIC
PME_TO_Ack	0001 1011	101	Generate interrupt to GIC
Error Signaling			
ERR_COR	0011 0000	000	Generate interrupt to GIC
ERR_NONFATAL	0011 0001	000	Generate interrupt to GIC
ERR_FATAL	0011 0011	000	Generate interrupt to GIC

The following table provides a complete list of supported inbound messages in EP mode.

Table 19-15. PCI Express EP Inbound Message Handling

Name	Code[7:0]	Routing[2:0]	Action
Power Management			
PME_Turn_Off	0001 1001	011	
Slot Power Limit Support			
Set_Slot_Power_Limit	0101 0000	100	Update power value in PCI Express device capability register in configuration space.

19.6.1.9 Error Handling

The PCI Express specification classifies errors as correctable and uncorrectable. Correctable errors result in degraded performance, but uncorrectable errors generally result in functional failures. As shown in the figure below, uncorrectable errors can further be classified as fatal or non-fatal.

Figure 19-9. PCI Express Error Classification

19.6.1.9.1 PCI Express Error Logging and Signaling

The figure below shows the PCI Express-defined sequence of operations related to signaling and logging of errors detected by a device. Note that the PCI Express controller on this device supports the advanced error handling capabilities shown within the dotted lines.

Figure 19-10. PCI Express Device Error Signaling Flowchart

19.6.2 Interrupts

Both message signaled interrupts (MSI) and legacy INTx are supported.

- Legacy INTx interrupts are handled by virtual-wire message transactions. See Inbound Messages.

19.6.3 Initial Credit Advertisement

To prevent overflowing of the receiver's buffers and for ordering compliance purposes, the transmitter cannot send transactions unless it has enough flow control (FC) credits to send. Each device maintains an FC credit pool. The FC information is conveyed between the two link partners by DLLPs during link training (initial credit advertisement). The transaction layer performs the FC accounting functions. One FC unit is four DWs (16-bytes) of data.

Table 19-16. Initial credit advertisement

Credit Type	Initial Credit Advertisement
PH (Memory Write, Message Write)	8
PD (Memory Write, Message Write)	128
NPH (Memory Read, IO Read, Cfg Read, Cfg Write)	12
NPD (IO Write, Cfg Write)	2
CPLH (Memory Read Completion, IO R/W Completion, Cfg R/W Completion)	Infinite
CPLD (Memory Read Completion, IO Read Completion, Cfg Read Completion)	Infinite

19.6.4 Power Management

All device power states are supported with the exception of D3cold. Only L0s ASPM mode is supported if enabled by configuring the Link Control register's bits 1-0 in configuration space. Note that there is no power saving in the controller when the device is put into a non-D0 state. The only power saving is the I/O drivers when the controller is put into a non-L0 link state.

Table 19-17. Power Management State Supported

Component D-State	Permissible Interconnect State	Action
D0	L0, L0s	In full operation.
D1	L0, L0s, L1	All outbound traffic is stalled. All inbound traffic is thrown away. The only exceptions are PME messages and configuration transactions.
D2	L0, L0s, L1	All outbound traffic is stalled. All inbound traffic is thrown away. The only exceptions are PME messages and configuration transactions.

Table continues on the next page...

Table 19-17. Power Management State Supported (continued)

Component D-State	Permissible Interconnect State	Action
D3hot	L0, L0s, L1, L2/L3 Ready	All outbound traffic is stalled. All inbound traffic is thrown away. The only exceptions are PME messages and configuration transactions. Note that if a transition of D3hot to D0 occurs, a reset is performed to the controller's configuration space. In addition, link training restarts.
D3cold	L3	Completely off.

19.6.4.1 L2/L3 Ready Link State

The L2/L3 Ready link state is entered after the EP device is put into a D3hot state followed by a PME_Turn_Off/PME_TO_Ack message handshake protocol. Exiting this state requires a POR reset or a WAKE_B signal from the EP device. The PCI Express controller (in EP mode) does not support the generation of beacon; therefore, as an alternative, the device can use one of the GPIO signals as an enable to an external tristate buffer to generate a WAKE_B signal, as shown in the figure below.

Figure 19-11. WAKE_B Generation Example

In RC mode, the WAKE_B signal from the EP device can be connected to one of the external interrupt inputs to service the WAKE_B request.

19.6.5 Hot Reset

When a hot reset condition occurs, the controller (in both RC and EP mode) initiates a clean-up of all outstanding transactions and returns to an idle state. All configuration register bits that are non-sticky are reset. Link training takes place subsequently. The device is permitted to generate a hot reset condition on the bus when it is configured as an RC device by setting the "Secondary Bus Reset" bit in the Bridge Control Register in

the configuration space. As an EP device, it is not permitted to generate a hot reset condition; it can only detected a hot reset condition and initiates the clean-up procedure appropriately.

19.6.6 Alternative Routing ID (ARI, SR-IOV only)

ARI is not applicable to Root Complex; all other SR-IOV Capable Devices (Devices that include at least one PF) shall implement the ARI Capability in each Function. PF Routing IDs are determined based on the Next Function Number in the Alternative Routing ID Interpretation Extended Capability (ARI) register. VF Routing IDs are determined using the First VF Offset and the VF Stride configuration settings.

19.6.7 Functional Level Reset (FLR, SR-IOV only)

An FLR reset can be issued for a PF or VF by software by setting the IFLR bit in PCI Express device control register associated with the PF/VF. The controller will finish processing all outstanding transactions associated with the PF/VF and return to an idle state. Any new transactions targeted toward the PF or VF while the IFLR bit is set will be dropped (posted transactions) or errored out (outbound non-posted transactions) or returned UR (inbound non-posted transactions) until the IFLR bit is cleared by hardware. Traffic that are toward PF or VF without the IFLR bit set continue as before.

19.7 Initialization/Application Information

19.7.1 Gen 3 Link Equalization

Link equalization allows components to adjust the transmitter and receiver setup of each lane to improve the signal quality and meet the requirements for operating at Gen 3 data rates. However, link equalization cannot be performd in loopback mode. Before attempting link training in loopback mode link equalization must be disabled by setting the "Equalization Disable" bit in the Gen3 Control Register at PCI Express extended configuration space offset 890h.

19.7.2 Configuring the chip for inbound CCSR accesses

The chip must be configured using the following procedure prior to receiving any inbound memory transactions targeting CCSR space:

1. Program COHERENCY_CONTROL_3_OFF (offset 8E8h) = 0000_0000h
2. Program COHERENCY_CONTROL_2_OFF (offset 8E4h) = 0000_0000h
3. Program COHERENCY_CONTROL_1_OFF (offset 8E0h) = 1000_0000h

This procedure establishes a boundary between CCSR and memory at 1000_0000h in the system memory map.

NOTE

Any inbound memory transactions targeting CCSR space must set NO_SNOOP = 1 in the TLP.

19.7.3 Poisoned TLP handling

In some cases, the handling of inbound Poisoned TLPs needs special attention. Inbound Poisoned TLP requests are dropped and the Poisoned TLP flag in the Uncorrectable Error Status Register (bit 12) is set, provided it is not masked. However, for an inbound Poisoned TLP response to an outbound request, the response is forwarded to the application as a completion and the Poisoned TLP flag in the Uncorrectable Error Status Register (bit 12) is set (again, provided it is not masked). This allows the requesting transaction to terminate without stalling, although the error is flagged. In this case, the error handler should cause the response to be discarded.

Chapter 20
Power Management

20.1 The PMU registers as implemented on the chip

DPAA2 PMU supports 5 single threaded e200 cores. The AIOP cores are arranged as clusters of four cores each and the Management complex cores are arranged as a single cluster.

Table 20-1. DPAA PMU Cores as implemented on chip

Register Bit Number	Core number	Assignment
0	0	Management Complex Core
1-4	1-4	AIOP Core

Table 20-2. DPAA PMU Clusters as implemented on chip

Register Bit Number	Cluster number	Assignment
0	0	Management Complex Core Cluster
1	1	AIOP Core Cluster

20.2 Overview

The Power Management Unit (PMU) performs all device-level tasks associated with device power management.

The PMU communicates with embedded cores and coherency modules extensively to achieve both device run control and power management function.

This document focuses solely on the mechanisms of the PM unit that control the power managment aspects of the device and each of the processor clusters/cores/threads within the device.

This device provides a set of hierarchical power management states to significantly reduce dynamic power consumption at the processor core, cluster, and overall device. The device also supports various means of wakeup (for example network traffic, interrupts, timers) from these power managed states to return to run mode.

20.3 Arm GPP Cores and Cluster Level Power Management

20.3.1 Central SoC Power Management Unit Features

- GPP Software controlled power management, which minimizes power consumption of blocks when they are idle.
- GPP Software controlled cluster PCL10/PCL15/PCL20/PCL30 mode and device LPM10/20/30 mode.
- Supports Arm GIC interrupt based wake-up and the following wake-up sources:
  - Wake on GPP and DPAA core debug events.
  - Wake on device debug events.
  - Wake on GPP core and cluster reset events
  - Wake on device reset events.
- Arm CPU Power Management:
  - Independent Power Management Control of each CPU
  - STANDBYWFI (PW15) State where:
    - Clocks are internally gated in the Arm CPU except when servicing snoops.
  - STANDBYWF*-retain (PW20) where:
    - CPU voltage is lowered to a level where all CPU state is retained while leakage is reduced.
- Arm Cluster Power Management:
  - Independent Power Management Control of each cluster
  - STANDBYWFIL2 (PCL10) State where:

20.3.2 Arm GPP Core Power Management State Summary- Core and Core Cluster level

The following tables summarize the core, cluster, domain and device PM modes.

NOTE

A resumable power state in the core indicates that the core can exit from a power managed state and return to Full On without the core(s) being reset.

Table 20-3. Arm GPP Core and Cluster Power Management State Summary

Power Management State Name	Resumable	Services Snoops	Device Clocks	Core Clocks	Core Voltage	State Retained	Status visible	Description
STANDBYWF* (PW15)	Y	Y	Y	N	Y	Y	TWAITSR0 or PCWFESR0	The *core* referred to is currently in the ... in the STANDBYWF* (PW15) state.
STANDBYWF*-retain (PW20)	Y	Y	Y	N	N	Y	PCPW20SR	The core referred to is currently in a power gating with state retention state.

Table 20-4. GPP Cluster Power Management State Summary

Power Management State Name	Resumable	Services Snoops	Device Clocks	Core Clocks	State Retained for Core PH20	Book E Core State Entry	Description
STANDBYWFIL2 (PCL10)	Y	N	Y	N	Y	CLPCL10SR	The cluster referred to is in a STANDBYWFIL2 state

20.3.3 Arm Power Management

20.3.3.1 Arm Overview

The Arm cores cluster does not support PMU initiated low power modes. The Arm cores cluster support low power initiated by Arm ISA and Arm dynamic power management logic. The Arm cores cluster power management is described in the Arm Core Technical Reference Manuals. The following table summarizes the Arm cores cluster low power modes.

Table 20-5. Thread/Core/Cluster Low Power Modes

Arm Arch	Comments
STANDBYWFI (PW15)	While the Arm CPU is in WFI state the clocks in the CPU can dynamically and temporarily enabled to service snoops or debug resources access.
STANDBYWF*-retain (PW20)	It is possible to acheive STANDBYWFE-retain state, however Arm does not recommend retention state with WFE due to the short-term nature of and high likelihood of more frequent or spurious WFE exit conditions. For longer term retention power savings, it is recommended to use STANDBYWFI-retain (PW20).

Table continues on the next page...

Table 20-5. Thread/Core/Cluster Low Power Modes (continued)

STANDBYWFIL2 (PCL10)	While the Arm processor is in L2 WFI state all internal clocks in the processor are gated except for the PCLKDBG.

20.3.3.2 Arm GPP Core PM State: Entry and Exit

Arm GPP Core PM State: Entry and Exit for Power Management [1]		
Power Management State Name	Entry via ...	Exit via ...
STANDBYWFI (PW15)	1. Set the CPUECLTR[CPURETCTL]2 to a zero value [2] 2. Execute WFI (wait for interrupt) instruction	• STANDBYWFI permanent exit • STANDBYWFI temporary exit • Debug interrupt without masking Various LPM20 device wake conditions, as listed in the table below .
STANDBYWFE (PW15)	1. Set the CPUECLTR[CPURETCTL] to a zero value 2. Execute WFE (wait for event) instruction [3]	• STANDBYWFE permanent exit • STANDBYWFE temporary exit • Debug interrupt without masking Various LPM20 device wake conditions, as listed in the table below.
STANDBYWFI-retain (PW20)	1. Set the CPUECLTR[CPURETCTL] to a non-zero value 2. Set CPUECTLR[SMPEN] to 1 3. Execute WFI (wait for interrupt) instruction	• STANDBYWFI permanent exit STANDBYWFI temporary exit • Debug interrupt without masking • Various LPM20 device wake conditions, as listed in table below.
STANDBYWFE-retain (PW20)	1. Set the CPUECLTR[CPURETCTL] to a non-zero value Set CPUECTLR[SMPEN] to 1 2. Execute WFE (wait for event) instruction	• STANDBYWFE permanent exit • STANDBYWFE temporary exit • Debug interrupt without masking Various LPM20 device wake conditions, as listed in the table below.
Core Powerdown (PW30) [4]	Core Powerdown (PW30) is reachable but not supported. Since the lowest power state is already acheived for the Arm GPP core is PW20, there is no need to program entry to PW30. PW30 is included for reference only, to highlight the constraint that CPUECTLR[SMPEN] must be set to 1 for entry to PW20. 1. Set the CPUECLTR[CPURETCTL] to a non-zero value 2. Set CPUECTLR[SMPEN] to 0 3. Execute WFI (wait for interrupt) instruction	Core warm reset or any device-level reset.

1. This information is taken from Arm Reference Materials (Armv7 and Armv8 Architectural Reference Manuals, and relevant Arm core-specific Technical Reference Manuals.) This table is provided as summary and is derived from from Arm documentation.

2. Arm Core CPUECTLR_EL1[CPURETCTL]. Since CPUECTLR_EL1[CPURETCTL] relies on the Arm Generic Timer, Arm Generic timer for this core must be enabled and running.
3. Special software programming considerations must be implemented when using WFE. Caveats include, but are not limited to: 1) WFE is typically placed in a spin loop, and 2) the first execution of a WFE may fall-through if the event register is set when the WFE instruction executes. See the Armv8 Architectural Reference Manual and Arm Core Technical Reference Manual for details on WFE.
4. Core Powerdown (PW30) is reachable but not supported. Since the lowest power state is already acheived for the Arm GPP core in PW20, there is no need to program entry to PW30. PW30 is included for reference only, to highlight the constraint that CPUECTLR[SMPEN] must be set to 1 for entry to PW20.

Arm Defined STANDBYWFI and STANDBYWFE Exit Conditions				
Short Notation	Arm Power Management State(s)	FSL Power Management State Name	Type of Exit	Exit Conditions [1]
STANDBYWFI permanent exit	STANDBYWFI	PW15	permanent	• Any interrupt request.[2] • A core debug halt request. • Core warm reset or any device-level reset.
	STANDBYWFI -retain	PW20		
STANDBYWFI temporary exit	STANDBYWFI	PW15	temporary	• A snoop request that must be serviced by the core L1 data cache. • A cache or TLB maintenance operation that must be serviced by the core L1 instruction cache, data cache, or TLB. • An APB access to the debug or trace registers residing in the core power domain.
	STANDBYWFI -retain	PW20		
STANDBYWFE permanent exit	STANDBYWFE	PW15	permanent	• Any interrupt request that is not masked. An external debug request debug event. • Core warm reset or any device-level reset. • EVENTI signal asserted by another CPU in the system. • External global monitor is cleared (CLREXMONREQ).
	STANDBYWFE -retain	PW20		

Table continues on the next page...

Arm Defined STANDBYWFI and STANDBYWFE Exit Conditions				
STANDBYWFE temporary exit	STANDBYWFE	PW15	temporary	• A snoop request that must be serviced by the core L1 data cache. • A cache or TLB maintenance operation that must be serviced by the core L1 instruction cache, data cache, or TLB. • An APB access to the debug or trace registers residing in the core power domain.
	STANDBYWFE -retain	PW20		

1. Exit conditions column is a list. Any one of the items is sufficient for exit.
2. Interrupts may still be masked in the GIC. Masking in the GIC prevents interrupt delivery to the core and therefore does not cause an exit.

20.3.3.3 Arm GPP Cluster PM State: Entry and Exit

Arm GPP Cluster PM State: Entry and Exit		
Power Management State Name	Entry via ...	Exit via ...
	1. On all cores in the cluster entering PCL10, execute the steps in Steps for preparing GPP Arm core for PCL10 or LPM20. 2. Idle the Arm GPP core snoop and ACP interfaces and perform an L2 Flush for the cluster entering PCL10 / LPM20 as described in “Arm GPP Core: Idle Snoop and L2 Flush Sequence”.	1. STANDBYWFI permanent exit 2. Debug Interrupt without masking 3. Various LPM20 device wake conditions .

20.3.3.4 Steps for preparing GPP Arm core for PCL10 or LPM20

1. Clear the appropriate System Control Register C bit, data or unified cache enable, to prevent additional data cache allocation.
2. Disable L2 prefetches by writing a one to bit [38] and zeros to bits [36:35, 33:32] of the CPU Extended Control Register. Execute an ISB instruction to ensure the CPU Extended Control Register write is complete.

3. Execute an ISB instruction to ensure the CPU Extended Control Register write is complete.
4. Execute a DSB instruction to ensure completion of any prior prefetch requests.
5. Clean and invalidate all data from the L1 data cache. This is done with the cisw instruction. The L2 duplicate Snoop Tag RAM for this core is now empty. This prevents any new data cache snoops or data cache maintenance operations from other processors in the processor being issued in this core.
6. Clear the CPUECLTR.SMPEN bit. Clearing the SMPEN bit enables the core to be taken out of coherency by preventing the core from receiving instruction cache, TLB, or BTB maintenance operations broadcast by other processors in the multiprocessor.
7. Note that step (above) of Arm's "Individual core powerdown" sequence is intentionally not included, so that each core may enter a deep power saving mode, STANDBYWFI-retain (PW20). Instead of Arm's step 6, ensure that SMPEN is set to '1'.
8. Ensure that the system does not send interrupts to the core that is being powered down. In addition, mask interrupts in the cores by
   - Setting the each core's PSTATE register D,A,I,F fields to '1'.
   - Writing the GICR_WAKER.ProcessorSleep bits to '1' for all cores in the cluster that are entering PCL10 or LPM20.
9. Set the DBGOSDLR.DLK, Double lock control bit, that forces the debug interfaces to be quiescent.
10. Invalidate all TLB entries (I-side and D-side) at all 3 exception levels:TLBI ALL EL1, TLBI ALL EL2, TLBI ALL EL3.
11. Invalidate the Icache.
12. Execute an ISB instruction to ensure that all of the System register changes from the previous steps have been committed.
13. Execute a DSB instruction to ensure that all instruction cache, TLB and branch prediction operations issued by any core in the processor have completed.
14. For all cores in the cluster entering PCL10 or LPM20, follow the entry sequence to place each of those cores in STANDBYWFI-retain (PW20).

20.3.3.4.1 LPM20 Operation

LPM20 and device debug softstop are two different manners for gating off clocks to the entire device. Both are recoverable but they are used for different purposes. LPM20 is a device low power state.

20.3.3.4.1.1 LPM20 Entry Sequence

Prior to entering LPM20 state (the LPM20 state is ultimately intiatated in the last step below by setting PMU POWMGTCSR[LPM20_RQ] to '1'), the following steps must be performed:

- Software running on the last GPP active core, (must be physical cluster 0), is expected to quiesce all external devices/internal IPs through configuration.
- Some IPs such as WRIOP which do not provide a power management idle/stop interface also have to be idled through software as part of the software initiated power mode entry sequence.
- Idle IP that does not support a PM idle/stop interface. For example, USB and PCIe.
- On all cores except the ‘last active GPP core’, execute the steps in "Steps for preparing GPP Arm core for PCL10 or LPM20".
- On all clusters except the ‘last active cluster’, idle the Arm GPP core snoop and ACP interfaces and perform an L2 Flush.
- On the ‘last active GPP core’, Prepare the EPU .
- On the ‘last active GPP core’, ensure coherency by performing all steps prior to step 11 in "Steps for preparing GPP Arm core for PCL10 or LPM20".
- On the ‘last active GPP core’, enter LPM20 by executing the following sequence:
- Initiate LPM20 request by setting POWMGTCSR[LPM20_RQ] to ‘1’.
- AIOP Cores and clusters are placed by HW in their lowest power state, and clusters in PCL20, and AIOP domain in DM10.
- Execute an ISB instruction to ensure that the additional System register changes from the previous steps have been committed.
- Execute a DSB instruction to ensure any additional D-side accesses related to the previous steps have completed.
    - loop:
    - WFE
    - if (EPU counter == 0) branch to loop // only exit from this loop if this exit from WFE is a wake request event., as captured in the EPU counter.

20.3.3.4.1.2 LPM20 Exit Sequence

Upon receiving an LPM20 exit condition, the code sequence below continues after exiting from the WFI at the end of the Entry sequence

1. Re-enable the GPP ACP and snoop interfaces:
    - On the cluster/core which is managing the PCL10 entry/exit sequence (physical cluster 0, core 0), write CLAINACTCLRR and CLSINACTCLRR.
2. Re-enable the GPP ACP and snoop interfaces:
    - On the cluster/core which is managing the LPM20 entry/exit sequence (physical cluster 0, core 0), write CLAINACTCLRR and CLSINACTCLRR.
3. Re-enable the data cache on each core in the cluster which was in LPM20 by writing the System Control Register C bit ArmCore.SCTLR[C] bit to ‘1’. [optional - SW may choose to defer this later to the operating system.]
4. Re-enable the snoop domain by

- Writing CCN_HN_F[SNOOP_DOMAIN_CTL_SET] to ‘1’,
- Polling CCN_HN_F[SNOOP_DOMAIN_CTL] for ‘1’ for cores in the cluster.

5. Restore DBG:EPU and tMMA state that was modified in the LPM20 entry sequence.
6. Prior to the next steps (ProcessorSleep and later step regarding unmasking interrupts), if software chooses to continue the wake-up process starting at the instruction after WFI, and not continue to the interrupt vector, software should now clear the interrupt status for that wake-up interrupt.
7. Write the GICR_WAKER.ProcessorSleep bits to ‘0’ or their saved value1 for all cores in the cluster that entered LPM20 in order to re-enable the GIC to send interrupts to those cores. The wake-up interrupt also serves as the exit condition from PW15 or PW20 states. This enables the LPM20 wake-up conditions, such as wake-on-LAN, to reach a cluster / GPP core per GIC interrupt table programming.
8. Upon being awakened by the wake-up interrupt, instruction execution will continue at the instruction after the WFIWFE.
9. Unmask or re-enable interrupts that were altered in “Arm GPP: Individual Core Preparation Sequence"
10. Unmask and re-enable interrupts that were altered by setting the each core’s PSTATE register D,A,I,F fields to ‘0’ or their saved value.

20.3.3.4.2 STANDBYWFIL2 (PCL10) Entry and Exit Sequences

All clusters being placed in PCL10 is not supported. Physical cluster 0 must remain active and cannot be placed in PCL10. For deeper power savings, see LPM20 state.

20.3.3.4.2.1 PCL10 Entry Sequence

Please refer Arm GPP Cluster PM State: Entry and Exit

20.3.3.4.2.2 PCL10 Exit Requirements for Arm GPP Clusters

For exiting PCL10 on Arm GPP clusters, in systems where there is another active cluster that can manage cluster(s) in PCL10, software must

- From the cluster/core which is managing the PCL10 entry/exit sequence (physical cluster 0), send an interrupt to one or more cores in the cluster in PCL10 state. This initiates the exit from PCL10 and re-enables the clocks. Those cores not receiving interrupts remain in the state they were placed into prior to entering PCL10 (e.g. PW15, PW20.
- Wait for clocks to be asserted to the cluster in PCL10:
  - Poll PCPCL10SR and wait for it to be deasserted for the cluster in PCL10
- Re-enable the GPP ACP and snoop interfaces:

  - On the cluster/core which is managing the PCL10 entry/exit sequence, write CLAINACTCLRR and CLSINACTCLRR
- Re-enable the data cache on each core in the cluster which was in PCL10 by writing the System Control Register C bit ArmCore.SCTLR[C] bit to ‘1’.
- Re-enable the snoop domain by Writing CCN_HN_F[SNOOP_DOMAIN_CTL_SET] to ‘1’, Polling CCN_HN_F[SNOOP_DOMAIN_CTL] for ‘1’ for cores in the cluster.
- Prior to the next steps (ProcessorSleep and later step regarding unmasking interrupts), if software chooses to continue the wake-up process starting at the instruction after WFI, and not continue to the interrupt vector, software should now clear the interrupt status for that wake-up interrupt.
- Write the GICR_WAKER.ProcessorSleep bits to ‘0’ or their saved value1 for all cores in the cluster that entered PCL10 or LPM20 in order to re-enable the GIC to send interrupts to those cores. The wake-up interrupt also serves as the exit condition from PW15 or PW20, or PH20 states.
- Upon being awakened by the wake-up interrupt, since interrupts were masked in Steps for preparing GPP Arm core for PCL10 or LPM20, instruction execution will continue at the instruction after the WFI.
- Unmask and re-enable interrupts that were altered in “Steps for preparing GPP Arm core for PCL10 or LPM20, by setting the each core’s PSTATE register D,A,I,F fields to ‘0’ or their saved value.

20.3.3.5 Memory Map and Register Definition

This section describes the memory map for the Power Management Unit block including the memory map tables.

20.3.3.5.1 PMU_Arm register descriptions

This section identifies Power Management resources that are not included as part of a processor core or platform IP.

This section identifies Power Management resources that are included as part of Arm processor power management.

20.3.3.5.1.1 PMU_Arm CCSR memory map

PMU_Arm base address: 1E3_0000h

Offset	Register	Width (In bits)	Access	Reset value
40h	Core Wait Status Register (TWAITSR0)	32	RO	0000_0000h
50h	Core Watchdog Detect Status Register (TWDTDSR0)	32	RO	0000_0000h
1B0h	Core Time Base Halt Control Register n (TTBHLTCR0)	32	RW	0000_0000h
810h	Physical Core WFE Status Register (PCWFESR)	32	RW	0000_0000h
830h	Physical Core PW20 Status Register (PCPW20SR)	32	RW	0000_0000h
840h	Physical Core PH30 Status Register (PCPH30SR)	32	RW	0000_0000h
8A0h	Physical Core Time Base Enable Register (PCTBENR)	32	RW	0000_0000h
8A4h	Physical Core Time Base Clock Select Register (PCTBCKSELR)	32	RW	0000_0000h
1000h	Cluster PCL10 (Stop) Status Register (CLPCL10SR)	32	RW	0000_0000h
100Ch	Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)	32	RW	0000_0000h
1060h	Cluster Wakeup Event Delay Register (CLWAKEEVENTDLYR)	32	RW	0003_0003h
1100h	Cluster ACP AINACTS Set Control Register (CLAINACTSETR)	32	RW	0000_0000h
1104h	Cluster ACP AINACTS Clear Control Register (CLAINACTCLRR)	32	RW	0000_0000h
1108h	Cluster Skyros SINACT Set Control Register (CLSINACTSETR)	32	RW	0000_0000h
110Ch	Cluster Skyros SINACT Clear Control Register (CLSINACTCLRR)	32	RW	0000_0000h
1110h	Cluster L2FLUSH Set Control Register (CLL2FLUSHSETR)	32	RW	0000_0000h
1114h	Cluster L2FLUSH Clear Control Register (CLSL2FLUSHCLRR)	32	RW	0000_0000h
1118h	Cluster L2 FLUSH Done Status Register (CLL2FLUSHSR)	32	RW	0000_0000h
18A0h	Cluster Time Base Enable Register (CLTBENR)	32	RW	0000_0000h
FBF8h	IP Revision 1 Register (IPREV1R)	32	RO	0000_0000h
FBFCh	IP Revision 2 Register (IPREV2R)	32	RO	0000_0000h

20.3.3.5.1.2 Core Wait Status Register (TWAITSR0)

20.3.3.5.1.2.1 Offset

Register	Offset
TWAITSR0	40h

20.3.3.5.1.2.2 Function

This is one of a set of Core Power Management Registers. This register is used for reporting *wait status* per core.

20.3.3.5.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								T_WAITING7	T_WAITING6	T_WAITING5	T_WAITING4	T_WAITING3	T_WAITING2	T_WAITING1	T_WAITING0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.2.4 Fields

Field	Function
31-8 —	Reserved
7 T_WAITING7	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
6 T_WAITING6	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
5 T_WAITING5	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
4 T_WAITING4	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
3 T_WAITING3	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
2 T_WAITING2	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
1 T_WAITING1	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
0 T_WAITING0	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state

20.3.3.5.1.3 Core Watchdog Detect Status Register (TWDTDSR0)

20.3.3.5.1.3.1 Offset

Register	Offset
TWDTDSR0	50h

20.3.3.5.1.3.2 Function

This is one of a set of Core Power Management Registers. This register is used for reporting *detection of WDT timeout* per core as flagged on its WRS bus.

20.3.3.5.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TWDT_DET7	TWDT_DET6	TWDT_DET5	TWDT_DET4	TWDT_DET3	TWDT_DET2	TWDT_DET1	TWDT_DET0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.3.4 Fields

Field	Function
31-8 —	Reserved
7 TWDT_DET7	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU
6 TWDT_DET6	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU

Table continues on the next page...

Field	Function
5 TWDT_DET5	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU
4 TWDT_DET4	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU
3 TWDT_DET3	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU
2 TWDT_DET2	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU
1 TWDT_DET1	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU
0 TWDT_DET0	Core WDT Detect Status. 0b - core has *not* signalled a WRS condition to be serviced by PMU 1b - core *has* signalled a WRS condition to be serviced by PMU

20.3.3.5.1.4 Core Time Base Halt Control Register n (TTBHLTCR0)

20.3.3.5.1.4.1 Offset

Register	Offset
TTBHLTCR0	1B0h

20.3.3.5.1.4.2 Function

TTBHLTCR defines the time based behavior when coreis in halted state. The feature provides user more flexible timebase control in multi-coreprogramming debug.

20.3.3.5.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TTBHLT7	TTBHLT6	TTBHLT5	TTBHLT4	TTBHLT3	TTBHLT2	TTBHLT1	TTBHLT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.4.4 Fields

Field	Function
31-8 —	Reserved
7 TTBHLT7	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
6 TTBHLT6	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
5 TTBHLT5	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
4 TTBHLT4	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
3 TTBHLT3	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
2 TTBHLT2	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
1 TTBHLT1	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
0 TTBHLT0	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted

20.3.3.5.1.5 Physical Core WFE Status Register (PCWFESR)

20.3.3.5.1.5.1 Offset

Register	Offset
PCWFESR	810h

20.3.3.5.1.5.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *WFE status* per physical core.

20.3.3.5.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.5.4 Fields

Field	Function
31-8 —	Reserved
7 PC7	Physical Core WFE Status. 0b - Physical Core 7 is not in its WFE state 1b - Physical Core 7 is in its WFE state
6 PC6	Physical Core WFE Status. 0b - Physical Core 6 is not in its WFE state 1b - Physical Core 6 is in its WFE state
5 PC5	Physical Core WFE Status. 0b - Physical Core 5 is not in its WFE state 1b - Physical Core 5 is in its WFE state
4	Physical Core WFE Status.

Table continues on the next page...

Field	Function
PC4	0b - Physical Core 4 is not in its WFE state 1b - Physical Core 4 is in its WFE state
3 PC3	Physical Core WFE Status. 0b - Physical Core 3 is not in its WFE state 1b - Physical Core 3 is in its WFE state
2 PC2	Physical Core WFE Status. 0b - Physical Core 2 is not in its WFE state 1b - Physical Core 2 is in its WFE state
1 PC1	Physical Core WFE Status. 0b - Physical Core 1 is not in its WFE state 1b - Physical Core 1 is in its WFE state
0 PC0	Physical Core WFE Status. 0b - Physical Core 0 is not in its WFE state 1b - Physical Core 0 is in its WFE state

20.3.3.5.1.6 Physical Core PW20 Status Register (PCPW20SR)

20.3.3.5.1.6.1 Offset

Register	Offset
PCPW20SR	830h

20.3.3.5.1.6.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *PW20 status* per physical core.

20.3.3.5.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.6.4 Fields

Field	Function
31-8 —	Reserved
7 PC7	Physical Core PW20 Status. 0b - Physical Core 7 is not in PW20 state 1b - Physical Core 7 is in PW20 state
6 PC6	Physical Core PW20 Status. 0b - Physical Core 6 is not in PW20 state 1b - Physical Core 6 is in PW20 state
5 PC5	Physical Core PW20 Status. 0b - Physical Core 5 is not in PW20 state 1b - Physical Core 5 is in PW20 state
4 PC4	Physical Core PW20 Status. 0b - Physical Core 4 is not in PW20 state 1b - Physical Core 4 is in PW20 state
3 PC3	Physical Core PW20 Status. 0b - Physical Core 3 is not in PW20 state 1b - Physical Core 3 is in PW20 state
2 PC2	Physical Core PW20 Status. 0b - Physical Core 2 is not in PW20 state 1b - Physical Core 2 is in PW20 state
1 PC1	Physical Core PW20 Status. 0b - Physical Core 1 is not in PW20 state 1b - Physical Core 1 is in PW20 state
0 PC0	Physical Core PW20 Status. 0b - Physical Core 0 is not in PW20 state 1b - Physical Core 0 is in PW20 state

20.3.3.5.1.7 Physical Core PH30 Status Register (PCPH30SR)

20.3.3.5.1.7.1 Offset

Register	Offset
PCPH30SR	840h

20.3.3.5.1.7.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *PH30 status* per physical core.

20.3.3.5.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.7.4 Fields

Field	Function
31-8 —	Reserved
7 PC7	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
6 PC6	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
5 PC5	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
4 PC4	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
3 PC3	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
2 PC2	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
1 PC1	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
0 PC0	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state

20.3.3.5.1.8 Physical Core Time Base Enable Register (PCTBENR)

20.3.3.5.1.8.1 Offset

Register	Offset
PCTBENR	8A0h

20.3.3.5.1.8.2 Function

PCTBENR provides a mechanism for enabling clocks to any core timebases on the device.

20.3.3.5.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PCTBEN7	PCTBEN6	PCTBEN5	PCTBEN4	PCTBEN3	PCTBEN2	PCTBEN1	PCTBEN0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.8.4 Fields

Field	Function
31-8 —	Reserved
7 PCTBEN7	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
6 PCTBEN6	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
5 PCTBEN5	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled

Table continues on the next page...

Field	Function
4 PCTBEN4	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
3 PCTBEN3	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
2 PCTBEN2	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
1 PCTBEN1	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
0 PCTBEN0	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled

20.3.3.5.1.9 Physical Core Time Base Clock Select Register (PCTBCKSELR)

20.3.3.5.1.9.1 Offset

Register	Offset
PCTBCKSELR	8A4h

20.3.3.5.1.9.2 Function

PCTBCKSELR selects the clock source for each core's timebase.

20.3.3.5.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PCTBCKSEL7	PCTBCKSEL6	PCTBCKSEL5	PCTBCKSEL4	PCTBCKSEL3	PCTBCKSEL2	PCTBCKSEL1	PCTBCKSEL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.9.4 Fields

Field	Function
31-8 —	Reserved
7 PCTBCKSEL7	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
6 PCTBCKSEL6	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
5 PCTBCKSEL5	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
4 PCTBCKSEL4	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
3 PCTBCKSEL3	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
2 PCTBCKSEL2	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
1 PCTBCKSEL1	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal
0 PCTBCKSEL0	Core Timebase Input Clock Select. 0b - Core Timebase is clocked with Platform Clock/16 1b - Core Timebase is clocked with the RTC signal

20.3.3.5.1.10 Cluster PCL10 (Stop) Status Register (CLPCL10SR)

20.3.3.5.1.10.1 Offset

Register	Offset
CLPCL10SR	1000h

20.3.3.5.1.10.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *PCL10 status* per cluster.

20.3.3.5.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.10.4 Fields

Field	Function
31-2 —	Reserved
1 CL1	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state
0 CL0	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state

20.3.3.5.1.11 Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)

20.3.3.5.1.11.1 Offset

Register	Offset
CLPCL10PSR	100Ch

20.3.3.5.1.11.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for reporting previous *PCL10 status* per cluster. It is used by software to know which power saving state it was in before wake up by interrupt.

20.3.3.5.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CL_P_PCL10_1	CL_P_PCL10_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.11.4 Fields

Field	Function
31-2 —	Reserved
1 CL_P_PCL10_1	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.
0 CL_P_PCL10_0	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.

20.3.3.5.1.12 Cluster Wakeup Event Delay Register (CLWAKEEVENTDLYR)

20.3.3.5.1.12.1 Offset

Register	Offset
CLWAKEEVENTDLYR	1060h

20.3.3.5.1.12.2 Function

This register specifies the timer delay for cluster wakeup events seqencing and is provided to allow variable delays in the staggering of both CLREXMONREQ to Arm cores, as well as staggering of EVENTO signals.

20.3.3.5.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												EVENTI_DLY			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												CLREXMONREQ_DLY			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

20.3.3.5.1.12.4 Fields

Field	Function
31-20 —	Reserved
19-16 EVENTI_DLY	EVENTI delay Stagger delay value between assertions of subsequent EVENTI output signals (from PMU block), in platform cycles 0000b - no delay 0001b - 1 cycle 0010b - 4 cycles 0011b - 16 cycles (default) 0100b - 64 cycles 0101b - 256 cycles 0110b - 1024 cycles 0111b - 4096 cycles 1000b - 16384 cycles 1001b - Reserved 1111b - Reserved
15-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 CLREXMONREQ_DLY	CLREXMONREQ delay Stagger delay value between assertions of subsequent CLREXMONREQ signals, in platform cycles 0000b - no delay 0001b - 1 cycle 0010b - 4 cycles 0011b - 16 cycles (default) 0100b - 64 cycles 0101b - 256 cycles 0110b - 1024 cycles 0111b - 4096 cycles 1000b - 16384 cycles 1001b - Reserved 1111b - Reserved

20.3.3.5.1.13 Cluster ACP AINACTS Set Control Register (CLAINACTSETR)

20.3.3.5.1.13.1 Offset

Register	Offset
CLAINACTSETR	1100h

20.3.3.5.1.13.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for system to indicate the cluster ACP port is inactive after the cluster snoop domain in CCN has been closed.

20.3.3.5.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLAINACT_SET1	CLAINACT_SET0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.13.4 Fields

Field	Function
31-2 —	Reserved
1 CLAINACT_SET1	Write one to indicate the cluster ACP port is inactive after the cluster snoop domain in Cache Coherent Interconnect has been closed. 1b - Indicates cluster 1 acp port is inactive.
0 CLAINACT_SET0	Write one to indicate the cluster ACP port is inactive after the cluster snoop domain in Cache Coherent Interconnect has been closed. 1b - Indicates cluster 0 acp port is inactive.

20.3.3.5.1.14 Cluster ACP AINACTS Clear Control Register (CLAINACTCLRR)

20.3.3.5.1.14.1 Offset

Register	Offset
CLAINACTCLRR	1104h

20.3.3.5.1.14.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for system to clear the cluster AINACT control the re-enable the ACP port.

20.3.3.5.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLAINACT_CLR1	CLAINACT_CLR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.14.4 Fields

Field	Function
31-2 —	Reserved
1 CLAINACT_CLR1	Write one to clear the cluster AINACT control the re-enable the ACP port. 1b - Indicates cluster 1 acp port is re-enabled.
0 CLAINACT_CLR0	Write one to clear the cluster AINACT control the re-enable the ACP port. 1b - Indicates cluster 0 acp port is re-enabled.

20.3.3.5.1.15 Cluster Skyros SINACT Set Control Register (CLSINACTSETR)

20.3.3.5.1.15.1 Offset

Register	Offset
CLSINACTSETR	1108h

20.3.3.5.1.15.2 Function

This is one of a set of Cluster Power Management Registers. This register is used to deactivate the ACE port or the Skyros port depending on the type of ARM core cluster instantiated in the device.

20.3.3.5.1.15.3 Diagram

20.3.3.5.1.15.4 Fields

Field	Function
31-2 —	Reserved
1 CLSINACT_SET 1	Write one to indicate the cluster Skyros port is inactive after the cluster snoop domain in Cache Coherent Interconnect has been closed. 1b - Indicates cluster 1 skyros port is inactive.
0 CLSINACT_SET 0	Write one to indicate the cluster Skyros port is inactive after the cluster snoop domain in Cache Coherent Interconnect has been closed. 1b - Indicates cluster 0 skyros port is inactive.

20.3.3.5.1.16 Cluster Skyros SINACT Clear Control Register (CLSINACTCLRR)

20.3.3.5.1.16.1 Offset

Register	Offset
CLSINACTCLRR	110Ch

20.3.3.5.1.16.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for system to clear the cluster SINACT control the re-enable the Skyros port.

20.3.3.5.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLSINACT_CLR1	CLSINACT_CLR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.16.4 Fields

Field	Function
31-2 —	Reserved
1 CLSINACT_CLR1	This is one of a set of Cluster Domain Power Management Registers. This register is used for the system to clear the cluster SINACT control and re-enable the Skyros port. 1b - Indicates cluster 1 skyros port is re-enabled.
0 CLSINACT_CLR0	This is one of a set of Cluster Domain Power Management Registers. This register is used for the system to clear the cluster SINACT control and re-enable the Skyros port. 1b - Indicates cluster 0 skyros port is re-enabled.

20.3.3.5.1.17 Cluster L2FLUSH Set Control Register (CLL2FLUSHSETR)

20.3.3.5.1.17.1 Offset

Register	Offset
CLL2FLUSHSETR	1110h

20.3.3.5.1.17.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for system to request the cluster to flush its L2 cache.

20.3.3.5.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLL2FLUSH_SET1	CLL2FLUSH_SET0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.17.4 Fields

Field	Function
31-2 —	Reserved
1 CLL2FLUSH_SET1	Write one to request the cluster to flush its L2 cache. 1b - Request cluster 1 to flush its L2 cache.
0 CLL2FLUSH_SET0	Write one to request the cluster to flush its L2 cache. 1b - Request cluster 0 to flush its L2 cache.

20.3.3.5.1.18 Cluster L2FLUSH Clear Control Register (CLSL2FLUSHCLRR)

20.3.3.5.1.18.1 Offset

Register	Offset
CLSL2FLUSHCLRR	1114h

20.3.3.5.1.18.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for system to clear the cluster L2 flush request.

20.3.3.5.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLL2FLUSH_CLR1	CLL2FLUSH_CLR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.18.4 Fields

Field	Function
31-2 —	Reserved
1 CLL2FLUSH_CLR1	Write one to clear the cluster L2 flush request. 1b - Indicates cluster 1 flush request is cleared.
0 CLL2FLUSH_CLR0	Write one to clear the cluster L2 flush request. 1b - Indicates cluster 0 flush request is cleared.

20.3.3.5.1.19 Cluster L2 FLUSH Done Status Register (CLL2FLUSHSR)

20.3.3.5.1.19.1 Offset

Register	Offset
CLL2FLUSHSR	1118h

20.3.3.5.1.19.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *L2 Flush status* per cluster.

20.3.3.5.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.19.4 Fields

Field	Function
31-2 —	Reserved
1 CL1	Cluster 1 L2 flush Status. 0b - Cluster 1 is not in the L2 flush state 1b - Cluster 1 is in the L2 flush state
0 CL0	Cluster 0 L2 flush Status. 0b - Cluster 0 is not in the L2 flush state 1b - Cluster 0 is in the L2 flush state

20.3.3.5.1.20 Cluster Time Base Enable Register (CLTBENR)

20.3.3.5.1.20.1 Offset

Register	Offset
CLTBENR	18A0h

20.3.3.5.1.20.2 Function

CLTBENR provides a mechanism for enabling clocks to any core timebases on the device.

20.3.3.5.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLTBEN1	CLTBEN0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.20.4 Fields

Field	Function
31-2 —	Reserved
1 CLTBEN1	Cluster Timebase Enable. 0b - Selected Cluster Timebase is disabled 1b - Selected Cluster Timebase is enabled
0 CLTBEN0	Cluster Timebase Enable. 0b - Selected Cluster Timebase is disabled 1b - Selected Cluster Timebase is enabled

20.3.3.5.1.21 IP Revision 1 Register (IPREV1R)

20.3.3.5.1.21.1 Offset

Register	Offset
IPREV1R	FBF8h

20.3.3.5.1.21.2 Function

This register is used for software to query the IP revision number.

20.3.3.5.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IP_MJ								IP_MN							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.21.4 Fields

Field	Function
31-16	IP_ID

Table continues on the next page...

Field	Function
IP_ID	
15-8 IP_MJ	IP_MJ
7-0 IP_MN	IP_MN

20.3.3.5.1.22 IP Revision 2 Register (IPREV2R)

20.3.3.5.1.22.1 Offset

Register	Offset
IPREV2R	FBFCh

20.3.3.5.1.22.2 Function

This register is used for software to query the IP revision number.

20.3.3.5.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								IP_INT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								IP_CFG							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.3.3.5.1.22.4 Fields

Field	Function
31-24 —	Reserved
23-16 IP_INT	IP_INT
15-8	Reserved

Table continues on the next page...

Field	Function
—	
7-0 IP_CFG	IP_CFG

20.3.4 Cluster L2 Clock Gating

The cluster asserts STANBYWFIL2 to indicate it achieves L2 WFI low power state which is a prequisite power state for L2 state retention, L2 power gating and device low power modes.

The L2 WFI low power mode entry requires handshake/interlock between the cluster and the PMU as described in the following sequence:

- All CPU enter WFI
- SW disconnects the cluster from the CCN snoop domain then sets the PMU CLPMCR[AINACTS] to close the ACP slave port and CLPMCR[L2FLUSHREQ] to initiate L2 HW flush
- PMU closes cluster ACP slave port by asserting AINACTS
- PMU initiates L2 HW flush
- PMU waits for L2 HW flush done then sets PMU CLPMSR[L2FLUSHDONE]
- SW polls PMU CLPMSR[L2FLUSHDONE] then w1c to CLPMSR[L2FLUSHDONE] to clear the L2 HW flush request and status
- SW sets the CLPMCR[SINACT] to close the Skyros master port
- PMU closes cluster ACE master port by asserting SINACT
- Cluster asserts STANBYWFIL2 to indicate it achieves L2 WFI low power state

In L2 WFI, the PMU can request the CGU to gate the cluster clock via the CLKEN input.

20.4 Device Power Management

20.4.1 Central SoC Power Management Unit Features - Device and Domain

- Device Power Management:
  - LPM10 State where:
    - Not all cores are in Run (PH00) state.
  - LPM20 State where:
    - All cores are in STANDBYWF*-retain (PW20) state.

- Cluster is in STANDBYWFIL2 (PCL10) state.
- Platform clock is disabled.

20.4.2 States of Operation

20.4.2.1 Arm GPP Core Power Management State Summary- Core and Core Cluster level

The following tables summarize the core, cluster, domain and device PM modes.

NOTE

A resumable power state in the core indicates that the core can exit from a power managed state and return to Full On without the core(s) being reset.

Table 20-6. Arm GPP Core and Cluster Power Management State Summary

Power Management State Name	Resumable	Services Snoops	Device Clocks	Core Clocks	Core Voltage	State Retained	Status visible	Description
STANDBYWF* (PW15)	Y	Y	Y	N	Y	Y	TWAITSR0 or PCWFESR0	The *core* referred to is currently in the ... in the STANDBYWF* (PW15) state.
STANDBYWF*-retain (PW20)	Y	Y	Y	N	N	Y	PCPW20SR	The core referred to is currently in a power gating with state retention state.

Table 20-7. GPP Cluster Power Management State Summary

Power Management State Name	Resumable	Services Snoops	Device Clocks	Core Clocks	State Retained for Core PH20	Book E Core State Entry	Description
STANDBYWFIL2 (PCL10)	Y	N	Y	N	Y	CLPCL10SR	The cluster referred to is in a STANDBYWFIL2 state

20.4.3 LPM20 Operation Sequence

The PMU HW sequence of LPM20 operation is described below:

- Prior entering device LPM20 sleep by setting PMU POWMGTCSR[LPM20_RQ] to 1, software running on the last GPP active core, e.g. usually core #0 in cluster #0, is expected to quiesce all external devices/internal IPs through configuration. Some IPs such as WRIOP which do not provide a power management idle/stop interface would have to be idled through software as part of the software initiated power mode entry sequence.
- Software is also expected to insure memory coherency then to put the Arm cores cluster into WFIx states. This includes the following steps.
- Software initiates GPP clusters L2 HW flush by writting to CLL2FLUSHSETR register in PMU and wait for L2 flush done status by polling CLL2FLUSHSR register in PMU.
- Software removes the GPP cores clusters from the CCN snoop domains per CCN programming model.
- Software disables the GPP clusters ACP interfaces by writting to CLAINACTSETR register in PMU.
- Software disables the GPP clusters ACE/Skyros interfaces by writting to CLSINACTSETR register in PMU.
- Software running on the last core initiates LPM20 request by setting PMU register POWMGTCSR[LPM20_RQ] to 1 then exeucute WFI to put itself into STANDBYWFI.
- The PMU device power management control logic takes device LPM20 request and start LPM20 sequence.
- PMU watches the IP idle status and confirms that IPs report idle.
- PMU asserts stop request to all master IPs in device, and waits for the stop acknowledgement from all master IPs. After master IP receives stop request, it will stop mastering any new transaction. After its all transaction tenures are completed (request, response, data tenure), it will assert stop acknowledgement back to PMU.
- PMU asserts stop request to the IPs which can be served as both master and target device, and wait for the stop acknowledgement from all master/target IPs. During this process, the deadlock case could happen. If deadlock case has been detected by PMU, PMU will de-assert the stop request to these IPs and let the transaction goes through to resolve the deadlock.
- PMU wait for all cores clusters to stop and all memory to idle.
    - CPU reach the STANDBYWFI state.
    - L2 reaches the STANDBYWFIL2 state.
- PMU wait for DPAA complex to stop.

  - The DPAA complex consists mainly of the AIOP tiles and the MC complex and is separately controlled via the DPAA PMU which is a slave PM logic to the SoC PMU.
  - The device LPM20 request triggers the DPAA PMU to stop the AIOP tiles and the MC complex.
- PMU asserts stop request to all target IPs in device, and wait for the stop acknowledgement from all target IPs.
- PMU waits for CCN to reach the Q_STOPPED (QACCEPTn = 1'b0) state. Note that CNN HCG is achieved via its Q-channel interface.
- At this point, device is in sleeping state. PMU requests the clock control unit (CCU) to clock gate all IPs in device except for necessary IPs for wake up. For example, GIC and USB (if programmed). PMU disables core timebase facilities.
- Upon a wakeup event, the IPs which detect a wakup condition signal an interrupt to the GIC on its SPI (Shared Peripheral Interrupt) or PPI (Private Peripheral Interrupt) interfaces. The GIC processes the wakeup interrupts and streams the interrupt messages to the targets CPU.
- If the CPU are in low power state as indicated by GICR_WAKER.ProcessorSleep=1, the GIC generates wakeup requests to the the CPU and the PMU to exit low power modes. The GIC WakeRequest output are connected to the central SoC PMU for this purpose.

20.4.4 Memory Map and Register Definition

This section describes the memory map for the Power Management Unit block including the memory map tables.

20.4.4.1 PMU_SOC register descriptions

This section identifies Power Management resources that are not included as part of a processor core or platform IP.

This section identifies Power Management resources that are included as part of SoC power management.

20.4.4.1.1 PMU_SOC CCSR memory map

PMU_SOC base address: 1E3_0000h

Offset	Register	Width (In bits)	Access	Reset value
40h	Core Wait Status Register (TWAITSR0)	32	RO	0000_0000h
1B0h	Core Time Base Halt Control Register n (TTBHLTCR0)	32	RW	0000_0000h
824h	Physical Core PH20 Set Control Register (PCPH20SETR)	32	RW	0000_0000h
828h	Physical Core PH20 Clear Control Register (PCPH20CLRR)	32	RW	0000_0000h
840h	Physical Core PH30 Status Register (PCPH30SR)	32	RW	0000_0000h
8A0h	Physical Core Time Base Enable Register (PCTBENR)	32	RW	0000_0000h
8A8h	Time Base Clock Divider Register (TBCLKDIVR)	32	RW	0000_0000h
1000h	Cluster PCL10 (Stop) Status Register (CLPCL10SR)	32	RW	0000_0000h
100Ch	Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)	32	RW	0000_0000h
4000h	Power Management Control and Status Register (POWMGTCSR)	32	RW	0000_0000h
4040h	IP Powerdown Exception Control Register 0 (IPPDEXPCR0)	32	RW	0000_0000h
4044h	IP Powerdown Exception Control Register 1 (IPPDEXPCR1)	32	RW	0000_0000h
4048h	IP Powerdown Exception Control Register 2 (IPPDEXPCR2)	32	RW	0000_0000h
404Ch	IP Powerdown Exception Control Register 3 (IPPDEXPCR3)	32	RW	0000_0000h
4050h	IP Powerdown Exception Control Register 4 (IPPDEXPCR4)	32	RW	0000_0000h
4054h	IP Powerdown Exception Control Register 5 (IPPDEXPCR5)	32	RW	0000_0000h

20.4.4.1.2 Core Wait Status Register (TWAITSR0)

20.4.4.1.2.1 Offset

Register	Offset
TWAITSR0	40h

20.4.4.1.2.2 Function

This is one of a set of Core Power Management Registers. This register is used for reporting *wait status* per core.

20.4.4.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								T_WAITING7	T_WAITING6	T_WAITING5	T_WAITING4	T_WAITING3	T_WAITING2	T_WAITING1	T_WAITING0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.2.4 Fields

Field	Function
31-8 —	Reserved
7 T_WAITING7	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
6 T_WAITING6	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
5 T_WAITING5	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
4 T_WAITING4	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
3 T_WAITING3	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
2 T_WAITING2	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
1 T_WAITING1	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
0 T_WAITING0	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state

20.4.4.1.3 Core Time Base Halt Control Register n (TTBHLTCR0)

20.4.4.1.3.1 Offset

Register	Offset
TTBHLTCR0	1B0h

20.4.4.1.3.2 Function

TTBHLTCR defines the time based behavior when coreis in halted state. The feature provides user more flexible timebase control in multi-coreprogramming debug.

20.4.4.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TTBHLT7	TTBHLT6	TTBHLT5	TTBHLT4	TTBHLT3	TTBHLT2	TTBHLT1	TTBHLT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.3.4 Fields

Field	Function
31-8 —	Reserved
7 TTBHLT7	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
6 TTBHLT6	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
5 TTBHLT5	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted

Table continues on the next page...

Field	Function
	1b - Core Timebase is not stopped when coreis halted
4 TTBHLT4	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
3 TTBHLT3	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
2 TTBHLT2	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
1 TTBHLT1	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
0 TTBHLT0	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted

20.4.4.1.4 Physical Core PH20 Set Control Register (PCPH20SETR)

20.4.4.1.4.1 Offset

Register	Offset
PCPH20SETR	824h

20.4.4.1.4.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for requesting that a physical core be placed in the *PH20* state. The true value read from the register means there is a pending physical core PH20 request.

20.4.4.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PC_PH20_RQ7	PC_PH20_RQ6	PC_PH20_RQ5	PC_PH20_RQ4	PC_PH20_RQ3	PC_PH20_RQ2	PC_PH20_RQ1	PC_PH20_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.4.4 Fields

Field	Function
31-8 —	Reserved
7 PC_PH20_RQ7	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 7 in PH20 state
6 PC_PH20_RQ6	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 6 in PH20 state
5 PC_PH20_RQ5	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 5 in PH20 state
4 PC_PH20_RQ4	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 4 in PH20 state
3 PC_PH20_RQ3	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 3 in PH20 state
2 PC_PH20_RQ2	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 2 in PH20 state
1 PC_PH20_RQ1	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 1 in PH20 state
0 PC_PH20_RQ0	Write one to trigger physical core PH20 request. This bit is clear by interrupt event or write-one event on PCPH20CLRRn register. 1b - Request to put physical core 0 in PH20 state

20.4.4.1.5 Physical Core PH20 Clear Control Register (PCPH20CLRR)

20.4.4.1.5.1 Offset

Register	Offset
PCPH20CLRR	828h

20.4.4.1.5.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for waking up the corebe placed in the *PH20* state. The true value read from the register only means there is a pending physical core PH20 request.

20.4.4.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CP_PH20_RQ7	CP_PH20_RQ6	CP_PH20_RQ5	CP_PH20_RQ4	CP_PH20_RQ3	CP_PH20_RQ2	CP_PH20_RQ1	CP_PH20_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.5.4 Fields

Field	Function
31-8 —	Reserved
7 CP_PH20_RQ7	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 7 is receiving a PH20 request from CPPH20SETR

Table continues on the next page...

Field	Function
6 CP_PH20_RQ6	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 6 is receiving a PH20 request from CPPH20SETR
5 CP_PH20_RQ5	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 5 is receiving a PH20 request from CPPH20SETR
4 CP_PH20_RQ4	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 4 is receiving a PH20 request from CPPH20SETR
3 CP_PH20_RQ3	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 3 is receiving a PH20 request from CPPH20SETR
2 CP_PH20_RQ2	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 2 is receiving a PH20 request from CPPH20SETR
1 CP_PH20_RQ1	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 1 is receiving a PH20 request from CPPH20SETR
0 CP_PH20_RQ0	Write one to cancel physical core PH20 request. The bit is set by write-one-event to CPPH20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH20 request for that core. 1b - Physical core 0 is receiving a PH20 request from CPPH20SETR

20.4.4.1.6 Physical Core PH30 Status Register (PCPH30SR)

20.4.4.1.6.1 Offset

Register	Offset
PCPH30SR	840h

20.4.4.1.6.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *PH30 status* per physical core.

20.4.4.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.6.4 Fields

Field	Function
31-8 —	Reserved
7 PC7	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
6 PC6	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
5 PC5	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
4 PC4	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
3 PC3	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
2 PC2	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
1 PC1	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
0 PC0	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state

20.4.4.1.7 Physical Core Time Base Enable Register (PCTBENR)

20.4.4.1.7.1 Offset

Register	Offset
PCTBENR	8A0h

20.4.4.1.7.2 Function

PCTBENR provides a mechanism for enabling clocks to any core timebases on the device.

20.4.4.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PCTBEN7	PCTBEN6	PCTBEN5	PCTBEN4	PCTBEN3	PCTBEN2	PCTBEN1	PCTBEN0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.7.4 Fields

Field	Function
31-8 —	Reserved
7 PCTBEN7	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
6 PCTBEN6	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
5 PCTBEN5	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
4	Core Timebase Enable.

Table continues on the next page...

Field	Function
PCTBEN4	0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
3 PCTBEN3	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
2 PCTBEN2	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
1 PCTBEN1	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
0 PCTBEN0	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled

20.4.4.1.8 Time Base Clock Divider Register (TBCLKDIVR)

20.4.4.1.8.1 Offset

Register	Offset
TBCLKDIVR	8A8h

20.4.4.1.8.2 Function

TBCLKDIVR selects the clock source for each core's timebase.

20.4.4.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											tbclkdiv				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.8.4 Fields

Field	Function
31-5 —	Reserved
4-0 tbclkdiv	Timebase clock divider configuration 0x00 : 1/16 (default) 0x02 : 1/8 0x06 : 1/24 0x08 : 1/32

20.4.4.1.9 Cluster PCL10 (Stop) Status Register (CLPCL10SR)

20.4.4.1.9.1 Offset

Register	Offset
CLPCL10SR	1000h

20.4.4.1.9.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *PCL10 status* per cluster.

20.4.4.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.9.4 Fields

Field	Function
31-2 —	Reserved
1 CL1	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state
0 CL0	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state

20.4.4.1.10 Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)

20.4.4.1.10.1 Offset

Register	Offset
CLPCL10PSR	100Ch

20.4.4.1.10.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for reporting previous *PCL10 status* per cluster. It is used by software to know which power saving state it was in before wake up by interrupt.

20.4.4.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CL_P_PCL10_1	CL_P_PCL10_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.10.4 Fields

Field	Function
31-2 —	Reserved
1 CL_P_PCL10_1	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.
0 CL_P_PCL10_0	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.

20.4.4.1.11 Power Management Control and Status Register (POWMGTCSR)

20.4.4.1.11.1 Offset

Register	Offset
POWMGTCSR	4000h

20.4.4.1.11.2 Function

This is for performing Power Management at the device level. This register is used for entering the device's LPM state. In addition, it provides status indicating successful entry into the correspinding power management state.

20.4.4.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LYNX_PD	Reserved										LPM20_RQ	Reserved		LPM35_RQ	LPM40_RQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						LPM20_ST	P_LPM20_ST	Reserved					LPM35_ST	LPM40_ST	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.11.4 Fields

Field	Function
31 LYNX_PD	Lynx and Protocol Converter Powerdown Control 0b - Lynx and Protocol Converter are not powerdown in LPM state. 1b - Lynx and Protocol Converter are powerdown in LPM state for further power saving. If this bit is set, after system wake up from LPM state, Lynx has to go through the training sequence to return back to function.
30-21 —	Reserved
20 LPM20_RQ	LPM20 state Request. This bit is clear by interrupt event. 0b - No request to put device in LPM20 state 1b - Request to place device in LPM20 state.
19-18 —	Reserved
17 LPM35_RQ	LPM35 state Request. This bit is clear by device warm reset, hard reset, or POR reset event. 0b - No request to put device in LPM20 state 1b - Request to place device in LPM20 state.
16 LPM40_RQ	LPM40 state Request. This bit is clear by interrupt event. 0b - No request to put device in LPM40 state 1b - Request to place device in LPM40 state.
15-10 —	Reserved
9 LPM20_ST	LPM20 Status 0b - Device is not attempting to reach LPM20 state 1b - The device is attempting to enter LPM20 state because POWMGTCSR[LPM20_RQ] is set.

Table continues on the next page...

Field	Function
8 P_LPM20_ST	Previous LPM20 Status. It is used by software to know which state the device was in before being wake up. Before software set POWMGTCSR[LPM20_RQ], we expect software W1C [P_LPM20_ST] Bit. The bit is set when LPM20_ST has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Device was not in LPM20 state before being wake up by interrupt. 1b - Device was in LPM20 state before being wake up by interrupt.
7-3 —	Reserved
2 LPM35_ST	LPM35 Status 0b - Device is not attempting to reach LPM35 state 1b - The device is attempting to enter LPM35 state because POWMGTCSR[LPM35_RQ] is set.
1 LPM40_ST	LPM40 Status 0b - Device is not attempting to reach LPM40 state 1b - The device is attempting to enter LPM40 state because POWMGTCSR[LPM40_RQ] is set.
0 —	Reserved

20.4.4.1.12 IP Powerdown Exception Control Register 0 (IPPDEXPCR0)

20.4.4.1.12.1 Offset

Register	Offset
IPPDEXPCR0	4040h

20.4.4.1.12.2 Function

IPPDEXPCRn provides a mechanism for preventing clock gating certain IP during device sleep in order to make certain IP a wake-up source. For example,

- Wake on LAN (magic packet) - Requires ethernet controller preventing clock gating during LPM20
- Wake on USB (unplug/plug event) - Requires USB controller preventing clock gating during LPM20
- Wake on eSDHC (card detect event) - Requires eSDHC controller preventing clock gating during LPM20
- Wake on GPIO - Requires GPIO controller preventing clock gating during device sleep

NOTE

This functionality is optionally used on a particular device.

20.4.4.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									SEC	Reserved					SATA1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		USB2	USB1	Reserved			qDMA	Reserved					eSDHC	Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.12.4 Fields

Field	Function
31-23 —	Reserved
22 SEC	SEC
21-17 —	Reserved
16 SATA1	SATA1
15-14 —	Reserved
13 USB2	USB2
12 USB1	USB1
11-9 —	Reserved
8 qDMA	qDMA
7-3 —	Reserved
2 eSDHC	eSDHC

Table continues on the next page...

Field	Function
1-0 —	Reserved

20.4.4.1.13 IP Powerdown Exception Control Register 1 (IPPDEXPCR1)

20.4.4.1.13.1 Offset

Register	Offset
IPPDEXPCR1	4044h

20.4.4.1.13.2 Function

IPPDEXPCRn provides a mechanism for preventing clock gating certain IP during device sleep in order to make certain IP a wake-up source. For example,

- Wake on LAN (magic packet) - Requires ethernet controller preventing clock gating during LPM20
- Wake on USB (unplug/plug event) - Requires USB controller preventing clock gating during LPM20
- Wake on eSDHC (card detect event) - Requires eSDHC controller preventing clock gating during LPM20
- Wake on GPIO - Requires GPIO controller preventing clock gating during device sleep

NOTE

This functionality is optionally used on a particular device.

20.4.4.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						MAC_10	MAC_9	MAC_8	MAC_7	MAC_6	MAC_5	MAC_4	MAC_3	MAC_2	MAC_1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.13.4 Fields

Field	Function
31-10 —	Reserved
9 MAC_10	MAC_10
8 MAC_9	MAC_9
7 MAC_8	MAC_8
6 MAC_7	MAC_7
5 MAC_6	MAC_6
4 MAC_5	MAC_5
3 MAC_4	MAC_4
2 MAC_3	MAC_3
1 MAC_2	MAC_2
0 MAC_1	MAC_1

20.4.4.1.14 IP Powerdown Exception Control Register 2 (IPPDEXPCR2)

20.4.4.1.14.1 Offset

Register	Offset
IPPDEXPCR2	4048h

20.4.4.1.14.2 Function

IPPDEXPCRn provides a mechanism for preventing clock gating certain IP during device sleep in order to make certain IP a wake-up source. For example,

- Wake on LAN (magic packet) - Requires ethernet controller preventing clock gating during LPM20
- Wake on USB (unplug/plug event) - Requires USB controller preventing clock gating during LPM20
- Wake on eSDHC (card detect event) - Requires eSDHC controller preventing clock gating during LPM20
- Wake on GPIO - Requires GPIO controller preventing clock gating during device sleep

NOTE

This functionality is optionally used on a particular device.

20.4.4.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															AIOP
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		MC	QBMAN	Reserved									PCIe3	PCIe2	PCIe1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.14.4 Fields

Field	Function
31-17 —	Reserved
16 AIOP	AIOP
15-14 —	Reserved
13 MC	MC
12 QBMAN	QBMAN
11-3	Reserved

Table continues on the next page...

Field	Function
—	
2 PCIe3	PCIe3
1 PCIe2	PCIe2
0 PCIe1	PCIe1

20.4.4.1.15 IP Powerdown Exception Control Register 3 (IPPDEXPCR3)

20.4.4.1.15.1 Offset

Register	Offset
IPPDEXPCR3	404Ch

20.4.4.1.15.2 Function

IPPDEXPCRn provides a mechanism for preventing clock gating certain IP during device sleep in order to make certain IP a wake-up source. For example,

- Wake on LAN (magic packet) - Requires ethernet controller preventing clock gating during LPM20
- Wake on USB (unplug/plug event) - Requires USB controller preventing clock gating during LPM20
- Wake on eSDHC (card detect event) - Requires eSDHC controller preventing clock gating during LPM20
- Wake on GPIO - Requires GPIO controller preventing clock gating during device sleep

NOTE

This functionality is optionally used on a particular device.

20.4.4.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						I2C4	I2C3	Reserved		SPI	QSPI	DUART2	DUART1	I2C2	I2C1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.15.4 Fields

Field	Function
31-10 —	Reserved
9 I2C4	I2C4
8 I2C3	I2C3
7-6 —	Reserved
5 SPI	SPI
4 QSPI	QSPI
3 DUART2	DUART2
2 DUART1	DUART1
1 I2C2	I2C2
0 I2C1	I2C1

20.4.4.1.16 IP Powerdown Exception Control Register 4 (IPPDEXPCR4)

20.4.4.1.16.1 Offset

Register	Offset
IPPDEXPCR4	4050h

20.4.4.1.16.2 Function

IPPDEXPCRn provides a mechanism for preventing clock gating certain IP during device sleep in order to make certain IP a wake-up source. For example,

- Wake on LAN (magic packet) - Requires ethernet controller preventing clock gating during LPM20
- Wake on USB (unplug/plug event) - Requires USB controller preventing clock gating during LPM20
- Wake on eSDHC (card detect event) - Requires eSDHC controller preventing clock gating during LPM20
- Wake on GPIO - Requires GPIO controller preventing clock gating during device sleep

NOTE

This functionality is optionally used on a particular device.

20.4.4.1.16.3 Diagram

20.4.4.1.16.4 Fields

Field	Function
31-21 —	Reserved
20 PEBM	PEBM
19-15 —	Reserved
14 FlexTImers1_4	FlexTImers1-4
13 TMU	TMU
12 OCRAM	OCRAM
11 —	Reserved
10 DBG	DBG
9 GPIO	GPIO
8 IFC	IFC
7-1 —	Reserved
0 DDR1	DDR1

20.4.4.1.17 IP Powerdown Exception Control Register 5 (IPPDEXPCR5)

20.4.4.1.17.1 Offset

Register	Offset
IPPDEXPCR5	4054h

20.4.4.1.17.2 Function

IPPDEXPCRn provides a mechanism for preventing clock gating certain IP during device sleep in order to make certain IP a wake-up source. For example,

- Wake on LAN (magic packet) - Requires ethernet controller preventing clock gating during LPM20
- Wake on USB (unplug/plug event) - Requires USB controller preventing clock gating during LPM20
- Wake on eSDHC (card detect event) - Requires eSDHC controller preventing clock gating during LPM20
- Wake on GPIO - Requires GPIO controller preventing clock gating during device sleep

NOTE

This functionality is optionally used on a particular device.

20.4.4.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															WRIOP
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.4.4.1.17.4 Fields

Field	Function
31-1 —	Reserved
0 WRIOP	WRIOP

20.5 Reset Modes

20.5.1 Power-On Reset and Hard Reset State

During Power-on Reset and Hard Reset, the following hardware structures are reset:

- All CCSR registers
- Device power management state machine
- Domain power management state machine
- Cluster power management state machine
- Core power management state machine

20.5.2 Power Management States

20.5.2.1 PH10 State

In PH10 state, the core suspends instruction execution, significantly reducing the power consumption of the core. Snooping of the L1/L2 cache are still supported and thus the data in the data cache is kept coherent. Interrupts directed to the core are monitored by the device and cause PMU to use the defined handshake mechanism to exit the core from PH10 state to allow the core to recognize and process the interrupt.

The core's timer facilities are still enabled during core PH10 state, and core timebase interrupts can be generated. All device logic external to the core remains fully operational in core PH10 state.

20.5.2.2 PH15 State

In PH15 state, all clocks internal to the core are turned off except for its timer facilities clock (the core timebase). The L1/L2 caches do not respond to snoops in core PH15 state, so if coherency is required, the L1/L2 cache must be flushed before entering core PH15 state.

All device logic external to core remains fully operational in core PH15 state.

It is not possible for the device to place the core into PH15 state. This is because the device cannot initiate a cache flush of the core. Since this is not possible, if the device did place a core in PH15 state, then coherency would be lost because snooping does not occur in PH15 state.

20.5.2.3 PH20 State

Core PH20 state is core power gating with state retention.

20.5.2.4 PH30 State

Core PH30 state is core power gating without state retention. Interrupt is ignored for core being placed in PH30 state.

20.5.2.5 PCL10 State

PCL10 is a DPAA2 cluster power management state which cluster clock is gated off. PCL10 state is not recoverable with cores in PH30 but is recoverable with cores in PH20.

For cluster which is placed in PCL10 and core PH30 is a requirement for PCL10, interrupt to any core within PCL10 cluster should be ignored. Issue core warm reset to core within the cluster when cluster in PCL10 is illegal condition.

For cluster which is placed in PCL10 and core PH20 is a requirement for PCL10, interrupt to any core within PCL10 cluster will wake up cluster from PCL10 state to PCL00 state.

20.6 External Signal Description

Table 20-8. PMU Detailed Signal Descriptions (continued)

Signal	I/O	Description	
ASLEEP	O	Asleep. After negation of PORESET, ASLEEP is asserted until the device completes its power-on reset sequence and reaches its ready state.	
		State Meaning	Asserted- Indicates that the device is either still in its power-on reset sequence or it has reached a device power management state LPM20 or greater after a power-down command is issued by software. Negated- The device is not in device power management state LPM20 or greater. (It has either awake from a power-down state, or has completed the POR sequence.)
		Timing	Assertion- May occur at any time; may be asserted asynchronously to the input clocks. Negation- Negates synchronously with SYSCLK when leaving power-on sequence; otherwise negation is asynchronous.
RTC	I	Real Time Clock. Please refer to the hardware specification of device for timing specification. The signal can be optionally use to clock the global timers in the programmable interrupt controller.	
		Timing	Assertion/Negation - See the hardware specification of device for specific timing information for this signal.

20.7 Satellite DPAA Power Management Unit Features

- GPP Software controlled power management, which minimizes power consumption of blocks when they are idle.

- GPP Software controlled core PH10 mode, physical core PH15/PH20/PH30 mode, cluster PCL10 mode and device LPM10/20/30/35/40 mode.
- Enter software-controlled power mode through core or external master (e.g. PCI-Express).
- Power Core Power Management:
  - Independent Power Management Control of each core
  - PH10 State where:
    - Instruction fetching is suspended
    - All previously fetched instructions are completed
    - Core is halted but clocks remain active
  - Entry into Core Power Management states via:
    - PMU TPH10SETR for request to enter the Core 's PH10 State
    - Event Processing Unit (EPU) triggering to enter the Core 's PH10 mode
    - Device pin assertion to enter the Core 's PH10 mode
  - Independent wakeup from per core :
    - Set PMU TPH10CLRR to clear the Core PH10 request.
    - Debug interrupt
    - Core hard reset
    - Core debug halt request
- Power Core Power Management:
  - Independent Power Management Control of each Core
  - PH15 State where:
    - Entry into PH15 core power management state via PMU PCPH15SETR for request to enter the Core's PH15 state. PH15 is also triggered when GIC requests to place core in warm reset or core is disabled by COREDISR register.
    - Entered from the both core s of the core in PH10 state.
    - PH15 state wake up source is captured in Power Management States Entry and Exit Conditions
    - Core caches (for the core has PH15 request pending) must be explicitly flushed before entering core nap to maintain cache coherence. Is there a requirement?
    - locks are gated off
  - PH20 State where:
    - Entry into PH20 core power management state via PMU PCPH20SETR for request to enter the Core's PH20 state.
  - Independent wakeup from per core:
    - Set PMU TPH15/20/30CLRR to clear the Core PH15/20/30 request.
    - Debug interrupt

    - Core hard reset
    - Core debug halt request
  - PH30 State where:
    - Entry into PH30 core power management state via PMU PCPH30SETR for request to enter the Core's PH30 state.
    - Interrupt is ignored for core in PH30 state.
    - No state is retained.
    - The only wakeup method is through core warm reset.
- Power Cluster Power Management:
  - PCL10 State where:
    - Clock distribution is inhibited to cluster functional unit.
    - Prior to enter PCL10 state, all cores in cluster should be placed in PH20 mode
    - The L2 cache no longer continues to participate in snooping activities, software should always flush, then invalidate the L2 cache prior to intiating PCL10 state to ensure that any modified data is written out to backing store.
  - The entry and exit mode of cluster power management is captured in Power Management States Entry and Exit Conditions

20.7.1 Power PC Overview

PMU supports minimizing the power consumption through software controlled thread/core power management state (thread PH10, core PH15/PH20/PH30)

PMU can handshake with thread to halt the instruction sequence and handshake with Coherency Module to gracefully stop the internal Coherency network.

PMU allows several wake-up event sources to exit low power mode (internal timer, internal and external interrupts). The wake-up events are mapped to GIC interrupts to generate a wake-up interrupt to the PMU.

20.7.2 DPAA2 Core Power Management State Summary- Core, Cluster, Domain and Device Levels

The following tables summarize the core, cluster, domain and device PM modes.

NOTE

A resumable power state in the core indicates that the core can exit from a power managed state and return to Full On without the core(s) being reset.

Table 20-9. DPAA2 Core and Cluster Power Management State Summary

Power Management State Name	Resumable	Services Snoops	Device Clocks	Core Clocks	Core Voltage	State Retained	Status visible	Description
PW10	Y	Y	Y	Y	Y	All	TWAITSR0	The *core/thread* referred to is is no longer executing or fetching instructions.
PH10	Y	Y	Y	Y	Y	All	TPH10SR0	This is not applicable to Arm cores cluster which does not support PMU initiated PH10 state
PH15	Y	N	Y	N	Y	Y	TWAITSR0 or PCWFESR0	The *core* referred to is currently in the *napping* state.
PH20	Y	N	Y	N	N	Y	PCPH20SR	
PH30	N	N	Y	N	N	None		The core referred to is currently in a powered down state.

Table 20-10. Cluster PM Mode Summary for Power Management

Power Management State Name	Resumable NO'	Services Snoops	Device Clocks	Core Clocks	Core Voltage	State Retained	Status visible	Description
PCL10 (Cluster Stopped)	Y	Y	N	Y	N	Y	CLPCL10SR	The cluster referred to is in a clock gated state.
PCL20 (Cluster Retained)	Y	Y	N	Y	N	Y	CLPCL20SR	The cluster referred to is in a power gated state retained state.

Table 20-11. Domain Power Management State Summary

Domain Mode Name	Resumable NOT	Services Snoops	Device Clocks	IP Clocks	Domain Voltage	State Retained	Status Visible	Description
Power Management Modes								
DM10	Y	Y	Y	Y	Y	All	DMDPM10SR	
DM15	Y	N	Y	N	Y	All	DMPDM15SR	

20.7.3 Power Management States Entry and Exit Conditions

The following tables summarize the core, cluster, domain and device PM modes entry and exit for Power Management.

Table 20-12. Arm GPP Core PM State: Entry and Exit for Power Management

Power Management State Name	Entry via ...	Exit via ...
PH10 PM	• Set corresponding bit in PMU/TPH10SETR[T*] to 1	• Set PMU/TPH10CLRR[T*] • GIC wake requests. • Core timebase interrupt if enabled via PCTBENR. •

Table continues on the next page...

Table 20-12. Arm GPP Core PM State: Entry and Exit for Power Management (continued)

Power Management State Name	Entry via ...	Exit via ...
		• Core Hard reSetevent (The power management entry need to be cleared during reSetto avoid the power management event re-posted) • Core debug halt request • Debug interrupt without masking • Various LPM20 exits conditions as listed in Table 21-7 may cause exit of this state if TPH**SETR is not set."
PH15	• PH15 may be reached if cluster containing this core is placed into PCL10. • Core disable request by setting COREDISR[CORE] = 1	• Set corresponding bit in PMU PCPH15CLRR to 1. • GIC wake requests. • Core timebase interrupt if enable via PCTBENR. • GIC core warm reSetrequest deassertion • Core debug halt request • Debug interrupt without masking • Various LPM20 exits conditions as listed in Table 21-7 may cause exit of this state if TPH**SETR is not set." • Core disabled via COREDISR can only waken up via device hard reSetor POR reset
PH20	• PH15 may be reached if cluster containing this core is placed into PCL20.	• Set corresponding bit in PMU PCPH20CLRR to 1 • GIC wake requests. • Core timebase interrupt if enable via PCTBENR. • GIC core warm reSetrequest deassertion • Core debug halt request • Debug interrupt without masking • Various LPM20 exits conditions as listed in Table 21-7 may cause exit of this state if TPH**SETR is not set."

Table 20-13. Cluster PM Mode: Entry and Exit for Power Management

Cluster Mode Name	Entry via ...	Exit via ...
PCL10	• Set corresponding bit in PMU CLPCL10SETR register to 1.	• Set corresponding bit in PMU CLPCL10CLRR register to 1 to clear the power management request. • Core hard reset. • Device hard reset
PCL20	Set corresponding bit in PMU CLPCL20SETR register to 1	Set corresponding bit in the PMU CLPCL20CLRR to 1 to clear the power management request.

Table 20-14. Device PM Mode: Entry and Exit for Power Management

Device Mode Name	Entry via ...	Exit via ...
LPM20	• Set PMU POWMGTCSR[LPM20_REQ] to 1	• Clearing PMU POWMGTCSR[LPM20_RQ] • GIC wake requests. • Debug interrupt without masking • Wake on LAN - Magic packet, if programmed • Wake on USB - plug/unplug, if programmed • Wake on eSDHC - Card detect, if programmed • Wake on GPIO if programmed • Wake on device IRQ pins
LPM30		

20.7.4 Memory Map and Register Definition

This section describes the memory map for the Power Management Unit block including the memory map tables.

20.7.4.1 DP_PMU_PPC register descriptions

This section identifies Power Management resources that are not included as part of a processor core or platform IP.

This section identifies Power Management resources that are included as part of DPAA2 processor core power management.

20.7.4.1.1 DP_PMU_PPC CCSR memory map

DP_PMU_PPC base address: 829_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Core PH10 Status Register (TPH10SR0)	32	RO	0000_0000h
10h	Core PH10 Set Control Register (TPH10SETR0)	32	RW	0000_0000h
20h	Core PH10 Clear Control Register (TPH10CLRR)	32	RW	0000_0000h
30h	Core PH10 Previous Status Register (TPH10PSR0)	32	W1C	0000_0000h
40h	Core Wait Status Register (TWAITSR0)	32	RO	0000_0000h
1B0h	Core Time Base Halt Control Register n (TTBHLTCR0)	32	RW	0000_0000h
800h	Physical Core PH15 Status Register (PCPH15SR)	32	RO	0000_0000h
804h	Physical Core PH15 Set Control Register (PCPH15SETR)	32	RW	0000_0000h
808h	Physical Core PH15 Clear Control Register (PCPH15CLRR)	32	RW	0000_0000h
80Ch	Physical Core PH15 Previous Status Register (PCPH15PSR)	32	RW	0000_0000h
820h	Physical Core PH20 Status Register (PCPH20SR)	32	RW	0000_0000h
82Ch	Physical Core PH20 Previous Status Register (PCPH20PSR)	32	RW	0000_0000h
83Ch	Physical Core PW20 State Retention Configuration Register (PCRETCR)	32	RW	0000_0000h
840h	Physical Core PH30 Status Register (PCPH30SR)	32	RW	0000_0000h
8A0h	Physical Core Time Base Enable Register (PCTBENR)	32	RW	0000_0000h
1000h	Cluster PCL10 (Stop) Status Register (CLPCL10SR)	32	RW	0000_0000h
1004h	Cluster PCL10 (Stop) Set Control Register (CLPCL10SETR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1008h	Cluster PCL10 (Stop) Clear Control Register (CLPCL10CLRR)	32	RW	0000_0000h
100Ch	Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)	32	RW	0000_0000h
1020h	Cluster PCL20 (Retain) Status Register (CLPCL20SR)	32	RW	0000_0000h
1024h	Cluster PCL20 (Retain) Set Control Register (CLPCL20SETR)	32	RW	0000_0000h
1028h	Cluster PCL20 (Retain) Clear Control Register (CLPCL20CLRR)	32	RW	0000_0000h
102Ch	Cluster PCL20 (Retain) Previous Status Register (CLPCL20PSR)	32	RW	0000_0000h
2000h	Cluster Domain DM10 (Halt) Status Register (DMPDM10SR)	32	RW	0000_0000h
2004h	Cluster Domain DM10 (Halt) Set Control Register (DMPDM10SETR)	32	RW	0000_0000h
2008h	Cluster Domain DM10 (Halt) Clear Control Register (DMPDM10CLRR)	32	RW	0000_0000h
200Ch	Cluster Domain DM10 (Halt) Previous Status Register (DMPDM10PSR)	32	RW	0000_0000h
2020h	Cluster Domain DM15 (Stop) Status Register (DMPDM15SR)	32	RW	0000_0000h
2024h	Cluster Domain DM15 (Stop) Set Control Register (DMPDM15SETR)	32	RW	0000_0000h
2028h	Cluster Domain DM15 (Stop) Clear Control Register (DMPDM15CLRR)	32	RW	0000_0000h
202Ch	Cluster Domain DM15 (Stop) Previous Status Register (DMPDM15PSR)	32	RW	0000_0000h
2040h	Cluster Domain DM20 (Retain) Status Register (DMPDM20SR)	32	RW	0000_0000h
2044h	Cluster Domain DM20 (Retain) Set Control Register (DMPDM20SETR)	32	RW	0000_0000h
2048h	Cluster Domain DM20 (Retain) Clear Control Register (DMPDM20CLRR)	32	RW	0000_0000h
204Ch	Cluster Domain DM20 (Retain) Previous Status Register (DMPDM20PSR)	32	RW	0000_0000h
C000h	Cluster Domain Warm Reset Control Register (RSTCR)	32	RW	0000_0000h
C020h	Cluster Domain Warm Reset Disable Control Register (RSTEXPCR)	32	RW	0000_0000h

20.7.4.1.2 Core PH10 Status Register (TPH10SR0)

20.7.4.1.2.1 Offset

Register	Offset
TPH10SR0	0h

20.7.4.1.2.2 Function

This is one of a set of corepower management registers. This register is used for reporting *PH10 status* per core.

20.7.4.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														T17	T16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.2.4 Fields

Field	Function
31-18 —	Reserved
17 T17	Core 17 PH10 Status. 0b - core 17 is not in the PH10 state 1b - core 17 is in the PH10 state
16 T16	Core 16 PH10 Status. 0b - core 16 is not in the PH10 state 1b - core 16 is in the PH10 state
15 T15	Core 15 PH10 Status. 0b - core 15 is not in the PH10 state 1b - core 15 is in the PH10 state
14 T14	Core 14 PH10 Status. 0b - core 14 is not in the PH10 state 1b - core 14 is in the PH10 state
13 T13	Core 13 PH10 Status. 0b - core 13 is not in the PH10 state 1b - core 13 is in the PH10 state
12 T12	Core 12 PH10 Status. 0b - core 12 is not in the PH10 state 1b - core 12 is in the PH10 state
11 T11	Core 11 PH10 Status. 0b - core 11 is not in the PH10 state 1b - core 11 is in the PH10 state
10 T10	Core 10 PH10 Status. 0b - core 10 is not in the PH10 state 1b - core 10 is in the PH10 state
9	Core 9 PH10 Status.

Table continues on the next page...

Field	Function
T9	0b - core 9 is not in the PH10 state 1b - core 9 is in the PH10 state
8 T8	Core 8 PH10 Status. 0b - core 8 is not in the PH10 state 1b - core 8 is in the PH10 state
7 T7	Core 7 PH10 Status. 0b - core 7 is not in the PH10 state 1b - core 7 is in the PH10 state
6 T6	Core 6 PH10 Status. 0b - core 6 is not in the PH10 state 1b - core 6 is in the PH10 state
5 T5	Core 5 PH10 Status. 0b - core 5 is not in the PH10 state 1b - core 5 is in the PH10 state
4 T4	Core 4 PH10 Status. 0b - core 4 is not in the PH10 state 1b - core 4 is in the PH10 state
3 T3	Core 3 PH10 Status. 0b - core 3 is not in the PH10 state 1b - core 3 is in the PH10 state
2 T2	Core 2 PH10 Status. 0b - core 2 is not in the PH10 state 1b - core 2 is in the PH10 state
1 T1	Core 1 PH10 Status. 0b - core 1 is not in the PH10 state 1b - core 1 is in the PH10 state
0 T0	Core 0 PH10 Status. 0b - core 0 is not in the PH10 state 1b - core 0 is in the PH10 state

20.7.4.1.3 Core PH10 Set Control Register (TPH10SETR0)

20.7.4.1.3.1 Offset

Register	Offset
TPH10SETR0	10h

20.7.4.1.3.2 Function

This is one of a set of Core Power Management Registers. This register is used for requesting that a corebe placed in the *PH10* state. The true value read from the register means there is a pending corePH10 request.

20.7.4.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														T_PH10_REQ17	T_PH10_REQ16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	T_PH10_REQ15	T_PH10_REQ14	T_PH10_REQ13	T_PH10_REQ12	T_PH10_REQ11	T_PH10_REQ10	T_PH10_REQ9	T_PH10_REQ8	T_PH10_REQ7	T_PH10_REQ6	T_PH10_REQ5	T_PH10_REQ4	T_PH10_REQ3	T_PH10_REQ2	T_PH10_REQ1	T_PH10_REQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.3.4 Fields

Field	Function
31-18 —	Reserved
17 T_PH10_REQ17	Write one to trigger core 17 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 17 in PH10 state. 1b - Request to put core 17 in PH10 state.
16 T_PH10_REQ16	Write one to trigger core 16 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 16 in PH10 state. 1b - Request to put core 16 in PH10 state.
15 T_PH10_REQ15	Write one to trigger core 15 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 15 in PH10 state. 1b - Request to put core 15 in PH10 state.
14	Write one to trigger core 14 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register.

Table continues on the next page...

Field	Function
T_PH10_REQ14	0b - No request to put core 14 in PH10 state. 1b - Request to put core 14 in PH10 state.
13 T_PH10_REQ13	Write one to trigger core 13 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 13 in PH10 state. 1b - Request to put core 13 in PH10 state.
12 T_PH10_REQ12	Write one to trigger core 12 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 12 in PH10 state. 1b - Request to put core 12 in PH10 state.
11 T_PH10_REQ11	Write one to trigger core 11 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 11 in PH10 state. 1b - Request to put core 11 in PH10 state.
10 T_PH10_REQ10	Write one to trigger core 10 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 10 in PH10 state. 1b - Request to put core 10 in PH10 state.
9 T_PH10_REQ9	Write one to trigger core 9 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 9 in PH10 state. 1b - Request to put core 9 in PH10 state.
8 T_PH10_REQ8	Write one to trigger core 8 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 8 in PH10 state. 1b - Request to put core 8 in PH10 state.
7 T_PH10_REQ7	Write one to trigger core 7 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 7 in PH10 state. 1b - Request to put core 7 in PH10 state.
6 T_PH10_REQ6	Write one to trigger core 6 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 6 in PH10 state. 1b - Request to put core 6 in PH10 state.
5 T_PH10_REQ5	Write one to trigger core 5 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 5 in PH10 state. 1b - Request to put core 5 in PH10 state.
4 T_PH10_REQ4	Write one to trigger core 4 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 4 in PH10 state. 1b - Request to put core 4 in PH10 state.
3 T_PH10_REQ3	Write one to trigger core 3 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 3 in PH10 state. 1b - Request to put core 3 in PH10 state.

Table continues on the next page...

Field	Function
2 T_PH10_REQ2	Write one to trigger core 2 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 2 in PH10 state. 1b - Request to put core 2 in PH10 state.
1 T_PH10_REQ1	Write one to trigger core 1 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 1 in PH10 state. 1b - Request to put core 1 in PH10 state.
0 T_PH10_REQ0	Write one to trigger core 0 PH10 request. This bit is cleared by interrupt event or write-one event on TPH10CLRRn register. 0b - No request to put core 0 in PH10 state. 1b - Request to put core 0 in PH10 state.

20.7.4.1.4 Core PH10 Clear Control Register (TPH10CLRR)

20.7.4.1.4.1 Offset

Register	Offset
TPH10CLRR	20h

20.7.4.1.4.2 Function

This is one of a set of Core Power Management Registers. This register is used for waking up the corebe placed in the *PH10* state. The true value read from the register only means there is a pending corePH10 request.

20.7.4.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														T_PH10_REQ17	T_PH10_REQ16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	T_PH10_REQ15	T_PH10_REQ14	T_PH10_REQ13	T_PH10_REQ12	T_PH10_REQ11	T_PH10_REQ10	T_PH10_REQ9	T_PH10_REQ8	T_PH10_REQ7	T_PH10_REQ6	T_PH10_REQ5	T_PH10_REQ4	T_PH10_REQ3	T_PH10_REQ2	T_PH10_REQ1	T_PH10_REQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.4.4 Fields

Field	Function
31-18 —	Reserved
17 T_PH10_REQ17	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 17 in PH10 state. 1b - Request to put core 17 in PH10 state.
16 T_PH10_REQ16	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 16 in PH10 state. 1b - Request to put core 16 in PH10 state.
15 T_PH10_REQ15	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 15 in PH10 state. 1b - Request to put core 15 in PH10 state.
14 T_PH10_REQ14	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 14 in PH10 state. 1b - Request to put core 14 in PH10 state.

Table continues on the next page...

Field	Function
13 T_PH10_REQ13	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 13 in PH10 state. 1b - Request to put core 13 in PH10 state.
12 T_PH10_REQ12	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 12 in PH10 state. 1b - Request to put core 12 in PH10 state.
11 T_PH10_REQ11	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 11 in PH10 state. 1b - Request to put core 11 in PH10 state.
10 T_PH10_REQ10	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 10 in PH10 state. 1b - Request to put core 10 in PH10 state.
9 T_PH10_REQ9	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 9 in PH10 state. 1b - Request to put core 9 in PH10 state.
8 T_PH10_REQ8	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 8 in PH10 state. 1b - Request to put core 8 in PH10 state.
7 T_PH10_REQ7	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 7 in PH10 state. 1b - Request to put core 7 in PH10 state.
6 T_PH10_REQ6	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 6 in PH10 state. 1b - Request to put core 6 in PH10 state.
5 T_PH10_REQ5	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 5 in PH10 state. 1b - Request to put core 5 in PH10 state.
4 T_PH10_REQ4	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core .

Table continues on the next page...

Field	Function
	0b - No request to put core 4 in PH10 state. 1b - Request to put core 4 in PH10 state.
3 T_PH10_REQ3	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 3 in PH10 state. 1b - Request to put core 3 in PH10 state.
2 T_PH10_REQ2	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 2 in PH10 state. 1b - Request to put core 2 in PH10 state.
1 T_PH10_REQ1	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 1 in PH10 state. 1b - Request to put core 1 in PH10 state.
0 T_PH10_REQ0	Write one to cancel core PH10 request. The bit is set by write-one-event to TPH10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending corePH10 request for that core . 0b - No request to put core 0 in PH10 state. 1b - Request to put core 0 in PH10 state.

20.7.4.1.5 Core PH10 Previous Status Register (TPH10PSR0)

20.7.4.1.5.1 Offset

Register	Offset
TPH10PSR0	30h

20.7.4.1.5.2 Function

This is one of a set of Core Power Management Registers. This register is used for reporting previous *PH10 status* per core. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														T_P_PH10_17	T_P_PH10_16
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	T_P_PH10_15	T_P_PH10_14	T_P_PH10_13	T_P_PH10_12	T_P_PH10_11	T_P_PH10_10	T_P_PH10_9	T_P_PH10_8	T_P_PH10_7	T_P_PH10_6	T_P_PH10_5	T_P_PH10_4	T_P_PH10_3	T_P_PH10_2	T_P_PH10_1	T_P_PH10_0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.5.4 Fields

Field	Function
31-18 —	Reserved
17 T_P_PH10_17	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
16 T_P_PH10_16	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
15 T_P_PH10_15	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
14 T_P_PH10_14	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
13 T_P_PH10_13	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c.

Table continues on the next page...

Field	Function
	0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
12 T_P_PH10_12	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
11 T_P_PH10_11	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
10 T_P_PH10_10	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
9 T_P_PH10_9	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
8 T_P_PH10_8	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
7 T_P_PH10_7	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
6 T_P_PH10_6	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
5 T_P_PH10_5	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
4 T_P_PH10_4	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
3 T_P_PH10_3	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
2 T_P_PH10_2	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.

Table continues on the next page...

Field	Function
1 T_P_PH10_1	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.
0 T_P_PH10_0	Core PH10 previous status. The bit is set when corresponding TPH10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - core was not in the PH10 state before wake up by interrupt. 1b - core was in the PH10 state before wake up by interrupt.

20.7.4.1.6 Core Wait Status Register (TWAITSR0)

20.7.4.1.6.1 Offset

Register	Offset
TWAITSR0	40h

20.7.4.1.6.2 Function

This is one of a set of Core Power Management Registers. This register is used for reporting *wait status* per core.

20.7.4.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														T_WAITING17	T_WAITING16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	T_WAITING15	T_WAITING14	T_WAITING13	T_WAITING12	T_WAITING11	T_WAITING10	T_WAITING9	T_WAITING8	T_WAITING7	T_WAITING6	T_WAITING5	T_WAITING4	T_WAITING3	T_WAITING2	T_WAITING1	T_WAITING0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.6.4 Fields

Field	Function
31-18 —	Reserved
17 T_WAITING17	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
16 T_WAITING16	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
15 T_WAITING15	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
14 T_WAITING14	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
13 T_WAITING13	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
12 T_WAITING12	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
11 T_WAITING11	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
10 T_WAITING10	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
9 T_WAITING9	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
8 T_WAITING8	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
7 T_WAITING7	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
6 T_WAITING6	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
5 T_WAITING5	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state

Table continues on the next page...

Field	Function
4 T_WAITING4	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
3 T_WAITING3	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
2 T_WAITING2	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
1 T_WAITING1	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state
0 T_WAITING0	Core Wait Status. 0b - Core is not in its Wait state 1b - Core is in its Wait state

20.7.4.1.7 Core Time Base Halt Control Register n (TTBHLTCR0)

20.7.4.1.7.1 Offset

Register	Offset
TTBHLTCR0	1B0h

20.7.4.1.7.2 Function

TTBHLTCR defines the time based behavior when coreis in halted state. The feature provides user more flexible timebase control in multi-coreprogramming debug.

20.7.4.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														TTBHLT17	TTBHLT16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TTBHLT15	TTBHLT14	TTBHLT13	TTBHLT12	TTBHLT11	TTBHLT10	TTBHLT9	TTBHLT8	TTBHLT7	TTBHLT6	TTBHLT5	TTBHLT4	TTBHLT3	TTBHLT2	TTBHLT1	TTBHLT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.7.4 Fields

Field	Function
31-18 —	Reserved
17 TTBHLT17	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
16 TTBHLT16	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
15 TTBHLT15	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
14 TTBHLT14	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
13 TTBHLT13	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
12 TTBHLT12	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
11 TTBHLT11	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
10 TTBHLT10	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted

Table continues on the next page...

Field	Function
	1b - Core Timebase is not stopped when coreis halted
9 TTBHLT9	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
8 TTBHLT8	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
7 TTBHLT7	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
6 TTBHLT6	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
5 TTBHLT5	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
4 TTBHLT4	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
3 TTBHLT3	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
2 TTBHLT2	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
1 TTBHLT1	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted
0 TTBHLT0	Core Timebase Behavior during core halted 0b - Core Timebase is stopped when coreis halted 1b - Core Timebase is not stopped when coreis halted

20.7.4.1.8 Physical Core PH15 Status Register (PCPH15SR)

20.7.4.1.8.1 Offset

Register	Offset
PCPH15SR	800h

20.7.4.1.8.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *PH15 status* per physical core.

20.7.4.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC17	PC16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.8.4 Fields

Field	Function
31-18 —	Reserved
17 PC17	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
16 PC16	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
15 PC15	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
14 PC14	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
13 PC13	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
12 PC12	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
11	Physical Core PH15 Status.

Table continues on the next page...

Field	Function
PC11	0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
10 PC10	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
9 PC9	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
8 PC8	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
7 PC7	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
6 PC6	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
5 PC5	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
4 PC4	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
3 PC3	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
2 PC2	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
1 PC1	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state
0 PC0	Physical Core PH15 Status. 0b - Physical Core is not in its PH15 state 1b - Physical Core is in its PH15 state

20.7.4.1.9 Physical Core PH15 Set Control Register (PCPH15SETR)

20.7.4.1.9.1 Offset

Register	Offset
PCPH15SETR	804h

20.7.4.1.9.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for requesting that a physical core be placed in the *PH15* state. The true value read from the register means there is a pending physical core PH15 request.

20.7.4.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC_PH15_RQ17	PC_PH15_RQ16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC_PH15_RQ15	PC_PH15_RQ14	PC_PH15_RQ13	PC_PH15_RQ12	PC_PH15_RQ11	PC_PH15_RQ10	PC_PH15_RQ9	PC_PH15_RQ8	PC_PH15_RQ7	PC_PH15_RQ6	PC_PH15_RQ5	PC_PH15_RQ4	PC_PH15_RQ3	PC_PH15_RQ2	PC_PH15_RQ1	PC_PH15_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.9.4 Fields

Field	Function
31-18 —	Reserved
17 PC_PH15_RQ17	Write one to trigger core 17 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 17 in PH15 state. 1b - Request to put core 17 in PH15 state.
16 PC_PH15_RQ16	Write one to trigger core 16 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 16 in PH15 state. 1b - Request to put core 16 in PH15 state.
15 PC_PH15_RQ15	Write one to trigger core 15 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 15 in PH15 state. 1b - Request to put core 15 in PH15 state.

Table continues on the next page...

Field	Function
14 PC_PH15_RQ14	Write one to trigger core 14 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 14 in PH15 state. 1b - Request to put core 14 in PH15 state.
13 PC_PH15_RQ13	Write one to trigger core 13 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 13 in PH15 state. 1b - Request to put core 13 in PH15 state.
12 PC_PH15_RQ12	Write one to trigger core 12 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 12 in PH15 state. 1b - Request to put core 12 in PH15 state.
11 PC_PH15_RQ11	Write one to trigger core 11 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 11 in PH15 state. 1b - Request to put core 11 in PH15 state.
10 PC_PH15_RQ10	Write one to trigger core 10 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 10 in PH15 state. 1b - Request to put core 10 in PH15 state.
9 PC_PH15_RQ9	Write one to trigger core 9 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 9 in PH15 state. 1b - Request to put core 9 in PH15 state.
8 PC_PH15_RQ8	Write one to trigger core 8 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 8 in PH15 state. 1b - Request to put core 8 in PH15 state.
7 PC_PH15_RQ7	Write one to trigger core 7 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 7 in PH15 state. 1b - Request to put core 7 in PH15 state.
6 PC_PH15_RQ6	Write one to trigger core 6 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 6 in PH15 state. 1b - Request to put core 6 in PH15 state.
5 PC_PH15_RQ5	Write one to trigger core 5 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 5 in PH15 state. 1b - Request to put core 5 in PH15 state.
4 PC_PH15_RQ4	Write one to trigger core 4 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 4 in PH15 state. 1b - Request to put core 4 in PH15 state.
3 PC_PH15_RQ3	Write one to trigger core 3 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 3 in PH15 state.

Table continues on the next page...

Field	Function
	1b - Request to put core 3 in PH15 state.
2 PC_PH15_RQ2	Write one to trigger core 2 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 2 in PH15 state. 1b - Request to put core 2 in PH15 state.
1 PC_PH15_RQ1	Write one to trigger core 1 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 1 in PH15 state. 1b - Request to put core 1 in PH15 state.
0 PC_PH15_RQ0	Write one to trigger core 0 PH15 request. This bit is cleared by interrupt event or write-one event on PCPH15CLRRn register. 0b - No request to put core 0 in PH15 state. 1b - Request to put core 0 in PH15 state.

20.7.4.1.10 Physical Core PH15 Clear Control Register (PCPH15CLRR)

20.7.4.1.10.1 Offset

Register	Offset
PCPH15CLRR	808h

20.7.4.1.10.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for waking up the corebe placed in the *PH15* state. The true value read from the register only means there is a pending physical core PH15 request.

20.7.4.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														CP_PH15_RQ17	CP_PH15_RQ16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CP_PH15_RQ15	CP_PH15_RQ14	CP_PH15_RQ13	CP_PH15_RQ12	CP_PH15_RQ11	CP_PH15_RQ10	CP_PH15_RQ9	CP_PH15_RQ8	CP_PH15_RQ7	CP_PH15_RQ6	CP_PH15_RQ5	CP_PH15_RQ4	CP_PH15_RQ3	CP_PH15_RQ2	CP_PH15_RQ1	CP_PH15_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.10.4 Fields

Field	Function
31-18 —	Reserved
17 CP_PH15_RQ17	Write one to cancel core 17 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 17 is not receiving a PH15 request from CPPH15SETR. 1b - core 17 is receiving a PH15 request from CPPH15SETR.
16 CP_PH15_RQ16	Write one to cancel core 16 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 16 is not receiving a PH15 request from CPPH15SETR. 1b - core 16 is receiving a PH15 request from CPPH15SETR.
15 CP_PH15_RQ15	Write one to cancel core 15 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 15 is not receiving a PH15 request from CPPH15SETR. 1b - core 15 is receiving a PH15 request from CPPH15SETR.
14 CP_PH15_RQ14	Write one to cancel core 14 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 14 is not receiving a PH15 request from CPPH15SETR. 1b - core 14 is receiving a PH15 request from CPPH15SETR.

Table continues on the next page...

Field	Function
13 CP_PH15_RQ13	Write one to cancel core 13 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 13 is not receiving a PH15 request from CPPH15SETR. 1b - core 13 is receiving a PH15 request from CPPH15SETR.
12 CP_PH15_RQ12	Write one to cancel core 12 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 12 is not receiving a PH15 request from CPPH15SETR. 1b - core 12 is receiving a PH15 request from CPPH15SETR.
11 CP_PH15_RQ11	Write one to cancel core 11 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 11 is not receiving a PH15 request from CPPH15SETR. 1b - core 11 is receiving a PH15 request from CPPH15SETR.
10 CP_PH15_RQ10	Write one to cancel core 10 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 10 is not receiving a PH15 request from CPPH15SETR. 1b - core 10 is receiving a PH15 request from CPPH15SETR.
9 CP_PH15_RQ9	Write one to cancel core 9 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 9 is not receiving a PH15 request from CPPH15SETR. 1b - core 9 is receiving a PH15 request from CPPH15SETR.
8 CP_PH15_RQ8	Write one to cancel core 8 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 8 is not receiving a PH15 request from CPPH15SETR. 1b - core 8 is receiving a PH15 request from CPPH15SETR.
7 CP_PH15_RQ7	Write one to cancel core 7 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 7 is not receiving a PH15 request from CPPH15SETR. 1b - core 7 is receiving a PH15 request from CPPH15SETR.
6 CP_PH15_RQ6	Write one to cancel core 6 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 6 is not receiving a PH15 request from CPPH15SETR. 1b - core 6 is receiving a PH15 request from CPPH15SETR.
5 CP_PH15_RQ5	Write one to cancel core 5 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 5 is not receiving a PH15 request from CPPH15SETR. 1b - core 5 is receiving a PH15 request from CPPH15SETR.
4 CP_PH15_RQ4	Write one to cancel core 4 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core.

Table continues on the next page...

Field	Function
	0b - core 4 is not receiving a PH15 request from CPPH15SETR. 1b - core 4 is receiving a PH15 request from CPPH15SETR.
3 CP_PH15_RQ3	Write one to cancel core 3 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 3 is not receiving a PH15 request from CPPH15SETR. 1b - core 3 is receiving a PH15 request from CPPH15SETR.
2 CP_PH15_RQ2	Write one to cancel core 2 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 2 is not receiving a PH15 request from CPPH15SETR. 1b - core 2 is receiving a PH15 request from CPPH15SETR.
1 CP_PH15_RQ1	Write one to cancel core 1 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 1 is not receiving a PH15 request from CPPH15SETR. 1b - core 1 is receiving a PH15 request from CPPH15SETR.
0 CP_PH15_RQ0	Write one to cancel core 0 PH15 request. The bit is set by write-one-event to CPPH15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending Physical Core PH15 request for that core. 0b - core 0 is not receiving a PH15 request from CPPH15SETR. 1b - core 0 is receiving a PH15 request from CPPH15SETR.

20.7.4.1.11 Physical Core PH15 Previous Status Register (PCPH15PSR)

20.7.4.1.11.1 Offset

Register	Offset
PCPH15PSR	80Ch

20.7.4.1.11.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting previous *PH15 status* per physical core. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC_P_PH15_17	PC_P_PH15_16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC_P_PH15_15	PC_P_PH15_14	PC_P_PH15_13	PC_P_PH15_12	PC_P_PH15_11	PC_P_PH15_10	PC_P_PH15_9	PC_P_PH15_8	PC_P_PH15_7	PC_P_PH15_6	PC_P_PH15_5	PC_P_PH15_4	PC_P_PH15_3	PC_P_PH15_2	PC_P_PH15_1	PC_P_PH15_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.11.4 Fields

Field	Function
31-18 —	Reserved
17 PC_P_PH15_17	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
16 PC_P_PH15_16	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
15 PC_P_PH15_15	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
14 PC_P_PH15_14	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
13 PC_P_PH15_13	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.

Table continues on the next page...

Field	Function
12 PC_P_PH15_12	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
11 PC_P_PH15_11	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
10 PC_P_PH15_10	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
9 PC_P_PH15_9	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
8 PC_P_PH15_8	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
7 PC_P_PH15_7	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
6 PC_P_PH15_6	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
5 PC_P_PH15_5	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
4 PC_P_PH15_4	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
3 PC_P_PH15_3	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
2 PC_P_PH15_2	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.
1 PC_P_PH15_1	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt.

Table continues on the next page...

Field	Function
	1b - Physical core was in the PH15 state before wake up by interrupt.
0 PC_P_PH15_0	Physical core PH15 previous status. The bit is set when corresponding PCPH15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH15 state before wake up by interrupt. 1b - Physical core was in the PH15 state before wake up by interrupt.

20.7.4.1.12 Physical Core PH20 Status Register (PCPH20SR)

20.7.4.1.12.1 Offset

Register	Offset
PCPH20SR	820h

20.7.4.1.12.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *PH20 status* per physical core.

20.7.4.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC17	PC16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.12.4 Fields

Field	Function
31-18 —	Reserved
17 PC17	Physical Core PH20 Status. 0b - Physical core is not in PH20 state

Table continues on the next page...

Field	Function
	1b - Physical core is in PH20 state
16 PC16	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
15 PC15	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
14 PC14	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
13 PC13	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
12 PC12	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
11 PC11	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
10 PC10	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
9 PC9	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
8 PC8	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
7 PC7	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
6 PC6	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
5 PC5	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
4 PC4	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
3 PC3	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
2	Physical Core PH20 Status.

Table continues on the next page...

Field	Function
PC2	0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
1 PC1	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state
0 PC0	Physical Core PH20 Status. 0b - Physical core is not in PH20 state 1b - Physical core is in PH20 state

20.7.4.1.13 Physical Core PH20 Previous Status Register (PCPH20PSR)

20.7.4.1.13.1 Offset

Register	Offset
PCPH20PSR	82Ch

20.7.4.1.13.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting previous *PH20 status* per physical core. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC_P_PH20_17	PC_P_PH20_16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC_P_PH20_15	PC_P_PH20_14	PC_P_PH20_13	PC_P_PH20_12	PC_P_PH20_11	PC_P_PH20_10	PC_P_PH20_9	PC_P_PH20_8	PC_P_PH20_7	PC_P_PH20_6	PC_P_PH20_5	PC_P_PH20_4	PC_P_PH20_3	PC_P_PH20_2	PC_P_PH20_1	PC_P_PH20_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.13.4 Fields

Field	Function
31-18 —	Reserved
17 PC_P_PH20_17	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
16 PC_P_PH20_16	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
15 PC_P_PH20_15	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
14 PC_P_PH20_14	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
13 PC_P_PH20_13	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
12 PC_P_PH20_12	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
11 PC_P_PH20_11	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
10 PC_P_PH20_10	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
9 PC_P_PH20_9	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
8 PC_P_PH20_8	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.

Table continues on the next page...

Field	Function
7 PC_P_PH20_7	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
6 PC_P_PH20_6	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
5 PC_P_PH20_5	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
4 PC_P_PH20_4	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
3 PC_P_PH20_3	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
2 PC_P_PH20_2	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
1 PC_P_PH20_1	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.
0 PC_P_PH20_0	Physical core PH20 previous status. The bit is set when corresponding PCPH20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Physical core was not in the PH20 state before wake up by interrupt. 1b - Physical core was in the PH20 state before wake up by interrupt.

20.7.4.1.14 Physical Core PW20 State Retention Configuration Register (PCRETCR)

20.7.4.1.14.1 Offset

Register	Offset
PCRETCR	83Ch

20.7.4.1.14.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for enabling PW20 state retention feature of a physical core.

20.7.4.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC_RET_EN17	PC_RET_EN16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC_RET_EN15	PC_RET_EN14	PC_RET_EN13	PC_RET_EN12	PC_RET_EN11	PC_RET_EN10	PC_RET_EN 9	PC_RET_EN 8	PC_RET_EN 7	PC_RET_EN 6	PC_RET_EN 5	PC_RET_EN 4	PC_RET_EN 3	PC_RET_EN 2	PC_RET_EN 1	PC_RET_EN 0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.14.4 Fields

Field	Function
31-18 —	Reserved
17 PC_RET_EN17	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
16 PC_RET_EN16	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
15 PC_RET_EN15	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
14 PC_RET_EN14	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
13 PC_RET_EN13	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled

Table continues on the next page...

Field	Function
12 PC_RET_EN12	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
11 PC_RET_EN11	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
10 PC_RET_EN10	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
9 PC_RET_EN9	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
8 PC_RET_EN8	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
7 PC_RET_EN7	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
6 PC_RET_EN6	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
5 PC_RET_EN5	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
4 PC_RET_EN4	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
3 PC_RET_EN3	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
2 PC_RET_EN2	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
1 PC_RET_EN1	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled
0 PC_RET_EN0	Write one to enable physical core state retention. 0b - Physical core PW20 state is not enabled 1b - Physical core PW20 state is enabled

20.7.4.1.15 Physical Core PH30 Status Register (PCPH30SR)

20.7.4.1.15.1 Offset

Register	Offset
PCPH30SR	840h

20.7.4.1.15.2 Function

This is one of a set of Physical Core Power Management Registers. This register is used for reporting *PH30 status* per physical core.

20.7.4.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PC17	PC16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.15.4 Fields

Field	Function
31-18 —	Reserved
17 PC17	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
16 PC16	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
15 PC15	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
14 PC14	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
13	Physical Core PH30 Status.

Table continues on the next page...

Field	Function
PC13	0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
12 PC12	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
11 PC11	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
10 PC10	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
9 PC9	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
8 PC8	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
7 PC7	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
6 PC6	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
5 PC5	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
4 PC4	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
3 PC3	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
2 PC2	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
1 PC1	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state
0 PC0	Physical Core PH30 Status. 0b - Physical core is not in PH30 state 1b - Physical core is in PH30 state

20.7.4.1.16 Physical Core Time Base Enable Register (PCTBENR)

20.7.4.1.16.1 Offset

Register	Offset
PCTBENR	8A0h

20.7.4.1.16.2 Function

PCTBENR provides a mechanism for enabling clocks to any core timebases on the device.

20.7.4.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														PCTBEN17	PCTBEN16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCTBEN15	PCTBEN14	PCTBEN13	PCTBEN12	PCTBEN11	PCTBEN10	PCTBEN9	PCTBEN8	PCTBEN7	PCTBEN6	PCTBEN5	PCTBEN4	PCTBEN3	PCTBEN2	PCTBEN1	PCTBEN0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.16.4 Fields

Field	Function
31-18 —	Reserved
17 PCTBEN17	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
16 PCTBEN16	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
15	Core Timebase Enable.

Table continues on the next page...

Field	Function
PCTBEN15	0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
14 PCTBEN14	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
13 PCTBEN13	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
12 PCTBEN12	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
11 PCTBEN11	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
10 PCTBEN10	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
9 PCTBEN9	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
8 PCTBEN8	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
7 PCTBEN7	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
6 PCTBEN6	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
5 PCTBEN5	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
4 PCTBEN4	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
3 PCTBEN3	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
2 PCTBEN2	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled
1 PCTBEN1	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled

Table continues on the next page...

Field	Function
0 PCTBEN0	Core Timebase Enable. 0b - Selected Core Timebase is disabled 1b - Selected Core Timebase is enabled

20.7.4.1.17 Cluster PCL10 (Stop) Status Register (CLPCL10SR)

20.7.4.1.17.1 Offset

Register	Offset
CLPCL10SR	1000h

20.7.4.1.17.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *PCL10 status* per cluster.

20.7.4.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL4	CL3	CL2	CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.17.4 Fields

Field	Function
31-5 —	Reserved
4 CL4	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state
3	Cluster 0 PCL10 Status.

Table continues on the next page...

Field	Function
CL3	0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state
2 CL2	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state
1 CL1	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state
0 CL0	Cluster 0 PCL10 Status. 0b - Cluster 0 is not in the PCL10 state 1b - Cluster 0 is in the PCL10 state

20.7.4.1.18 Cluster PCL10 (Stop) Set Control Register (CLPCL10SETR)

20.7.4.1.18.1 Offset

Register	Offset
CLPCL10SETR	1004h

20.7.4.1.18.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for requesting that a cluster be placed in the *PCL10* state. The true value read from the register means there is a pending cluster PCL10 request.

20.7.4.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_PCL10_RQ4	CL_PCL10_RQ3	CL_PCL10_RQ2	CL_PCL10_RQ1	CL_PCL10_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.18.4 Fields

Field	Function
31-5 —	Reserved
4 CL_PCL10_RQ 4	Write one to trigger cluster PCL10 request. This bit is clear by interrupt event or write-one event on CLPCL10CLRRn register. 0b - No request to put cluster4 in PCL10 state 1b - Request to put cluster4 in PCL10 state
3 CL_PCL10_RQ 3	Write one to trigger cluster PCL10 request. This bit is clear by interrupt event or write-one event on CLPCL10CLRRn register. 0b - No request to put cluster3 in PCL10 state 1b - Request to put cluster3 in PCL10 state
2 CL_PCL10_RQ 2	Write one to trigger cluster PCL10 request. This bit is clear by interrupt event or write-one event on CLPCL10CLRRn register. 0b - No request to put cluster2 in PCL10 state 1b - Request to put cluster2 in PCL10 state
1 CL_PCL10_RQ 1	Write one to trigger cluster PCL10 request. This bit is clear by interrupt event or write-one event on CLPCL10CLRRn register. 0b - No request to put cluster1 in PCL10 state 1b - Request to put cluster1 in PCL10 state
0 CL_PCL10_RQ 0	Write one to trigger cluster PCL10 request. This bit is clear by interrupt event or write-one event on CLPCL10CLRRn register. 0b - No request to put cluster0 in PCL10 state 1b - Request to put cluster0 in PCL10 state

20.7.4.1.19 Cluster PCL10 (Stop) Clear Control Register (CLPCL10CLRR)

20.7.4.1.19.1 Offset

Register	Offset
CLPCL10CLRR	1008h

20.7.4.1.19.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for waking up the cluster be placed in the *PCL10* state. The true value read from the register only means there is a pending cluster PCL10 request.

20.7.4.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_PCL10_RQ4	CL_PCL10_RQ3	CL_PCL10_RQ2	CL_PCL10_RQ1	CL_PCL10_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.19.4 Fields

Field	Function
31-5 —	Reserved
4 CL_PCL10_RQ4	Write one to cancel cluster PCL10 request. The bit is set by write-one-event to CLPCL10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL10 request for that cluster. 0b - Cluster4 is not receiving a PCL10 request from CLPCL10SETR 1b - Cluster4 is receiving a PCL10 request from CLPCL10SETR
3 CL_PCL10_RQ3	Write one to cancel cluster PCL10 request. The bit is set by write-one-event to CLPCL10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL10 request for that cluster. 0b - Cluster3 is not receiving a PCL10 request from CLPCL10SETR 1b - Cluster3 is receiving a PCL10 request from CLPCL10SETR
2 CL_PCL10_RQ2	Write one to cancel cluster PCL10 request. The bit is set by write-one-event to CLPCL10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL10 request for that cluster. 0b - Cluster2 is not receiving a PCL10 request from CLPCL10SETR 1b - Cluster2 is receiving a PCL10 request from CLPCL10SETR
1 CL_PCL10_RQ1	Write one to cancel cluster PCL10 request. The bit is set by write-one-event to CLPCL10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL10 request for that cluster. 0b - Cluster1 is not receiving a PCL10 request from CLPCL10SETR 1b - Cluster1 is receiving a PCL10 request from CLPCL10SETR
0 CL_PCL10_RQ0	Write one to cancel cluster PCL10 request. The bit is set by write-one-event to CLPCL10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL10 request for that cluster. 0b - Cluster0 is not receiving a PCL10 request from CLPCL10SETR 1b - Cluster0 is receiving a PCL10 request from CLPCL10SETR

20.7.4.1.20 Cluster PCL10 (Stop) Previous Status Register (CLPCL10PSR)

20.7.4.1.20.1 Offset

Register	Offset
CLPCL10PSR	100Ch

20.7.4.1.20.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for reporting previous *PCL10 status* per cluster. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_P_PCL10_4	CL_P_PCL10_3	CL_P_PCL10_2	CL_P_PCL10_1	CL_P_PCL10_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.20.4 Fields

Field	Function
31-5 —	Reserved
4 CL_P_PCL10_4	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.
3 CL_P_PCL10_3	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c.

Table continues on the next page...

Field	Function
	0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.
2 CL_P_PCL10_2	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.
1 CL_P_PCL10_1	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.
0 CL_P_PCL10_0	Cluster PCL10 previous status. The bit is set when corresponding CLPCL10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the PCL10 state before wake up by interrupt. 1b - Cluster was in the PCL10 state before wake up by interrupt.

20.7.4.1.21 Cluster PCL20 (Retain) Status Register (CLPCL20SR)

20.7.4.1.21.1 Offset

Register	Offset
CLPCL20SR	1020h

20.7.4.1.21.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *PCL20 status* per cluster.

20.7.4.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_P_PCL10_4	CL_P_PCL10_3	CL_P_PCL10_2	CL_P_PCL10_1	CL_P_PCL10_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.21.4 Fields

Field	Function
31-5 —	Reserved
4 CL_P_PCL10_4	Cluster PCL20 status 0b - Cluster4 is not in the PCL20 state 1b - Cluster4 is in the PCL20 state
3 CL_P_PCL10_3	Cluster PCL20 status 0b - Cluster3 is not in the PCL20 state 1b - Cluster3 is in the PCL20 state
2 CL_P_PCL10_2	Cluster PCL20 status 0b - Cluster2 is not in the PCL20 state 1b - Cluster2 is in the PCL20 state
1 CL_P_PCL10_1	Cluster PCL20 status 0b - Cluster1 is not in the PCL20 state 1b - Cluster1 is in the PCL20 state
0 CL_P_PCL10_0	Cluster PCL20 status 0b - Cluster0 is not in the PCL20 state 1b - Cluster0 is in the PCL20 state

20.7.4.1.22 Cluster PCL20 (Retain) Set Control Register (CLPCL20SETR)

20.7.4.1.22.1 Offset

Register	Offset
CLPCL20SETR	1024h

20.7.4.1.22.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for requesting that a cluster be placed in the *PCL20* state. The true value read from the register means there is a pending cluster PCL20 request.

20.7.4.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_PCL20_RQ4	CL_PCL20_RQ3	CL_PCL20_RQ2	CL_PCL20_RQ1	CL_PCL20_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.22.4 Fields

Field	Function
31-5 —	Reserved
4 CL_PCL20_RQ4	Write one to trigger cluster PCL20 request. This bit is clear by interrupt event or write-one event on CLPCL20CLRRn register. 0b - No Request to put Cluster4 in PCL20 state 1b - Request to put Cluster4 in PCL20 state
3 CL_PCL20_RQ3	Write one to trigger cluster PCL20 request. This bit is clear by interrupt event or write-one event on CLPCL20CLRRn register. 0b - No Request to put Cluster3 in PCL20 state 1b - Request to put Cluster3 in PCL20 state
2 CL_PCL20_RQ2	Write one to trigger cluster PCL20 request. This bit is clear by interrupt event or write-one event on CLPCL20CLRRn register. 0b - No Request to put Cluster2 in PCL20 state 1b - Request to put Cluster2 in PCL20 state
1 CL_PCL20_RQ1	Write one to trigger cluster PCL20 request. This bit is clear by interrupt event or write-one event on CLPCL20CLRRn register. 0b - No Request to put Cluster1 in PCL20 state 1b - Request to put Cluster1 in PCL20 state
0 CL_PCL20_RQ0	Write one to trigger cluster PCL20 request. This bit is clear by interrupt event or write-one event on CLPCL20CLRRn register. 0b - No Request to put Cluster0 in PCL20 state 1b - Request to put Cluster0 in PCL20 state

20.7.4.1.23 Cluster PCL20 (Retain) Clear Control Register (CLPCL20CLRR)

20.7.4.1.23.1 Offset

Register	Offset
CLPCL20CLRR	1028h

20.7.4.1.23.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for waking up the cluster be placed in the *PCL20* state. The true value read from the register only means there is a pending cluster PCL20 request.

20.7.4.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_PCL20_RQ4	CL_PCL20_RQ3	CL_PCL20_RQ2	CL_PCL20_RQ1	CL_PCL20_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.23.4 Fields

Field	Function
31-5 —	Reserved
4 CL_PCL20_RQ4	Write one to cancel cluster PCL20 request. The bit is set by write-one-event to CLPCL20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL20 request for that cluster. 0b - Cluster4 is receiving a PCL20 request from CLPCL20SETR 1b - Cluster4 is receiving a PCL20 request from CLPCL20SETR
3 CL_PCL20_RQ3	Write one to cancel cluster PCL20 request. The bit is set by write-one-event to CLPCL20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL20 request for that cluster. 0b - Cluster3 is receiving a PCL20 request from CLPCL20SETR 1b - Cluster3 is receiving a PCL20 request from CLPCL20SETR

Table continues on the next page...

Field	Function
2 CL_PCL20_RQ 2	Write one to cancel cluster PCL20 request. The bit is set by write-one-event to CLPCL20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL20 request for that cluster. 0b - Cluster2 is receiving a PCL20 request from CLPCL20SETR 1b - Cluster2 is receiving a PCL20 request from CLPCL20SETR
1 CL_PCL20_RQ 1	Write one to cancel cluster PCL20 request. The bit is set by write-one-event to CLPCL20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL20 request for that cluster. 0b - Cluster1 is receiving a PCL20 request from CLPCL20SETR 1b - Cluster1 is receiving a PCL20 request from CLPCL20SETR
0 CL_PCL20_RQ 0	Write one to cancel cluster PCL20 request. The bit is set by write-one-event to CLPCL20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster PCL20 request for that cluster. 0b - Cluster0 is receiving a PCL20 request from CLPCL20SETR 1b - Cluster0 is receiving a PCL20 request from CLPCL20SETR

20.7.4.1.24 Cluster PCL20 (Retain) Previous Status Register (CLPCL20PSR)

20.7.4.1.24.1 Offset

Register	Offset
CLPCL20PSR	102Ch

20.7.4.1.24.2 Function

This is one of a set of Cluster Power Management Registers. This register is used for reporting previous *PCL20 status* per cluster. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_P_PCL20_4	CL_P_PCL20_3	CL_P_PCL20_2	CL_P_PCL20_1	CL_P_PCL20_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.24.4 Fields

Field	Function
31-5 —	Reserved
4 CL_P_PCL20_4	Cluster PCL20 previous status. The bit is set when corresponding CLPCL20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 4 was not in the PCL20 state before wake up by interrupt. 1b - Cluster 4 was in the PCL20 state before wake up by interrupt.
3 CL_P_PCL20_3	Cluster PCL20 previous status. The bit is set when corresponding CLPCL20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 3 was not in the PCL20 state before wake up by interrupt. 1b - Cluster 3 was in the PCL20 state before wake up by interrupt.
2 CL_P_PCL20_2	Cluster PCL20 previous status. The bit is set when corresponding CLPCL20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 2 was not in the PCL20 state before wake up by interrupt. 1b - Cluster 2 was in the PCL20 state before wake up by interrupt.
1 CL_P_PCL20_1	Cluster PCL20 previous status. The bit is set when corresponding CLPCL20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 1 was not in the PCL20 state before wake up by interrupt. 1b - Cluster 1 was in the PCL20 state before wake up by interrupt.
0 CL_P_PCL20_0	Cluster PCL20 previous status. The bit is set when corresponding CLPCL20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 0 was not in the PCL20 state before wake up by interrupt. 1b - Cluster 0 was in the PCL20 state before wake up by interrupt.

20.7.4.1.25 Cluster Domain DM10 (Halt) Status Register (DMPDM10SR)

20.7.4.1.25.1 Offset

Register	Offset
DMPDM10SR	2000h

20.7.4.1.25.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *DM10 status* per cluster domain.

20.7.4.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL4	CL3	CL2	CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.25.4 Fields

Field	Function
31-5 —	Reserved
4 CL4	Cluster Domain 0 DM10 Status. 0b - Cluster Domain 4 is not in the DM10 state 1b - Cluster Domain 4 is in the DM10 state
3 CL3	Cluster Domain 0 DM10 Status. 0b - Cluster Domain 3 is not in the DM10 state 1b - Cluster Domain 3 is in the DM10 state
2 CL2	Cluster Domain 0 DM10 Status. 0b - Cluster Domain 2 is not in the DM10 state 1b - Cluster Domain 2 is in the DM10 state
1 CL1	Cluster Domain 0 DM10 Status. 0b - Cluster Domain 1 is not in the DM10 state 1b - Cluster Domain 1 is in the DM10 state
0	Cluster Domain 0 DM10 Status.

Field	Function
CL0	0b - Cluster Domain 0 is not in the DM10 state 1b - Cluster Domain 0 is in the DM10 state

20.7.4.1.26 Cluster Domain DM10 (Halt) Set Control Register (DMPDM10SETR)

20.7.4.1.26.1 Offset

Register	Offset
DMPDM10SETR	2004h

20.7.4.1.26.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for requesting that a cluster domain be placed in the *DM10* state. The true value read from the register means there is a pending cluster domain DM10 request.

20.7.4.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_DM10_RQ4	CL_DM10_RQ3	CL_DM10_RQ2	CL_DM10_RQ1	CL_DM10_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.26.4 Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Field	Function
4 CL_DM10_RQ4	Write one to trigger cluster domain DM10 request. This bit is clear by interrupt event or write-one event on DMPDM10CLRRn register. 0b - No Request to put Cluster Domain 4 is in DM10 state 1b - Request to put Cluster Domain 4 in DM10 state
3 CL_DM10_RQ3	Write one to trigger cluster domain DM10 request. This bit is clear by interrupt event or write-one event on DMPDM10CLRRn register. 0b - No Request to put Cluster Domain 3 is in DM10 state 1b - Request to put Cluster Domain 3 in DM10 state
2 CL_DM10_RQ2	Write one to trigger cluster domain DM10 request. This bit is clear by interrupt event or write-one event on DMPDM10CLRRn register. 0b - No Request to put Cluster Domain 2 is in DM10 state 1b - Request to put Cluster Domain 2 in DM10 state
1 CL_DM10_RQ1	Write one to trigger cluster domain DM10 request. This bit is clear by interrupt event or write-one event on DMPDM10CLRRn register. 0b - No Request to put Cluster Domain 1 is in DM10 state 1b - Request to put Cluster Domain 1 in DM10 state
0 CL_DM10_RQ0	Write one to trigger cluster domain DM10 request. This bit is clear by interrupt event or write-one event on DMPDM10CLRRn register. 0b - No Request to put Cluster Domain 0 is in DM10 state 1b - Request to put Cluster Domain 0 in DM10 state

20.7.4.1.27 Cluster Domain DM10 (Halt) Clear Control Register (DMPDM10CLRR)

20.7.4.1.27.1 Offset

Register	Offset
DMPDM10CLRR	2008h

20.7.4.1.27.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for waking up the cluster domain be placed in the *DM10* state. The true value read from the register only means there is a pending cluster domain DM10 request.

20.7.4.1.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_DM10_RQ4	CL_DM10_RQ3	CL_DM10_RQ2	CL_DM10_RQ1	CL_DM10_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.27.4 Fields

Field	Function
31-5 —	Reserved
4 CL_DM10_RQ4	Write one to cancel cluster domain DM10 request. The bit is set by write-one-event to DMPDM10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM10 request for that cluster domain. 0b - No DM10 request to Cluster Domain 4 1b - DM10 request to Cluster Domain 4
3 CL_DM10_RQ3	Write one to cancel cluster domain DM10 request. The bit is set by write-one-event to DMPDM10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM10 request for that cluster domain. 0b - No DM10 request to Cluster Domain 3 1b - DM10 request to Cluster Domain 3
2 CL_DM10_RQ2	Write one to cancel cluster domain DM10 request. The bit is set by write-one-event to DMPDM10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM10 request for that cluster domain. 0b - No DM10 request to Cluster Domain 2 1b - DM10 request to Cluster Domain 2
1 CL_DM10_RQ1	Write one to cancel cluster domain DM10 request. The bit is set by write-one-event to DMPDM10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM10 request for that cluster domain. 0b - No DM10 request to Cluster Domain 1 1b - DM10 request to Cluster Domain 1
0 CL_DM10_RQ0	Write one to cancel cluster domain DM10 request. The bit is set by write-one-event to DMPDM10SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM10 request for that cluster domain. 0b - No DM10 request to Cluster Domain 0 1b - DM10 request to Cluster Domain 0

20.7.4.1.28 Cluster Domain DM10 (Halt) Previous Status Register (DMPDM10PSR)

20.7.4.1.28.1 Offset

Register	Offset
DMPDM10PSR	200Ch

20.7.4.1.28.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for reporting previous *DM10 status* per cluster domain. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_P_DM10_4	CL_P_DM10_3	CL_P_DM10_2	CL_P_DM10_1	CL_P_DM10_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.28.4 Fields

Field	Function
31-5 —	Reserved
4 CL_P_DM10_4	Cluster DM10 previous status. The bit is set when corresponding DMPDM10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster Domain 4was not in the DM10 state before wake up by interrupt. 1b - Cluster Domain 4 is in the DM10 state before wake up by interrupt.

Table continues on the next page...

Field	Function
3 CL_P_DM10_3	Cluster DM10 previous status. The bit is set when corresponding DMPDM10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster Domain 3was not in the DM10 state before wake up by interrupt. 1b - Cluster Domain 3 is in the DM10 state before wake up by interrupt.
2 CL_P_DM10_2	Cluster DM10 previous status. The bit is set when corresponding DMPDM10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster Domain 2was not in the DM10 state before wake up by interrupt. 1b - Cluster Domain 2 is in the DM10 state before wake up by interrupt.
1 CL_P_DM10_1	Cluster DM10 previous status. The bit is set when corresponding DMPDM10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster Domain 1was not in the DM10 state before wake up by interrupt. 1b - Cluster Domain 1 is in the DM10 state before wake up by interrupt.
0 CL_P_DM10_0	Cluster DM10 previous status. The bit is set when corresponding DMPDM10SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster Domain 0was not in the DM10 state before wake up by interrupt. 1b - Cluster Domain 0 is in the DM10 state before wake up by interrupt.

20.7.4.1.29 Cluster Domain DM15 (Stop) Status Register (DMPDM15SR)

20.7.4.1.29.1 Offset

Register	Offset
DMPDM15SR	2020h

20.7.4.1.29.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *DM15 status* per cluster domain.

20.7.4.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL4	CL3	CL2	CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.29.4 Fields

Field	Function
31-5 —	Reserved
4 CL4	Cluster Domain 4 DM15 Status. 0b - Cluster Domain 4 is not in the DM15 state 1b - Cluster Domain 4 is in the DM15 state
3 CL3	Cluster Domain 3 DM15 Status. 0b - Cluster Domain 3 is not in the DM15 state 1b - Cluster Domain 3 is in the DM15 state
2 CL2	Cluster Domain 2 DM15 Status. 0b - Cluster Domain 2 is not in the DM15 state 1b - Cluster Domain 2 is in the DM15 state
1 CL1	Cluster Domain 1 DM15 Status. 0b - Cluster Domain 1 is not in the DM15 state 1b - Cluster Domain 1 is in the DM15 state
0 CL0	Cluster Domain 0 DM15 Status. 0b - Cluster Domain 0 is not in the DM15 state 1b - Cluster Domain 0 is in the DM15 state

20.7.4.1.30 Cluster Domain DM15 (Stop) Set Control Register (DMPDM15SETR)

20.7.4.1.30.1 Offset

Register	Offset
DMPDM15SETR	2024h

20.7.4.1.30.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for requesting that a cluster domain be placed in the *DM15* state. The true value read from the register means there is a pending cluster domain DM15 request.

20.7.4.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_DM15_RQ4	CL_DM15_RQ3	CL_DM15_RQ2	CL_DM15_RQ1	CL_DM15_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.30.4 Fields

Field	Function
31-5 —	Reserved
4 CL_DM15_RQ4	Write one to trigger cluster domain DM15 request. This bit is clear by interrupt event or write-one event on DMPDM15CLRRn register. 0b - No Request to put cluster domain 4 in DM15 state 1b - Request to put cluster domain 4 in DM15 state
3 CL_DM15_RQ3	Write one to trigger cluster domain DM15 request. This bit is clear by interrupt event or write-one event on DMPDM15CLRRn register. 0b - No Request to put cluster domain 3 in DM15 state 1b - Request to put cluster domain 3 in DM15 state
2 CL_DM15_RQ2	Write one to trigger cluster domain DM15 request. This bit is clear by interrupt event or write-one event on DMPDM15CLRRn register. 0b - No Request to put cluster domain 2 in DM15 state 1b - Request to put cluster domain 2 in DM15 state
1 CL_DM15_RQ1	Write one to trigger cluster domain DM15 request. This bit is clear by interrupt event or write-one event on DMPDM15CLRRn register. 0b - No Request to put cluster domain 1 in DM15 state 1b - Request to put cluster domain 1 in DM15 state
0 CL_DM15_RQ0	Write one to trigger cluster domain DM15 request. This bit is clear by interrupt event or write-one event on DMPDM15CLRRn register. 0b - No Request to put cluster domain 0 in DM15 state 1b - Request to put cluster domain 0 in DM15 state

20.7.4.1.31 Cluster Domain DM15 (Stop) Clear Control Register (DMPDM15CLRR)

20.7.4.1.31.1 Offset

Register	Offset
DMPDM15CLRR	2028h

20.7.4.1.31.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for waking up the cluster domain be placed in the *DM15* state. The true value read from the register only means there is a pending cluster domain DM15 request.

20.7.4.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_DM15_RQ4	CL_DM15_RQ3	CL_DM15_RQ2	CL_DM15_RQ1	CL_DM15_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.31.4 Fields

Field	Function
31-5 —	Reserved
4 CL_DM15_RQ4	Write one to cancel cluster domain DM15 request. The bit is set by write-one-event to DMPDM15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM15 request for that cluster domain. 0b - No DM15 request to cluster domain 4 1b - DM15 request to cluster domain 4

Table continues on the next page...

Field	Function
3 CL_DM15_RQ3	Write one to cancel cluster domain DM15 request. The bit is set by write-one-event to DMPDM15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM15 request for that cluster domain. 0b - No DM15 request to cluster domain 3 1b - DM15 request to cluster domain 3
2 CL_DM15_RQ2	Write one to cancel cluster domain DM15 request. The bit is set by write-one-event to DMPDM15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM15 request for that cluster domain. 0b - No DM15 request to cluster domain 2 1b - DM15 request to cluster domain 2
1 CL_DM15_RQ1	Write one to cancel cluster domain DM15 request. The bit is set by write-one-event to DMPDM15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM15 request for that cluster domain. 0b - No DM15 request to cluster domain 1 1b - DM15 request to cluster domain 1
0 CL_DM15_RQ0	Write one to cancel cluster domain DM15 request. The bit is set by write-one-event to DMPDM15SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM15 request for that cluster domain. 0b - No DM15 request to cluster domain 0 1b - DM15 request to cluster domain 0

20.7.4.1.32 Cluster Domain DM15 (Stop) Previous Status Register (DMPDM15PSR)

20.7.4.1.32.1 Offset

Register	Offset
DMPDM15PSR	202Ch

20.7.4.1.32.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for reporting previous *DM15 status* per cluster domain. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_P_DM15_4	CL_P_DM15_3	CL_P_DM15_2	CL_P_DM15_1	CL_P_DM15_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.32.4 Fields

Field	Function
31-5 —	Reserved
4 CL_P_DM15_4	Cluster DM15 previous status. The bit is set when corresponding DMPDM15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the DM15 state before wake up by interrupt. 1b - Cluster was in the DM15 state before wake up by interrupt.
3 CL_P_DM15_3	Cluster DM15 previous status. The bit is set when corresponding DMPDM15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the DM15 state before wake up by interrupt. 1b - Cluster was in the DM15 state before wake up by interrupt.
2 CL_P_DM15_2	Cluster DM15 previous status. The bit is set when corresponding DMPDM15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the DM15 state before wake up by interrupt. 1b - Cluster was in the DM15 state before wake up by interrupt.
1 CL_P_DM15_1	Cluster DM15 previous status. The bit is set when corresponding DMPDM15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the DM15 state before wake up by interrupt. 1b - Cluster was in the DM15 state before wake up by interrupt.
0 CL_P_DM15_0	Cluster DM15 previous status. The bit is set when corresponding DMPDM15SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster was not in the DM15 state before wake up by interrupt. 1b - Cluster was in the DM15 state before wake up by interrupt.

20.7.4.1.33 Cluster Domain DM20 (Retain) Status Register (DMPDM20SR)

20.7.4.1.33.1 Offset

Register	Offset
DMPDM20SR	2040h

20.7.4.1.33.2 Function

This is one of a set of cluster power management registers. This register is used for reporting *DM20 status* per cluster domain.

20.7.4.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL4	CL3	CL2	CL1	CL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.33.4 Fields

Field	Function
31-5 —	Reserved
4 CL4	Cluster Domain 4 DM20 Status. 0b - Cluster 4 is not in the DM20 state 1b - Cluster 4 is in the DM20 state
3 CL3	Cluster Domain 3 DM20 Status. 0b - Cluster 3 is not in the DM20 state 1b - Cluster 3 is in the DM20 state
2 CL2	Cluster Domain 2 DM20 Status. 0b - Cluster 2 is not in the DM20 state 1b - Cluster 2 is in the DM20 state
1 CL1	Cluster Domain 1 DM20 Status. 0b - Cluster 1 is not in the DM20 state 1b - Cluster 1 is in the DM20 state
0	Cluster Domain 0 DM20 Status.

Field	Function
CL0	0b - Cluster 0 is not in the DM20 state 1b - Cluster 0 is in the DM20 state

20.7.4.1.34 Cluster Domain DM20 (Retain) Set Control Register (DMPDM20SETR)

20.7.4.1.34.1 Offset

Register	Offset
DMPDM20SETR	2044h

20.7.4.1.34.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for requesting that a cluster domain be placed in the *DM20* state. The true value read from the register means there is a pending cluster domain DM20 request.

20.7.4.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_DM20_RQ4	CL_DM20_RQ3	CL_DM20_RQ2	CL_DM20_RQ1	CL_DM20_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.34.4 Fields

Field	Function
31-5 —	Reserved

Table continues on the next page...

Field	Function
4 CL_DM20_RQ4	Write one to trigger cluster domain DM20 request. This bit is clear by interrupt event or write-one event on DMPDM20CLRRn register. 0b - No request to put cluster domain 4 in DM20 state 1b - Request to put cluster domain 4 in DM20 state
3 CL_DM20_RQ3	Write one to trigger cluster domain DM20 request. This bit is clear by interrupt event or write-one event on DMPDM20CLRRn register. 0b - No request to put cluster domain 3 in DM20 state 1b - Request to put cluster domain 3 in DM20 state
2 CL_DM20_RQ2	Write one to trigger cluster domain DM20 request. This bit is clear by interrupt event or write-one event on DMPDM20CLRRn register. 0b - No request to put cluster domain 2 in DM20 state 1b - Request to put cluster domain 2 in DM20 state
1 CL_DM20_RQ1	Write one to trigger cluster domain DM20 request. This bit is clear by interrupt event or write-one event on DMPDM20CLRRn register. 0b - No request to put cluster domain 1 in DM20 state 1b - Request to put cluster domain 1 in DM20 state
0 CL_DM20_RQ0	Write one to trigger cluster domain DM20 request. This bit is clear by interrupt event or write-one event on DMPDM20CLRRn register. 0b - No request to put cluster domain 0 in DM20 state 1b - Request to put cluster domain 0 in DM20 state

20.7.4.1.35 Cluster Domain DM20 (Retain) Clear Control Register (DMPDM20CLRR)

20.7.4.1.35.1 Offset

Register	Offset
DMPDM20CLRR	2048h

20.7.4.1.35.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for waking up the cluster domain be placed in the *DM20* state. The true value read from the register only means there is a pending cluster domain DM20 request.

20.7.4.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_DM20_RQ4	CL_DM20_RQ3	CL_DM20_RQ2	CL_DM20_RQ1	CL_DM20_RQ0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.35.4 Fields

Field	Function
31-5 —	Reserved
4 CL_DM20_RQ4	Write one to cancel cluster domain DM20 request. The bit is set by write-one-event to DMPDM20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM20 request for that cluster domain. 1b - DM20 request to Cluster domain 4
3 CL_DM20_RQ3	Write one to cancel cluster domain DM20 request. The bit is set by write-one-event to DMPDM20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM20 request for that cluster domain. 1b - DM20 request to Cluster domain 3
2 CL_DM20_RQ2	Write one to cancel cluster domain DM20 request. The bit is set by write-one-event to DMPDM20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM20 request for that cluster domain. 1b - DM20 request to Cluster domain 2
1 CL_DM20_RQ1	Write one to cancel cluster domain DM20 request. The bit is set by write-one-event to DMPDM20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM20 request for that cluster domain. 1b - DM20 request to Cluster domain 1
0 CL_DM20_RQ0	Write one to cancel cluster domain DM20 request. The bit is set by write-one-event to DMPDM20SETR register. This bit is also clear by interrupt event. The read true value of the register bit means there is pending cluster domain DM20 request for that cluster domain. 1b - DM20 request to Cluster domain 0

20.7.4.1.36 Cluster Domain DM20 (Retain) Previous Status Register (DMPDM20PSR)

20.7.4.1.36.1 Offset

Register	Offset
DMPDM20PSR	204Ch

20.7.4.1.36.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for reporting previous *DM20 status* per cluster domain. It is used by software to know which power saving state it was in before wake up by interrupt.

20.7.4.1.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CL_P_DM20_4	CL_P_DM20_3	CL_P_DM20_2	CL_P_DM20_1	CL_P_DM20_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.36.4 Fields

Field	Function
31-5 —	Reserved
4 CL_P_DM20_4	Cluster DM20 previous status. The bit is set when corresponding DMPDM20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 4 was not in the DM20 state before wake up by interrupt. 1b - Cluster 4 was in the DM20 state before wake up by interrupt.
3 CL_P_DM20_3	Cluster DM20 previous status. The bit is set when corresponding DMPDM20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c.

Table continues on the next page...

Field	Function
	0b - Cluster 3 was not in the DM20 state before wake up by interrupt. 1b - Cluster 3 was in the DM20 state before wake up by interrupt.
2 CL_P_DM20_2	Cluster DM20 previous status. The bit is set when corresponding DMPDM20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 2 was not in the DM20 state before wake up by interrupt. 1b - Cluster 2 was in the DM20 state before wake up by interrupt.
1 CL_P_DM20_1	Cluster DM20 previous status. The bit is set when corresponding DMPDM20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 1 was not in the DM20 state before wake up by interrupt. 1b - Cluster 1 was in the DM20 state before wake up by interrupt.
0 CL_P_DM20_0	Cluster DM20 previous status. The bit is set when corresponding DMPDM20SR bit has 1 to 0 transition and it is caused by interrupt. The bit is w1c. 0b - Cluster 0 was not in the DM20 state before wake up by interrupt. 1b - Cluster 0 was in the DM20 state before wake up by interrupt.

20.7.4.1.37 Cluster Domain Warm Reset Control Register (RSTCR)

20.7.4.1.37.1 Offset

Register	Offset
RSTCR	C000h

20.7.4.1.37.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for requesting that a cluster domain be warm reset. The true value read from the register means there is pending cluster domain warm reset request; the bit value is cleared when the corresponding cluster domain warm reset sequence is complete.

20.7.4.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.37.4 Fields

Field	Function
31-1 —	Reserved
0 REQ	AIOP tile reset

20.7.4.1.38 Cluster Domain Warm Reset Disable Control Register (RSTEXPCR)

20.7.4.1.38.1 Offset

Register	Offset
RSTEXPCR	C020h

20.7.4.1.38.2 Function

This is one of a set of Cluster Domain Power Management Registers. This register is used for disabling a cluster domain from being warm reset.

20.7.4.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											DM_RST_DIS4	DM_RST_DIS3	DM_RST_DIS2	DM_RST_DIS1	DM_RST_DIS0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

20.7.4.1.38.4 Fields

Field	Function
31-5 —	Reserved
4 DM_RST_DIS4	Write one to disable cluster domain warm reset. 0b - Do not disable warm reset cluster domain 1b - Disable warm reset cluster domain
3 DM_RST_DIS3	Write one to disable cluster domain warm reset. 0b - Do not disable warm reset cluster domain 1b - Disable warm reset cluster domain
2 DM_RST_DIS2	Write one to disable cluster domain warm reset. 0b - Do not disable warm reset cluster domain 1b - Disable warm reset cluster domain
1 DM_RST_DIS1	Write one to disable cluster domain warm reset. 0b - Do not disable warm reset cluster domain 1b - Disable warm reset cluster domain
0 DM_RST_DIS0	Write one to disable cluster domain warm reset. 0b - Do not disable warm reset cluster domain 1b - Disable warm reset cluster domain

20.8 Initialization Information

This section describe PMU initialization information. PMU uses both POR reset and Hard reset for hardware initialization. Interrupt sampling, RUNN counter, and Core Group logic are initialized only through POR reset. SOC power management module and Core power management module are initialized through Hard reset. PMU CCSR/DCSR are reset either by POR reset or Hard reset depending on their functionality.

20.8.1 Low Power Modes Registers Initialization

The PMU is implemented to provide generic IP interfaces with the assumption that all IPs consistently follow the interfaces protocols. However, it has been observed that some IPs do not adhere to that assumption. In which cases, glue logic must be implemented in the PMU wrappers to resolve the differences between the IPs.

This section describes the hardware initializzation of some low power modes status register due to differences in IPs' implementation.

- TPH10SR initialization: When thread is still in reset or disabled, thread halted status is implied and the corresponding bit in TPH10SR is set until the thread comes out of

reset. There is no distinction in the logic if halted condition is due to thread in reset or thread disabled or thread power mode. If a power management halt is requested while the thread is in reset this inplies an automatic thread halted state and the corresponding bit in TPH10SR bit is set. If the power management halt request is cleared while the thread is still "halted in reset" then TPH10SR would continue to indicates thread halted as long as thread is still in reset. TPH10SR would eventually clear when the thread would come out of reset or boot hold off.

- PCPH15SR initialization: When core is still in reset or disabled, core stopped status is implied and the corresponding bit in PCPH15SR is set until the core comes out of reset. There is no distinction in the logic if stopped condition is due to core in reset or core disabled or core power mode. If a power management stop is requested while the core is in reset this inplies an automatic core stopped state and the corresponding bit in PCPH15SR bit is set. If the power management stop request is cleared while the core is still "stopped in reset" then PCPH15SR would continue to indicates core stopped as long as core is still in reset. PCPH15SR would eventually clear when the core would come out of reset or boot hold off.

Chapter 21
Quad Serial Peripheral Interface (QuadSPI)

21.1 QuadSPI as implemented on the chip

Module Configuration Register (QuadSPI_MCR)

Table 21-1. Module Configuration Register (QuadSPI_MCR)

Bit (31-24)	SCLKCFG	Description
31-29	Coarse delay	3'b000 - No delay 3'b001 - 90 degree dqs shift delay 3'b010 - 1 cycle dqs shift delay 3'b011 - 1 cycle + 90 degree dqs shift delay 3'b100 - 2 cycles dqs shift delay 3'b101 - 2 cycles + 90 degree dqs shift delay 3'b110 - Reserved 3'b111 - No delay
28-27	Reserved	
26	Shift delay	1'b0 - No delay 1'b1 - 90 degree dqs shift dela
25	Delay Chain	1'b0 - Use Delay chain 1'b1 - Don't use Delay chain
24	Loopback DQS	1'b0 - dqs is direct from PAD 1'b1 - internal loopback dqs

SOC Configuration Register (QuadSPI_SOCCR)

Table 21-2. SOC Configuration Register (QuadSPI_SOCCR)

Bit	Name	Description
7-0	Fine delay chain DQS FA	8'b00000000 - Delay of 1 mux

Table continues on the next page...

Table 21-2. SOC Configuration Register (QuadSPI_SOCCR) (continued)

Bit	Name	Description
		8'b00000001 - Delay of 1 buffer and 2 mux 8'b00000010 - Delay of 2 buffer and 3 mux 8'b00000011 - Delay of 3 buffer and 4 mux 8'b00000100 - Delay of 4 buffer and 5 mux 8'b00000101 - Delay of 5 buffer and 6 mux 8'b00000110 - Delay of 6 buffer and 7 mux 8'b00000111 - Delay of 7 buffer and 8 mux . . . 8'b00111111- Delay of 63 buffer and 64 mux 8'b01000000-8'b11111111 - Reserved
15-8	Reserved	
23-16	Fine delay chain DQS FB	8'b00000000 - Delay of 1 mux 8'b00000001 - Delay of 1 buffer and 2 mux 8'b00000010 - Delay of 2 buffer and 3 mux 8'b00000011 - Delay of 3 buffer and 4 mux 8'b00000100 - Delay of 4 buffer and 5 mux 8'b00000101 - Delay of 5 buffer and 6 mux 8'b00000110 - Delay of 6 buffer and 7 mux 8'b00000111 - Delay of 7 buffer and 8 mux . . . 8'b00111111- Delay of 63 buffer and 64 mux 8'b01000000-8'b11111111 - Reserved
31-24	Reserved	

Please refer IFC, QSPI Address Remapping for more details on QSPI Address Remapping.

21.2 Introduction

The Quad Serial Peripheral Interface (QuadSPI) block acts as an interface to a single or two external serial flash devices, each with up to eight bidirectional data lines.

21.2.1 Features

The QuadSPI supports the following features:

- Flexible sequence engine to support various flash vendor devices. As there is no specific standard, the module supports various kind of flashes from different vendors. Refer Serial Flash Devices for example sequences.
- Single, dual, quad and octal modes of operation.
- Double data rate (DDR)/Double trasfer rate (DTR) mode wherein the data is generated on every edge of the serial flash clock.
- Support for flash data strobe signal for data sampling in DDR and Single data rate (SDR) mode.
- Two identical serial flash devices can be connected and accessed in parallel for data read operations, forming one (virtual) flash memory with doubled readout bandwidth.
- AHB master to read RX Buffer data via AMBA AHB bus (64-bit width interface) or IP registers space (32-bit access).
    - AHB master can be a DMA with configurable inner loop size.
- Multimaster accesses with priority
    - Flexible and configurable buffer for each master
- Multiple interrupt conditions (see Table 21-12)
- All AHB accesses to flash devices are directly memory mapped to the chip system memory.
- Programmable sequence engine to cater to future command/protocol changes and able to support all existing vendor commands and operations. Software needs to select the corresponding sequence according to the connected flash device.
    - Supports 3-byte and 4-byte addressing.

21.2.2 Block Diagram

The following figure is a block diagram of the Quad Serial Peripheral Interface (QuadSPI) module.

Figure 21-1. QuadSPI Block Diagram

21.2.3 QuadSPI Modes of Operation

For power management through IPS interface access and correct config register programming sequences, QuadSPI supports three modes: normal, module disable and stop mode.

- Normal Mode can be used for write or read accesses to an external serial flash device.
  - Serial Flash Write: Data can be programmed into the flash via the IP interface only. Refer to Flash Programming for further details.
  - Serial Flash Read: Read the contents of the serial flash device. Two separate read channels are available via RX Buffer and AHB Buffer, see Flash Read.
- Stop Mode: The mode is used for power management. When a request is made to enter Stop Mode, the QuadSPI block acknowledges the request and completes the SFM Command in progress, then the system clocks to the QuadSPI block may be shut off
- Module Disable Mode: The mode is used for power management. The clock to the non-memory mapped logic in the QuadSPI can be stopped while in Module Disable Mode. The module enters the mode by setting QSPI_MCR[MDIS] or when a request is asserted by an external controller while QSPI_MCR[DOZE] is set.

21.2.4 Acronyms and Abbreviations

The following table contains acronyms and abbreviations used in this document.

Table 21-3. Acronyms and Abbreviations

Terms	Description
AHB	Advanced High-performance Bus, version of AMBA
AMBA	Advanced Microcontroller Bus Architecture
BE	Big Endian Byte Ordering
CS	Chip Select
DMA	Direct Memory Access
MB	Megabyte. Each MB is 1024 * 1024 bytes
IFM	Individual Flash Mode
PFM	Parallel Flash Mode
LSB	Least Significant Bit

Table continues on the next page...

Table 21-3. Acronyms and Abbreviations (continued)

Terms	Description
MSB	Most Significant Bit
PCS	Peripheral Chip Select
QSPI, QuadSPI	Quad Serial Peripheral Interface
SCK	Serial Communications Clock
w1c	Write 1 to clear, writing a 1 to this field resets the flag
I/O	Input output. In this document, I/O lines are also referred as **pads**.

21.2.5 Glossary for QuadSPI module

Table 21-4. Glossary

Term	Definition
AHB Command	An AHB Command is a SFM Command triggered by a read access to the address range belonging to the memory mapped access defined in Table 21-9. Refer to AHB Commands for details.
Asserted	A signal that is asserted is in its active state. An active low signal changes from logic level one to logic level zero when asserted, and an active high signal changes from logic level zero to logic level one.
Clear	To clear a bit or bits means to establish logic level zero on the bit or bits.
Clock Phase	Determines when the data should be sampled relative to the active edge of SCK
Clock Polarity	Determines the idle state of the SCK signal.
Drain	To remove entries from a FIFO by software or hardware.
Endianness	Byte Ordering scheme.
Field	Two or more register bits grouped together.
Fill	To add entries to a FIFO by software or hardware.
Host	Refers to another functional block in the device containing the QuadSPI module
Instruction Code	8 bits defining the type of command to be executed.
IP Command	An IP Command is a SFM Command triggered by writing into the QSPI_IPCR[SEQID] field.
Logic level one	The voltage that corresponds to Boolean true (1) state.
Logic level zero	The voltage that corresponds to Boolean false (0) state.
Negated	A signal that is negated is in its inactive state. An active low signal changes from logic level 0 to logic level 1 when negated, and an active high signal changes from logic level 1 to logic level 0.
QSPI_AMBA_BASE	The first address of the serial flash device as presented to the QuadSPI controller. This may be the base of the serial flash in the system address map or it may be a remapping, for instance to 0x0, done by the system. Refer to the system address map.
QSPI_ARDB_BASE	First address of QuadSPI Rx Buffer on system memory map.
Set	To set a bit or bits means to establish logic level one on the bit or bits.
RX Buffer PUSH Event	Addition of valid entries into the RX Buffer. In the default case each Buffer PUSH Event adds 2 entries to the RX Buffer since the interface to the serial clock domain is 64 bits in width. Depending on the number of bytes read from the serial flash device it is possible for the very last Buffer PUSH Event that only one entry is added. The QSPI_RBSR[RDBFL] field is incremented by the number of entries added to the RX Buffer.

Table continues on the next page...

Table 21-4. Glossary (continued)

Term	Definition
RX Buffer POP Event	Removal of valid entries from the RX Buffer. Each Buffer POP Event removes (QSPI_RBCT[WMRK] + 1) valid entries from the buffer. The QSPI_RBSR[RDBFL] field is decremented by the same number and the QSPI_RBSR[RDCTR] field is incremented accordingly.
Individual Flash Mode	Access to a single, individual serial flash device. Refer to Serial Flash Access Schemes for details.
Parallel Flash Mode	Read access to two serial flash devices attached to the QuadSPI module in parallel. Refer to Serial Flash Access Schemes for details.
SFM Command	Serial Flash Memory Command. A SFM command consists of an instruction code and all other parameters (for example, size or mode bytes) needed for that specific instruction code. Triggering a command either initiates a transaction on the external serial flash or results in an error. Refer to Table 21-22 for details on errors.
Single/Dual/Quad/Octal Instructions	Depending on the serial flash device connected to the QuadSPI module there will be instructions using a different number of data lines. • Single pad: Single line I/O with one data out and one data in line to/from the serial flash device. • Dual pad: Dual line I/O with two bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI module • Quad pad: Quad line I/O with 4 bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI • Octal pad: Octal line I/O with 8 bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI
Transaction	A transaction consists of all flags, data and signals in either direction to execute a command for an attached serial flash device. It is a combination of chip select, sclk, instruction code, address, mode- and/or dummy bytes, transmit and/or receive data.
LUT	Look-up table.

21.3 External Signal Description

This section provides the external signal information of the QuadSPI module.

The following table lists the external signals belonging to the module in conjunction with the different modes of operation.

Table 21-5. Signal Properties

Signal Name	Function	Direction	Description
PCSFA1	Peripheral Chip Select Flash A1	O	This signal is the chip select for the serial flash device A1. A1 represents the first device in a dual-die package flash A or the first of the two flash devices that share IOFA. Refer to Dual Die Flashes for more details.
PCSFA2	Peripheral Chip Select Flash A2	O	This signal is the chip select for the serial flash device A2. A2 represents the second device in a dual-die package flash A or the second of the two flash devices that share IOFA. Refer to Dual Die Flashes for more details.
PCSFB1	Peripheral Chip Select Flash B1	O	This signal is the chip select for the serial flash device B1. B1 represents the first device in a dual-die package flash B or the first of the two flash devices that share IOFB. Refer to Dual Die Flashes for more details.

Table continues on the next page...

Table 21-5. Signal Properties (continued)

Signal Name	Function	Direction	Description
PCSFB2	Peripheral Chip Select Flash B2	O	This signal is the chip select for the serial flash device B2. B2 represents the second device in a dual-die package flash B or the second of the two flash devices that share IOFB. Refer to Dual Die Flashes for more details.
SCKFA	Serial Clock Flash A	O	This signal is the serial clock output to the serial flash device A.
SCKFB	Serial Clock Flash B	O	This signal is the serial clock output to the serial flash device B.
IOFA[7:0]	Serial I/O Flash A	I/O	These signals are the data I/O lines to/from the serial flash device A. Refer to Driving External Signals for details about the signal drive and timing behavior. Note that the signal pins of the serial flash device may change their function according to the SFM Command executed, leaving them as control inputs when Single and Dual Instructions are executed. The module supports driving these inputs to dedicated values. In single I/O mode, QuadSPI drives data on IOFA[0] and expects data on IOFA[1].
IOFB[7:0]	Serial I/O Flash B	I/O	These signals are the data I/O lines to/from the serial flash device B. Refer to Driving External Signals for details about the signal drive and timing behavior. Note that the signal pins of the serial flash device may change their function according to the SFM Command executed, leaving them as control inputs when Single and Dual Instructions are executed. The module supports driving these inputs to dedicated values. In single I/O mode, QuadSPI drives data on IOFB[0] and expects data on IOFB[1].
DQSFA	Data Strobe signal Flash A	I	Data strobe signal for port A. Some flash vendors provide the DQS signal to which the read data is aligned in DDR mode.
DQSFB	Data Strobe signal Flash B	I	Data strobe signal for port B. Some flash vendors provide the DQS signal to which the read data is aligned in DDR mode.

21.3.1 Driving External Signals

The different phases of the serial flash access scheme are shown in the following figure.

Figure 21-2. Serial Flash Access Scheme

The different phases and the I/O driving characteristics of the QuadSPI module are characterized in the following way:

- IDLE: Serial flash device not selected. No interaction with the serial flash device. All IOFx signals driven.
- INSTRUCTION: Serial flash device selected. The instruction is sent to the serial flash device. All IOFx signals are driven.

- ADDRESS: Serial Flash Address is sent to the device. All IOFx signals are driven. Note that this phase is not applicable for all SFM Commands.
- MODE: Mode bytes are sent to the serial flash device. All IOFx signals are driven. Note that this phase is not applicable for all SFM Commands.
- DUMMY: Dummy clocks are provided to the serial flash device. Refer to the Figure 21-2 for the IOFx signals driven. The actual data lines required for the SFM Command executed are not driven for data read commands. Note that this phase is not applicable for all SFM Commands.
- DATA: Serial flash data are sent to or received from the serial flash device. Refer to the preceding figure for the IOFx signals driven. The actual data lines required for the SFM Command executed are not driven for data read commands. Note that this phase is not applicable for all SFM Commands.

The PCSFx and SCKFx signals are driven permanently throughout all the phases.

In Individual Flash Mode this applies to the selected flash device. In Parallel Flash Mode this applies to both serial flash devices simultaneously.

21.4 Memory Map and Register Definition

This section provides the memory map and register definitions of the QuadSPI module.

21.4.1 Register Write Access

This section describes the write access restriction terms that apply to all registers, which can be one of the following:

- **Register Write Access Restriction**

  For each register bit and register field, the write access conditions are specified in the detailed register description. A description of the write access conditions is given in the following table. If, for a specific register bit or field, none of the given write access conditions is fulfilled, any write attempt to this register bit or field is ignored without any notification. The values of the bits or fields are not changed.

  The condition term [A or B] indicates that the register or field can be written to if at least one of the conditions is fulfilled.

Table 21-6. Register Write Access Restrictions

Condition	Description
Anytime	No write access restriction.
Disabled Mode	Write access only if *QSPI_MCR[MDIS] = 1*.
Normal Mode	Write access only if the module is in *Normal Mode*.

- **Register Write Access Requirements**

  All registers can be accessed with 8-bit, 16-bit, and 32-bit wide operations. For some of the registers, at least a 16/32-bit wide write access is required to ensure correct operation. This write access requirement is stated in the detailed register description for each register affected.

21.4.2 Peripheral Bus Register Descriptions

This section provides the memory map and register definitions of the QuadSPI module.

QuadSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_0000	Module Configuration Register (QuadSPI_MCR)	32	R/W	See section	21.4.2.1/ 1869
20C_0008	IP Configuration Register (QuadSPI_IPCR)	32	R/W	0000_0000h	21.4.2.2/ 1872
20C_000C	Flash Configuration Register (QuadSPI_FLSHCR)	32	R/W	0000_0303h	21.4.2.3/ 1873
20C_0010	Buffer0 Configuration Register (QuadSPI_BUF0CR)	32	R/W	See section	21.4.2.4/ 1874
20C_0014	Buffer1 Configuration Register (QuadSPI_BUF1CR)	32	R/W	See section	21.4.2.5/ 1875
20C_0018	Buffer2 Configuration Register (QuadSPI_BUF2CR)	32	R/W	See section	21.4.2.6/ 1876
20C_001C	Buffer3 Configuration Register (QuadSPI_BUF3CR)	32	R/W	See section	21.4.2.7/ 1877
20C_0020	Buffer Generic Configuration Register (QuadSPI_BFGENCR)	32	R/W	0000_0000h	21.4.2.8/ 1878
20C_0024	SOC Configuration Register (QuadSPI_SOCCR)	32	R/W	0000_0000h	21.4.2.9/ 1879

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_0030	Buffer0 Top Index Register (QuadSPI_BUF0IND)	32	R/W	0000_0000h	21.4.2.10/ 1879
20C_0034	Buffer1 Top Index Register (QuadSPI_BUF1IND)	32	R/W	0000_0000h	21.4.2.11/ 1880
20C_0038	Buffer2 Top Index Register (QuadSPI_BUF2IND)	32	R/W	0000_0000h	21.4.2.12/ 1881
20C_0100	Serial Flash Address Register (QuadSPI_SFAR)	32	R/W	0000_0000h	21.4.2.13/ 1881
20C_0108	Sampling Register (QuadSPI_SMPR)	32	R/W	0000_0000h	21.4.2.14/ 1882
20C_010C	RX Buffer Status Register (QuadSPI_RBSR)	32	R	0000_0000h	21.4.2.15/ 1884
20C_0110	RX Buffer Control Register (QuadSPI_RBCT)	32	R/W	0000_0000h	21.4.2.16/ 1884
20C_0150	TX Buffer Status Register (QuadSPI_TBSR)	32	R	0000_0000h	21.4.2.17/ 1886
20C_0154	TX Buffer Data Register (QuadSPI_TBDR)	32	R/W	0000_0000h	21.4.2.18/ 1886
20C_015C	Status Register (QuadSPI_SR)	32	R	0000_3800h	21.4.2.19/ 1888
20C_0160	Flag Register (QuadSPI_FR)	32	w1c	0800_0000h	21.4.2.20/ 1891
20C_0164	Interrupt and DMA Request Select and Enable Register (QuadSPI_RSER)	32	R/W	0000_0000h	21.4.2.21/ 1894
20C_0168	Sequence Suspend Status Register (QuadSPI_SPNDST)	32	R	0000_0000h	21.4.2.22/ 1898
20C_016C	Sequence Pointer Clear Register (QuadSPI_SPTRCLR)	32	R/W	0000_0000h	21.4.2.23/ 1900
20C_0180	Serial Flash A1 Top Address (QuadSPI_SFA1AD)	32	R/W	See section	21.4.2.24/ 1901
20C_0184	Serial Flash A2 Top Address (QuadSPI_SFA2AD)	32	R/W	See section	21.4.2.25/ 1901
20C_0188	Serial Flash B1 Top Address (QuadSPI_SFB1AD)	32	R/W	See section	21.4.2.26/ 1902
20C_018C	Serial Flash B2 Top Address (QuadSPI_SFB2AD)	32	R/W	See section	21.4.2.27/ 1902
20C_0190	Data Learn Pattern Register (QuadSPI_DLPR)	32	R/W	AA55_3443h	21.4.2.28/ 1903
20C_0200	RX Buffer Data Register (QuadSPI_RBDR0)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0204	RX Buffer Data Register (QuadSPI_RBDR1)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0208	RX Buffer Data Register (QuadSPI_RBDR2)	32	R	0000_0000h	21.4.2.29/ 1903

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_020C	RX Buffer Data Register (QuadSPI_RBDR3)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0210	RX Buffer Data Register (QuadSPI_RBDR4)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0214	RX Buffer Data Register (QuadSPI_RBDR5)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0218	RX Buffer Data Register (QuadSPI_RBDR6)	32	R	0000_0000h	21.4.2.29/ 1903
20C_021C	RX Buffer Data Register (QuadSPI_RBDR7)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0220	RX Buffer Data Register (QuadSPI_RBDR8)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0224	RX Buffer Data Register (QuadSPI_RBDR9)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0228	RX Buffer Data Register (QuadSPI_RBDR10)	32	R	0000_0000h	21.4.2.29/ 1903
20C_022C	RX Buffer Data Register (QuadSPI_RBDR11)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0230	RX Buffer Data Register (QuadSPI_RBDR12)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0234	RX Buffer Data Register (QuadSPI_RBDR13)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0238	RX Buffer Data Register (QuadSPI_RBDR14)	32	R	0000_0000h	21.4.2.29/ 1903
20C_023C	RX Buffer Data Register (QuadSPI_RBDR15)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0240	RX Buffer Data Register (QuadSPI_RBDR16)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0244	RX Buffer Data Register (QuadSPI_RBDR17)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0248	RX Buffer Data Register (QuadSPI_RBDR18)	32	R	0000_0000h	21.4.2.29/ 1903
20C_024C	RX Buffer Data Register (QuadSPI_RBDR19)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0250	RX Buffer Data Register (QuadSPI_RBDR20)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0254	RX Buffer Data Register (QuadSPI_RBDR21)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0258	RX Buffer Data Register (QuadSPI_RBDR22)	32	R	0000_0000h	21.4.2.29/ 1903
20C_025C	RX Buffer Data Register (QuadSPI_RBDR23)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0260	RX Buffer Data Register (QuadSPI_RBDR24)	32	R	0000_0000h	21.4.2.29/ 1903

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_0264	RX Buffer Data Register (QuadSPI_RBDR25)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0268	RX Buffer Data Register (QuadSPI_RBDR26)	32	R	0000_0000h	21.4.2.29/ 1903
20C_026C	RX Buffer Data Register (QuadSPI_RBDR27)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0270	RX Buffer Data Register (QuadSPI_RBDR28)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0274	RX Buffer Data Register (QuadSPI_RBDR29)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0278	RX Buffer Data Register (QuadSPI_RBDR30)	32	R	0000_0000h	21.4.2.29/ 1903
20C_027C	RX Buffer Data Register (QuadSPI_RBDR31)	32	R	0000_0000h	21.4.2.29/ 1903
20C_0300	LUT Key Register (QuadSPI_LUTKEY)	32	R/W	5AF0_5AF0h	21.4.2.30/ 1904
20C_0304	LUT Lock Configuration Register (QuadSPI_LCKCR)	32	R/W	0000_0002h	21.4.2.31/ 1905
20C_0310	Look-up Table register (QuadSPI_LUT0)	32	R/W	See section	21.4.2.32/ 1906
20C_0314	Look-up Table register (QuadSPI_LUT1)	32	R/W	See section	21.4.2.32/ 1906
20C_0318	Look-up Table register (QuadSPI_LUT2)	32	R/W	See section	21.4.2.32/ 1906
20C_031C	Look-up Table register (QuadSPI_LUT3)	32	R/W	See section	21.4.2.32/ 1906
20C_0320	Look-up Table register (QuadSPI_LUT4)	32	R/W	See section	21.4.2.32/ 1906
20C_0324	Look-up Table register (QuadSPI_LUT5)	32	R/W	See section	21.4.2.32/ 1906
20C_0328	Look-up Table register (QuadSPI_LUT6)	32	R/W	See section	21.4.2.32/ 1906
20C_032C	Look-up Table register (QuadSPI_LUT7)	32	R/W	See section	21.4.2.32/ 1906
20C_0330	Look-up Table register (QuadSPI_LUT8)	32	R/W	See section	21.4.2.32/ 1906
20C_0334	Look-up Table register (QuadSPI_LUT9)	32	R/W	See section	21.4.2.32/ 1906
20C_0338	Look-up Table register (QuadSPI_LUT10)	32	R/W	See section	21.4.2.32/ 1906
20C_033C	Look-up Table register (QuadSPI_LUT11)	32	R/W	See section	21.4.2.32/ 1906
20C_0340	Look-up Table register (QuadSPI_LUT12)	32	R/W	See section	21.4.2.32/ 1906

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_0344	Look-up Table register (QuadSPI_LUT13)	32	R/W	See section	21.4.2.32/ 1906
20C_0348	Look-up Table register (QuadSPI_LUT14)	32	R/W	See section	21.4.2.32/ 1906
20C_034C	Look-up Table register (QuadSPI_LUT15)	32	R/W	See section	21.4.2.32/ 1906
20C_0350	Look-up Table register (QuadSPI_LUT16)	32	R/W	See section	21.4.2.32/ 1906
20C_0354	Look-up Table register (QuadSPI_LUT17)	32	R/W	See section	21.4.2.32/ 1906
20C_0358	Look-up Table register (QuadSPI_LUT18)	32	R/W	See section	21.4.2.32/ 1906
20C_035C	Look-up Table register (QuadSPI_LUT19)	32	R/W	See section	21.4.2.32/ 1906
20C_0360	Look-up Table register (QuadSPI_LUT20)	32	R/W	See section	21.4.2.32/ 1906
20C_0364	Look-up Table register (QuadSPI_LUT21)	32	R/W	See section	21.4.2.32/ 1906
20C_0368	Look-up Table register (QuadSPI_LUT22)	32	R/W	See section	21.4.2.32/ 1906
20C_036C	Look-up Table register (QuadSPI_LUT23)	32	R/W	See section	21.4.2.32/ 1906
20C_0370	Look-up Table register (QuadSPI_LUT24)	32	R/W	See section	21.4.2.32/ 1906
20C_0374	Look-up Table register (QuadSPI_LUT25)	32	R/W	See section	21.4.2.32/ 1906
20C_0378	Look-up Table register (QuadSPI_LUT26)	32	R/W	See section	21.4.2.32/ 1906
20C_037C	Look-up Table register (QuadSPI_LUT27)	32	R/W	See section	21.4.2.32/ 1906
20C_0380	Look-up Table register (QuadSPI_LUT28)	32	R/W	See section	21.4.2.32/ 1906
20C_0384	Look-up Table register (QuadSPI_LUT29)	32	R/W	See section	21.4.2.32/ 1906
20C_0388	Look-up Table register (QuadSPI_LUT30)	32	R/W	See section	21.4.2.32/ 1906
20C_038C	Look-up Table register (QuadSPI_LUT31)	32	R/W	See section	21.4.2.32/ 1906
20C_0390	Look-up Table register (QuadSPI_LUT32)	32	R/W	See section	21.4.2.32/ 1906
20C_0394	Look-up Table register (QuadSPI_LUT33)	32	R/W	See section	21.4.2.32/ 1906
20C_0398	Look-up Table register (QuadSPI_LUT34)	32	R/W	See section	21.4.2.32/ 1906

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_039C	Look-up Table register (QuadSPI_LUT35)	32	R/W	See section	21.4.2.32/ 1906
20C_03A0	Look-up Table register (QuadSPI_LUT36)	32	R/W	See section	21.4.2.32/ 1906
20C_03A4	Look-up Table register (QuadSPI_LUT37)	32	R/W	See section	21.4.2.32/ 1906
20C_03A8	Look-up Table register (QuadSPI_LUT38)	32	R/W	See section	21.4.2.32/ 1906
20C_03AC	Look-up Table register (QuadSPI_LUT39)	32	R/W	See section	21.4.2.32/ 1906
20C_03B0	Look-up Table register (QuadSPI_LUT40)	32	R/W	See section	21.4.2.32/ 1906
20C_03B4	Look-up Table register (QuadSPI_LUT41)	32	R/W	See section	21.4.2.32/ 1906
20C_03B8	Look-up Table register (QuadSPI_LUT42)	32	R/W	See section	21.4.2.32/ 1906
20C_03BC	Look-up Table register (QuadSPI_LUT43)	32	R/W	See section	21.4.2.32/ 1906
20C_03C0	Look-up Table register (QuadSPI_LUT44)	32	R/W	See section	21.4.2.32/ 1906
20C_03C4	Look-up Table register (QuadSPI_LUT45)	32	R/W	See section	21.4.2.32/ 1906
20C_03C8	Look-up Table register (QuadSPI_LUT46)	32	R/W	See section	21.4.2.32/ 1906
20C_03CC	Look-up Table register (QuadSPI_LUT47)	32	R/W	See section	21.4.2.32/ 1906
20C_03D0	Look-up Table register (QuadSPI_LUT48)	32	R/W	See section	21.4.2.32/ 1906
20C_03D4	Look-up Table register (QuadSPI_LUT49)	32	R/W	See section	21.4.2.32/ 1906
20C_03D8	Look-up Table register (QuadSPI_LUT50)	32	R/W	See section	21.4.2.32/ 1906
20C_03DC	Look-up Table register (QuadSPI_LUT51)	32	R/W	See section	21.4.2.32/ 1906
20C_03E0	Look-up Table register (QuadSPI_LUT52)	32	R/W	See section	21.4.2.32/ 1906
20C_03E4	Look-up Table register (QuadSPI_LUT53)	32	R/W	See section	21.4.2.32/ 1906
20C_03E8	Look-up Table register (QuadSPI_LUT54)	32	R/W	See section	21.4.2.32/ 1906
20C_03EC	Look-up Table register (QuadSPI_LUT55)	32	R/W	See section	21.4.2.32/ 1906
20C_03F0	Look-up Table register (QuadSPI_LUT56)	32	R/W	See section	21.4.2.32/ 1906

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
20C_03F4	Look-up Table register (QuadSPI_LUT57)	32	R/W	See section	21.4.2.32/ 1906
20C_03F8	Look-up Table register (QuadSPI_LUT58)	32	R/W	See section	21.4.2.32/ 1906
20C_03FC	Look-up Table register (QuadSPI_LUT59)	32	R/W	See section	21.4.2.32/ 1906
20C_0400	Look-up Table register (QuadSPI_LUT60)	32	R/W	See section	21.4.2.32/ 1906
20C_0404	Look-up Table register (QuadSPI_LUT61)	32	R/W	See section	21.4.2.32/ 1906
20C_0408	Look-up Table register (QuadSPI_LUT62)	32	R/W	See section	21.4.2.32/ 1906
20C_040C	Look-up Table register (QuadSPI_LUT63)	32	R/W	See section	21.4.2.32/ 1906

21.4.2.1 Module Configuration Register (QuadSPI_MCR)

The QuadSPI_MCR holds configuration data associated with QuadSPI operation.

Write:

- *SCLKCFG: Disabled Mode*
- *ISD3FB, ISD2FB,ISD3FA, ISD2FA: Disabled Mode*
- *All other fields: Anytime*

Address: 20C_0000h base + 0h offset = 20C_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCLKCFG								Reserved				ISD3FB	ISD2FB	ISD3FA	ISD2FA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOZE	MDIS	Reserved		0	0	Reserved	Reserved	DDR_EN	DQS_EN	Reserved	Reserved	END_CFG		SWRSTHD	SWRSTSD
W					CLR_TXF	CLR_RXF										
Reset	0	1	0	0	0	0	0	0	0	0	0	0	*	*	0	0

* Notes:
- END_CFG field: See the module configuration for the device specific reset value.

QuadSPI_MCR field descriptions

Field	Description
31–24 SCLKCFG	Serial Clock Configuration. This field configuration is chip specific. For details, refer to chip-specific QuadSPI information. It may be used for dividing clocks.

Table continues on the next page...

QuadSPI_MCR field descriptions (continued)

Field	Description
23–20 Reserved	This field is reserved.
19 ISD3FB	Idle Signal Drive IOFB[3] Flash B. This bit determines the logic level the IOFB[3] output of the QuadSPI module is driven to in the inactive state. Refer to Driving Flash Control Signals in Single and Dual Mode. 0 IOFB[3] is driven to logic L 1 IOFB[3] is driven to logic H
18 ISD2FB	Idle Signal Drive IOFB[2] Flash B. This bit determines the logic level the IOFB[2] output of the QuadSPI module is driven to in the inactive state. Refer to Driving Flash Control Signals in Single and Dual Mode. 0 IOFB[2] is driven to logic L 1 IOFB[2] is driven to logic H
17 ISD3FA	Idle Signal Drive IOFA[3] Flash A. This bit determines the logic level the IOFA[3] output of the QuadSPI module is driven to in the inactive state. Refer to Driving Flash Control Signals in Single and Dual Mode . 0 IOFA[3] is driven to logic L 1 IOFA[3] is driven to logic H
16 ISD2FA	Idle Signal Drive IOFA[2] Flash A. This bit determines the logic level the IOFA[2] output of the QuadSPI module is driven to in the inactive state. Refer to Driving Flash Control Signals in Single and Dual Mode . 0 IOFA[2] is driven to logic L 1 IOFA[2] is driven to logic H
15 DOZE	Doze Enable. The DOZE bit provides support for externally controlled Doze Mode power-saving mechanism. 0 A doze request will be ignored by the QuadSPI module 1 A doze request will be processed by the QuadSPI module
14 MDIS	Module Disable. The MDIS bit allows the clock to the non-memory mapped logic in the QuadSPI to be stopped, putting the QuadSPI in a software controlled power-saving state. 0 Enable QuadSPI clocks. 1 Allow external logic to disable QuadSPI clocks.
13–12 Reserved	This field is reserved.
11 CLR_TXF	Clear TX FIFO/Buffer. Invalidates the TX Buffer content. This is a self-clearing field. 0 No action. 1 Read and write pointers of the TX Buffer are reset to 0. QSPI_TBSR[TRCTR] is reset to 0.
10 CLR_RXF	Clear RX FIFO. Invalidates the RX Buffer. This is a self-clearing field. 0 No action. 1 Read and write pointers of the RX Buffer are reset to 0. QSPI_RBSR[RDBFL] is reset to 0.
9 Reserved	This field is reserved.
8 Reserved	This field is reserved.

Table continues on the next page...

QuadSPI_MCR field descriptions (continued)

Field	Description
7 DDR_EN	DDR mode enable 0 2x and 4x clocks are disabled for SDR instructions only 1 2x and 4x clocks are enabled supports both SDR and DDR instruction.
6 DQS_EN	DQS enable. This field is valid for both SDR and DDR mode. For more details, refer to Data Strobe (DQS) sampling method. 0 DQS disabled. 1 DQS enabled. When enabled, the incoming data is sampled on both the edges of DQS input when QSPI_MCR[DDR_EN] is set, else, on only one edge when QSPI_MCR[DDR_EN] is 0. The QSPI_SMPR[DDR_SMP] values are ignored.
5 Reserved	This field is reserved.
4 Reserved	This field is reserved.
3–2 END_CFG	Defines the endianness of the QuadSPI module. For more details refer to Byte Ordering Endianess
1 SWRSTHD	Software reset for AHB domain 0 No action 1 AHB domain flops are reset. Does not reset configuration registers. It is advisable to reset both the serial flash domain and AHB domain at the same time. Resetting only one domain might lead to side effects. **NOTE:** The software resets need the clock to be running to propagate to the design. The MCR[MDIS] should therefore be set to 0 when the software reset bits are asserted. Also, before they can be deasserted again (by setting MCR[SWRSTHD] to 0), it is recommended to set the MCR[MDIS] bit to 1. Once the software resets have been deasserted, the normal operation can be started by setting the MCR[MDIS] bit to 0.
0 SWRSTSD	Software reset for serial flash domain 0 No action 1 Serial Flash domain flops are reset. Does not reset configuration registers. It is advisable to reset both the serial flash domain and AHB domain at the same time. Resetting only one domain might lead to side effects. **NOTE:** The software resets need the clock to be running to propagate to the design. The MCR[MDIS] should therefore be set to 0 when the software reset bits are asserted. Also, before they can be deasserted again (by setting MCR[SWRSTSD] to 0), it is recommended to set the MCR[MDIS] bit to 1. Once the software resets have been deasserted, the normal operation can be started by setting the MCR[MDIS] bit to 0.

21.4.2.2 IP Configuration Register (QuadSPI_IPCR)

The IP configuration register provides all the configuration required for an IP initiated command. An IP command can be triggered by writing in the SEQID field of this register. If the SEQID field is written successfully, a new command to the external serial flash is started as per the sequence pointed to by the SEQID field. Refer to Normal Mode , for details about the command triggering and command execution.

Write:

- *QSPI_SR[IP_ACC]=0*

Address: 20C_0000h base + 8h offset = 20C_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved				SEQID				Reserved							PAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	IDATSZ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_IPCR field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–24 SEQID	Points to a sequence in the Look-up table. This field defines bits [6:2] of the LUT index. The bits [1:0] are always assumed to be 0. Refer to Look-up Table for more details. A write to this field triggers a transaction on the serial flash interface.
23–17 Reserved	This field is reserved.
16 PAR_EN	When set, a transaction to two serial flash devices is triggered in parallel mode. Refer to Parallel Flash Mode for more details.
IDATSZ	IP data transfer size. Defines the data transfer size in bytes of the IP command.

21.4.2.3 Flash Configuration Register (QuadSPI_FLSHCR)

The Flash configuration register contains the flash device specific timings that must be met by the QuadSPI controller for the device to function correctly.

Write:

- *QSPI_SR[AHB_ACC] = 0*
- *QSPI_SR[IP_ACC] = 0*

Address: 20C_0000h base + Ch offset = 20C_000Ch

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

QuadSPI_FLSHCR field descriptions

Field	Description
31–18 Reserved	This field is reserved.
17–16 TDH	Serial flash data in hold time. This helps in meeting the Data In Hold time requirement of a flash. This is valid only in DDR mode. **NOTE:** This field should be set to 0x00 in SDR mode (QuadSPI_MCR[DDR_EN]=0). Refer to Data input hold requirement of Flash for details. The valid TDH programming options are shown below. Other combinations are invalid. 00 Data aligned with the posedge of Internal reference clock of QuadSPI 01 Data aligned with 2x serial flash half clock 10 Data aligned with 4x serial flash half clock
15–12 Reserved	This field is reserved.
11–8 TCSH	Serial flash CS hold time in terms of serial flash clock cycles. Default value '3' should be used in case AHB prefetch size is less than or equal to 32-bytes DDR/16-bytes SDR. **NOTE:** The actual delay between chip select and clock is defined as: • TCSH = 1 SCK clk if N= 0/1 else, N SCK clk if N>1, where N is the setting of TCSH.
7–4 Reserved	This field is reserved. Reserved.
TCSS	Serial flash CS setup time in terms of serial flash clock cycles. **NOTE:** 1. The actual delay between chip select and clock is defined as: • TCSS = 0.5 SCK clk if N= 0/1 else, N+0.5 SCK clk if N>1, where N is the setting of TCSS. 2. Any update to the TCSS register bits is visible on the flash interface only from the second transaction following the update.

21.4.2.4 Buffer0 Configuration Register (QuadSPI_BUF0CR)

This register provides the configuration for any access to buffer0. An access is routed to buffer0 when the master port number of the incoming AHB request matches the MSTRID field of BUF0CR. Any buffer "miss" leads to a serial flash transaction being triggered as per the sequence pointed to the SEQID field. Buffer0 may also be configured as a high priority buffer by setting the HP_EN field of this register.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 10h offset = 20C_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HP_EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*

* Notes:
- MSTRID field: See the module configuration for reset value.

QuadSPI_BUF0CR field descriptions

Field	Description
31 HP_EN	High Priority Enable. When set, the master associated with this buffer is assigned a priority higher than the rest of the masters. An access by a high priority master will suspend any ongoing prefetch by another AHB master and will be serviced on high priority. Refer to Flexible AHB Buffers for details.
30–16 Reserved	This field is reserved.
15–8 ADATSZ	AHB data transfer size Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash. For example, a value of 0x2 will set transfer size to 16 bytes. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. Software should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
MSTRID	Master ID. The ID of the AHB master associated with BUFFER0. Any AHB access with this master port number is routed to this buffer. It must be ensured that the master IDs associated with all buffers must be different.

21.4.2.5 Buffer1 Configuration Register (QuadSPI_BUF1CR)

This register provides the configuration for any access to buffer1. An access is routed to buffer1 when the master port number of the incoming AHB request matches the MSTRID field of BUF1CR. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

Write:
- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 14h offset = 20C_0014h

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

* Notes:
- MSTRID field: See the module configuration for reset value.

QuadSPI_BUF1CR field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–8 ADATSZ	AHB data transfer size Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash. For example, a value of 0x2 will set transfer size to 16 bytes. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. Software should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
MSTRID	Master ID. The ID of the AHB master associated with BUFFER1. Any AHB access with this master port number is routed to this buffer. It must be ensured that the master IDs associated with all buffers must be different.

21.4.2.6 Buffer2 Configuration Register (QuadSPI_BUF2CR)

This register provides the configuration for any access to buffer2. An access is routed to buffer2 when the master port number of the incoming AHB request matches the MSTRID field of BUF2CR. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 18h offset = 20C_0018h

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

* Notes:

- MSTRID field: See the module configuration for the device specific reset value.

QuadSPI_BUF2CR field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved.
15–8 ADATSZ	AHB data transfer size Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash. For example, a value of 0x2 will set transfer size to 16 bytes. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. Software should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved. Reserved.

Table continues on the next page...

QuadSPI_BUF2CR field descriptions (continued)

Field	Description
MSTRID	Master ID. The ID of the AHB master associated with BUFFER2. Any AHB access with this master port number is routed to this buffer. It must be ensured that the master IDs associated with all buffers must be different.

21.4.2.7 Buffer3 Configuration Register (QuadSPI_BUF3CR)

This register provides the configuration for any access to buffer3. An access is routed to buffer3 when the master port number of the incoming AHB request matches the MSTRID field of BUF3CR. Any buffer "miss" leads to the buffer being flushed a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

In the case that the ALLMST field is not set, any such transaction (where master port number does not match any of the MSTRID fields) will be returned an ERROR response.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 1Ch offset = 20C_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	ALLMST	Reserved														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*

* Notes:

- MSTRID field: See the module configuration for the device specific reset value.

QuadSPI_BUF3CR field descriptions

Field	Description
31 ALLMST	All master enable. When set, buffer3 acts as an all-master buffer. Any AHB access with a master port number not matching with the master ID of buffer0 or buffer1 or buffer2 is routed to buffer3. When set, the MSTRID field of this register is ignored.
30–16 Reserved	This field is reserved. Reserved.

Table continues on the next page...

QuadSPI_BUF3CR field descriptions (continued)

Field	Description
15–8 ADATSZ	AHB data transfer size Defines the data transfer size in 8 Bytes of an AHB triggered access to serial flash. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. Software should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
MSTRID	Master ID. The ID of the AHB master associated with BUFFER3. Any AHB access with this master port number is routed to this buffer. It must be ensured that the master IDs associated with all buffers must be different.

21.4.2.8 Buffer Generic Configuration Register (QuadSPI_BFGENCR)

This register provides the generic configuration to any of the buffer accesses. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field. If the PAR_EN field is set, all the buffer accesses result in parallel accesses to the flashes.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 20h offset = 20C_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															PAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SEQID				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BFGENCR field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 PAR_EN	When set, a transaction to two serial flash devices is triggered in parallel mode. Refer to Parallel Flash Mode for more details.

Table continues on the next page...

QuadSPI_BFGENCR field descriptions (continued)

Field	Description
15–12 SEQID	Points to a sequence in the Look-up-table. The SEQID defines the bits [6:2] of the LUT index. The bits [1:0] are always assumed to be 0. Refer to Look-up Table. **NOTE:** If the sequence pointer differs between the new and previous sequence then the user should reset this. See QSPI_SPTRCLR for more information.
Reserved	This field is reserved.

21.4.2.9 SOC Configuration Register (QuadSPI_SOCCR)

This register is programmed at chip level for QuadSPI delay chain configuration. For details, refer to chip-specific QuadSPI information.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 24h offset = 20C_0024h

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

QuadSPI_SOCCR field descriptions

Field	Description
SOCCFG	SOC Configuration For details, refer to chip-specific QuadSPI information.

21.4.2.10 Buffer0 Top Index Register (QuadSPI_BUF0IND)

This register specifies the top index of buffer0, which defines its size. Note that that the 3 LSBs of this register are set to zero. This ensures that the buffer is 64-bit aligned, as each buffer entry is 64 bits long.

The register value should be set to the desired number of bytes. For example, setting BUF0IND[31:3] to 0 gives 0 bytes, 1 gives 8 bytes etc.

The size of buffer0 is the difference between BUF0IND and 0.

Software should ensure that BUF0IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 30h offset = 20C_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPINDX0																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF0IND field descriptions

Field	Description
31–3 TPINDX0	Top index of buffer 0.
Reserved	This field is reserved. Reserved.

21.4.2.11 Buffer1 Top Index Register (QuadSPI_BUF1IND)

This register specifies the top index of buffer1, which defines its size. Note that the 3 LSBs of this register are set to zero. This ensures that the buffer is 64-bit aligned as each buffer entry is 64 bits long.

The size of buffer1 is the difference between BUF1IND and BUF0IND. The register value should be entered in bytes. For example, If BUF0IND = 0x100 then setting BUF1IND = 0x130 will set buffer1 size to 0x30 bytes.

It is the responsibility of the software to ensure that BUF1IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 34h offset = 20C_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPINDX1																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF1IND field descriptions

Field	Description
31–3 TPINDX1	Top index of buffer 1.
Reserved	This field is reserved.

21.4.2.12 Buffer2 Top Index Register (QuadSPI_BUF2IND)

This register specifies the top index of buffer2, which defines its size. Note that that the 3 LSBs of this register are set to zero. This ensures that the buffer is 64-bit aligned as each buffer entry is 64 bits long.

The size of buffer2 is the difference between the BUF2IND and BUF1IND. The register value should be entered in bytes. For example, if BUF1IND = 0x130 then setting BUF2IND = 0x180 will set buffer2 size to 0x50 bytes.

It is the responsibility of the software to ensure that BUF2IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 38h offset = 20C_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPINDX2																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF2IND field descriptions

Field	Description
31–3 TPINDX2	Top index of buffer 2.
Reserved	This field is reserved.

21.4.2.13 Serial Flash Address Register (QuadSPI_SFAR)

The module automatically translates this address on the memory map to the address on the flash itself. When operating in 24-bit mode, only bits 23-0 are sent to the flash. In 32-bit mode, bits 27-0 are used with bits 31-28 driven to 0 Refer to Table 21-7 for the mapping between the access mode and the QSPI_SFAR content and to Normal Mode for details about the command triggering and command execution. The software should ensure that the serial flash address provided in the QSPI_SFAR register lies in the valid flash address range as defined in Table 21-7.

Write:

- *QSPI_SR[IP_ACC] = 0*

Address: 20C_0000h base + 100h offset = 20C_0100h

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

QuadSPI_SFAR field descriptions

Field	Description
SFADR	Serial Flash Address. The register content is used as byte address for all following IP Commands.

21.4.2.14 Sampling Register (QuadSPI_SMPR)

The Sampling Register allows configuration of how the incoming data from the external serial flash devices are sampled in the QuadSPI module.

Write: Disabled Mode

Address: 20C_0000h base + 108h offset = 20C_0108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													DDRSMP		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FSDLY	FSPHS	0		HSDLY	HSPHS	HSENA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_SMPR field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 DDRSMP	DDR Sampling point Select the sampling point for incoming data when serial flash is executing a DDR instruction. Refer to Figure 21-11 for details on the sampling points.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FSDLY	Full Speed Delay selection for SDR instructions. Select the delay with respect to the reference edge for the sample point valid for full speed commands.

Table continues on the next page...

QuadSPI_SMPR field descriptions (continued)

Field	Description
	0 One clock cycle delay 1 Two clock cycles delay. **NOTE:** Please refer to Supported read modes for its usage. It is ignored when using non-DQS DDR instructions.
5 FSPHS	Full Speed Phase selection for SDR instructions. Select the edge of the sampling clock valid for full speed commands. 0 Select sampling at non-inverted clock 1 Select sampling at inverted clock. **NOTE:** Please refer to Supported read modes for its usage. FSPHS programming is only supported with internal non-DQS SDR sampling. It must be '0' in other sampling modes.
4–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 HSDLY	Half Speed Delay selection for SDR instructions. Only relevant when HSENA bit is set. Select the delay with respect to the reference edge for the sample point valid for half speed commands. 0 One clock cycle delay 1 Two clock cycle delay
1 HSPHS	Half Speed Phase selection for SDR instructions. Only relevant when HSENA bit is set. Select the delay with respect to the reference edge for the sample point valid for half speed commands. 0 Select sampling at non-inverted clock 1 Select sampling at inverted clock
0 HSENA	Half Speed serial flash clock Enable This bit enables the divide by 2 of the clock to the external serial flash device for all commands, only in SDR. Refer to Serial Flash Clock Frequency Limitations for details. 0 Disable divide by 2 of serial flash clock for half speed commands 1 Enable divide by 2 of serial flash clock for half speed commands

21.4.2.15 RX Buffer Status Register (QuadSPI_RBSR)

This register contains information related to the receive data buffer.

Address: 20C_0000h base + 10Ch offset = 20C_010Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDCTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		RDBFL						Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_RBSR field descriptions

Field	Description
31–16 RDCTR	Read Counter. Indicates how many entries of 4 bytes have been removed from the RX Buffer. For example, a value of 0x2 would indicate 8 bytes have been removed. It is incremented by the number (QSPI_RBCT[WMRK] + 1) on RX Buffer POP event. The RX Buffer can be popped using DMA or pop flag QSPI_FR[RBDF]. The QSPI_RSER[RBDDE] defines which pop has to be done. For further details, refer to AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31) and "Data Transfer from the QuadSPI Module Internal Buffers" section in Flash Read.
15–14 Reserved	This field is reserved.
13–8 RDBFL	RX Buffer Fill Level. Indicates how many entries of 4 bytes are still available in the RX Buffer. For example, a value of 0x2 would indicate 8 bytes are available.
Reserved	This field is reserved.

21.4.2.16 RX Buffer Control Register (QuadSPI_RBCT)

This register contains control data related to the receive data buffer.

Write:

- *QSPI_SR[IP_ACC] = 0*

Address: 20C_0000h base + 110h offset = 20C_0110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							RXBRD	Reserved			WMRK				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_RBCT field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 RXBRD	RX Buffer Readout. This field specifies the access scheme for the RX Buffer readout. 0 RX Buffer content is read using the AHB Bus registers QSPI_ARDB0 to QSPI_ARDB31. For details, refer to Exclusive Access to Serial Flash for AHB Commands. 1 RX Buffer content is read using the IP Bus registers QSPI_RBDR0 to QSPI_RBDR31.
7–5 Reserved	This field is reserved.
WMRK	RX Buffer Watermark. This field determines when the readout action of the RX Buffer is triggered. When the number of valid entries in the RX Buffer is equal to or greater than the number given by (WMRK+1) the QSPI_SR[RXWE] flag is asserted.The value should be entered as the number of 4-byte entries minus 1. For example, a value of 0x0 would set the watermark to 4 bytes, 1 to 8 bytes, 2 to 12 bytes, and so on. For details, refer to DMA Usage.

21.4.2.17 TX Buffer Status Register (QuadSPI_TBSR)

This register contains information related to the transmit data buffer.

Address: 20C_0000h base + 150h offset = 20C_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TRCTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		TRBFL						Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_TBSR field descriptions

Field	Description
31–16 TRCTR	Transmit Counter. This field indicates how many entries of 4 bytes have been written into the TX Buffer by host accesses. It is reset to 0 when a 1 is written to QSPI_MCR[CLR_TXF]. It is incremented on each write access to the QSPI_TBDR register when another word has been pushed onto the TX Buffer. When it is not cleared the TRCTR field wraps around to 0. Refer to TX Buffer Data Register (QuadSPI_TBDR) for details.
15–14 Reserved	This field is reserved.
13–8 TRBFL	TX Buffer Fill Level. The TRBFL field contains the number of entries of 4 bytes each available in the TX Buffer for the QuadSPI module to transmit to the serial flash device.
Reserved	This field is reserved.

21.4.2.18 TX Buffer Data Register (QuadSPI_TBDR)

The QSPI_TBDR register provides access to the circular TX Buffer of depth 128 bytes. This buffer provides the data written into it as write data for the page programming commands to the serial flash device. Refer to Table 21-16 for the byte ordering scheme. A write transaction on the flash with data size of less than 32 bits will lead to the removal of four data entry from the TX buffer. The valid bits will be used and the rest of the bits will be discarded.

Write:

- *QSPI_SR[TXFULL] = 0*

32-bit write access required

Address: 20C_0000h base + 154h offset = 20C_0154h

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

QuadSPI_TBDR field descriptions

Field	Description
TXDATA	TX Data On write access the data is written into the next available entry of the TX Buffer and the QPSI_TBSR[TRBFL] field is updated accordingly. On a read access, the last data written to the register is returned.

21.4.2.19 Status Register (QuadSPI_SR)

The QSPI_SR register provides all available status information about SFM command execution and arbitration, the RX Buffer, TX Buffer, and the AHB Buffer.

Address: 20C_0000h base + 15Ch offset = 20C_015Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPSMP			Reserved	TXFULL	Reserved	Reserved	TXEDA	RXDMA	Reserved			RXFULL	Reserved		RXWE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	AHB3FUL	AHB2FUL	AHB1FUL	AHB0FUL	AHB3NE	AHB2NE	AHB1NE	AHB0NE	AHBTRN	AHBGNT	Reserved	Reserved	AHB_ACC	IP_ACC	BUSY
W																
Reset	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_SR field descriptions

Field	Description
31–29 DLPSMP	Data learning pattern sampling point. The sampling point found by the controller with the data learning pattern. • This is used for DDR only. • If the learning fails, this field will return garbage and DLPFF bit will be set. • In case of Data learning with DQS this field will return the reset value as sampling point match is not found in case of DQS. For details, refer to Data Learning.
28 Reserved	This field is reserved.
27 TXFULL	TX Buffer Full. Asserted when no more data can be stored.
26 Reserved	This field is reserved.
25 Reserved	This field is reserved.
24 TXEDA	Tx Buffer Enough Data Available Asserted when TX Buffer contains enough data for any pop operation to take place. There must be at least 128 bit data available in TX FIFO for any pop operation; otherwise, QSPI_FR[TBUF] will be set.

Table continues on the next page...

QuadSPI_SR field descriptions (continued)

Field	Description
23 RXDMA	RX Buffer DMA. Asserted when RX Buffer read out via DMA is active i.e DMA is requested or running.
22–20 Reserved	This field is reserved.
19 RXFULL	RX Buffer Full. Asserted when the RX Buffer is full, i.e. that QSPI_RBSR[RDBFL] field is equal to 32.
18–17 Reserved	This field is reserved.
16 RXWE	RX Buffer Watermark Exceeded. Asserted when the number of valid entries in the RX Buffer exceeds the number given in the QSPI_RBCT[WMRK] field.
15 Reserved	This field is reserved.
14 AHB3FUL	AHB 3 Buffer Full. Asserted when AHB 3 buffer is full.
13 AHB2FUL	AHB 2 Buffer Full. Asserted when AHB 2 buffer is full.
12 AHB1FUL	AHB 1 Buffer Full. Asserted when AHB 1 buffer is full.
11 AHB0FUL	AHB 0 Buffer Full. Asserted when AHB 0 buffer is full.
10 AHB3NE	AHB 3 Buffer Not Empty. Asserted when AHB 3 buffer contains data.
9 AHB2NE	AHB 2 Buffer Not Empty. Asserted when AHB 2 buffer contains data.
8 AHB1NE	AHB 1 Buffer Not Empty. Asserted when AHB 1 buffer contains data.
7 AHB0NE	AHB 0 Buffer Not Empty. Asserted when AHB 0 buffer contains data.
6 AHBTRN	AHB Access Transaction pending. Asserted when there is a pending request on the AHB interface. Refer to the AMBA specification for details.
5 AHBGNT	AHB Command priority Granted: Asserted when another module has been granted priority of AHB Commands against IP Commands. For details refer to Command Arbitration.
4 Reserved	This field is reserved.
3 RESERVED	This field is reserved.
2 AHB_ACC	AHB Access. Asserted when the transaction currently executed was initiated by AHB bus.
1 IP_ACC	IP Access. Asserted when transaction currently executed was initiated by IP bus.
0 BUSY	Module Busy. Asserted when module is currently busy handling a transaction to an external flash device.

21.4.2.20 Flag Register (QuadSPI_FR)

The QSPI_FR register provides all available flags about SFM command execution and arbitration which may serve as source for the generation of interrupt service requests. Note that the error flags in this register do not relate directly to the execution of the transaction in the serial flash device itself but only to the behavior and conditions visible in the QuadSPI module.

Write: Enabled Mode

Address: 20C_0000h base + 160h offset = 20C_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPFF	Reserved		Reserved	TBFF	TBUF	Reserved		ILLINE	Reserved					RBOF	RBDF
W	w1c				w1c	w1c			w1c						w1c	w1c
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ABSEF	Reserved	Reserved	ABOF	IUEF	Reserved			IPAEF	IPIEF	Reserved	IPGEF	Reserved			TFF
W	w1c			w1c	w1c				w1c	w1c		w1c				w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_FR field descriptions

Field	Description
31 DLPFF	Data Learning Pattern Failure Flag. Set when DATA_LEARN instruction was encountered in a sequence but no sampling point was found for the data learning patternin case only 8 bit data learning is requested for non DQS mode. The controller automatically starts sampling using the value in QSPI_SMPR[DDRSMP].If more than 8 bits data learning are requested with QSPI_MCR[DQS_EN] set to 0, and the sampling point found after first 8 bit match doesn't remain the same for the whole instruction duration, this flag is set. In case of Data learn with DQS this flag is set whenever the incoming data from flash on DQS edges doesn't match the pattern in QSPI_DLPR. For details, refer to Data Learning.
30–29 RESERVED	This field is reserved.
28 Reserved	This field is reserved.
27 TBFF	TX Buffer Fill Flag. Before writing to the TX buffer, this bit should be cleared. Then this bit has to be read back. If the bit is set, the TX Buffer can take more data. If the bit remains cleared, the TX buffer is full. Refer to Tx Buffer Operation for details.
26 TBUF	TX Buffer Underrun Flag. Set when the module tried to pull data although TX Buffer was emptyor the buffer contains less than 128 bits of data. The application must ensure that the buffer never goes empty during a transaction except for the last data fetch. The IP Command leading to the TX Buffer underrun is continued (data sent to the serial flash device is all F in case of valid TX underrun). Here valid 'underrun' means that, it should have occurred during the transacting such that few bytes (i.e., less than 4 bytes) are left in FIFO and remaining are filled with “FFFFh". Software should start TX transaction only when 128-bits are written in TX buffer. The application must clear the TX Buffer in response to this event by writing a '1' to the QSPI_MCR[CLR_TXF].
25–24 Reserved	This field is reserved.
23 ILLINE	Illegal Instruction Error Flag. Set when an illegal instruction is encountered by the controller in any of the sequences. As soon as this flag is set, user should assert QSPI_MCR[SWRSTSD], QSPI_MCR[SWRSTHD] and after software resets de-assertion in normal mode, QSPI_MCR[CLR_TXF] and QSPI_MCR[CLR_RXF] should be set to clear any remaining data in buffers i.e., reset to flash and

Table continues on the next page...

QuadSPI_FR field descriptions (continued)

Field	Description
	AHB domain after re-configuring the correct sequence instruction. Refer to Table 21-14 for a list of legal instructions.
22–18 Reserved	This field is reserved.
17 RBOF	RX Buffer Overflow Flag. Set when not all the data read from the serial flash device could be pushed into the RX Buffer. The IP Command leading to this condition is continued until the number of bytes according to the QSPI_IPCR[IDATSZ] field has been read from the serial flash device. The content of the RX Buffer is not changed.
16 RBDF	RX Buffer Drain Flag. Will be set if the QuadSPI_SR[RXWE] status bit is asserted. Writing 1 into this bit triggers one of the following actions: • If the RX Buffer has up to QuadSPI_RBCT[WMRK] valid entries then the flag is cleared. • If the RX Buffer has more than QuadSPI_RBCT[WMRK] valid entries and the QuadSPI_RSER[RBDDE] bit is not set (flag driven mode) a RX Buffer POP event is triggered. The flag remains set if the RX Buffer contains more than QuadSPI_RBCT[WMRK] valid entries after the RX Buffer POP event is finished. The flag is cleared if the RX Buffer contains less than or equal to QuadSPI_RBCT[WMRK] valid entries after the RX Buffer POP event is finished. Refer to "Receive Buffer Drain Interrupt or DMA Request" section in Normal Mode Interrupt and DMA Requests, for details.
15 ABSEF	AHB Sequence Error Flag. Set when the execution of an AHB Command is started with a WRITE or WRITE_DDR Command in the sequence pointed to by the QSPI_BUFxCR register. (QSPI_BUFxCR implies any one of QSPI_BUF0CR/QSPI_BUF1CR/QSPI_BUF2CR/QSPI_BUF3CR.) Communication with the serial flash device is terminated before the execution of WRITE/WRITE_DDR command by the QuadSPI module. The AHB bus request which triggered this command is answered with an ERROR response.
14 Reserved	Reserved This field is reserved.
13 Reserved	Reserved This field is reserved.
12 ABOF	AHB Buffer Overflow Flag. Set when the size of the AHB access exceeds the size of the AHB buffer. This condition can occur only if the QSPI_BUFxCR[ADATSZ] field is programmed incorrectly. The AHB Command leading to this condition is continued until the number of entries according to the QSPI_BUFxCR[ADATSZ] field has been read from the serial flash device. The content of the AHB Buffer is not changed.
11 IUEF	IP Command Usage Error Flag. Set when in parallel flash mode the execution of an IP Command is started and the sequence pointed to by the sequence ID contains a WRITE or a WRITE_DDR command. Refer to Table 21-14 table for the related commands. Communication with the serial flash device is terminated before the execution of WRITE/WRITE_DDR command by the QuadSPI module.
10–8 Reserved	This field is reserved.

Table continues on the next page...

QuadSPI_FR field descriptions (continued)

Field	Description
7 IPAEF	IP Command Trigger during AHB Access Error Flag. Set when the following condition occurs: • A write access occurs to the QSPI_IPCR[SEQID] field and the QSPI_SR[AHB_ACC] bit is set. Any command leading to the assertion of the IPAEF flag is ignored.
6 IPIEF	IP Command Trigger could not be executed Error Flag. Set when the QSPI_SR[IP_ACC] bit is set (i.e. an IP triggered command is currently executing) and any of the following conditions occurs: • Write access to the QSPI_IPCR register. Any command leading to the assertion of the IPIEF flag is ignored • Write access to the QSPI_SFAR register. • Write access to the QSPI_RBCT register.
5 Reserved	This field is reserved.
4 IPGEF	IP Command Trigger during AHB Grant Error Flag. Set when the following condition occurs: • A write access occurs to the QSPI_IPCR[SEQID] field and the QSPI_SR[AHBGNT] bit is set. Any command leading to the assertion of the IPGEF flag is ignored.
3–1 Reserved	This field is reserved.
0 TFF	IP Command Transaction Finished Flag. Set when the QuadSPI module has finished a running IP Command. If an error occurred the related error flags are valid, at the latest, in the same clock cycle when the TFF flag is asserted.

21.4.2.21 Interrupt and DMA Request Select and Enable Register (QuadSPI_RSER)

The QuadSPI_RSER register provides enables and selectors for the interrupts in the QuadSPI module.

NOTE

Each flag of the QuadSPI_FR register enabled as source for an interrupt prevents the QuadSPI module from entering Stop Mode or Module Disable Mode when this flag is set.

Write: Anytime

Address: 20C_0000h base + 164h offset = 20C_0164h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLPFIE	Reserved		Reserved	TBFIE	TBUIE	TBFDE	Reserved	ILLINIE	Reserved	RBDDE	Reserved			RBOIE	RBDIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ABSEIE	Reserved	Reserved	ABOIE	IUEIE	Reserved			IPAEIE	IPIEIE	Reserved	IPGEIE	Reserved			TFIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_RSER field descriptions

Field	Description
31 DLPFIE	Data Learning Pattern Failure Interrupt enable . Triggered by DLPFF flag in QSPI_FR register 0 No DLPFF interrupt will be generated 1 DLPFF interrupt will be generated
30–29 Reserved	This field is reserved.
28 Reserved	This field is reserved.
27 TBFIE	TX Buffer Fill Interrupt Enable 0 No TBFF interrupt will be generated 1 TBFF interrupt will be generated
26 TBUIE	TX Buffer Underrun Interrupt Enable 0 No TBUF interrupt will be generated 1 TBUF interrupt will be generated
25 TBFDE	TX Buffer Fill DMA Enable Enables generation of DMA requests for TX Buffer fill. When this is set, DMA requests are generated as long as the QuadSPI_SR[TXWA] status bit is set. **NOTE:** • Once set, writting a zero does not impact DMA transfers, even though the bit gets written a zero. • After QuadSPI_MCR[SWRSTHD] is used to do software reset for AHB Domain, this bit takes effect and needs to be written a 1 to re-enable DMA transfers. 0 No DMA request will be generated 1 DMA request will be generated
24 Reserved	This field is reserved.
23 ILLINIE	Illegal Instruction Error Interrupt Enable. Triggered by ILLINE flag in QSPI_FR 0 No ILLINE interrupt will be generated 1 ILLINE interrupt will be generated
22 Reserved	This field is reserved.
21 RBDDE	RX Buffer Drain DMA Enable: Enables generation of DMA requests for RX Buffer Drain. When this bit is set DMA requests are generated as long as the QSPI_SR[RXWE] status bit is set.

Table continues on the next page...

QuadSPI_RSER field descriptions (continued)

Field	Description
	0 No DMA request will be generated 1 DMA request will be generated
20–18 Reserved	This field is reserved.
17 RBOIE	RX Buffer Overflow Interrupt Enable 0 No RBOF interrupt will be generated 1 RBOF interrupt will be generated
16 RBDIE	RX Buffer Drain Interrupt Enable: Enables generation of IRQ requests for RX Buffer Drain. When this bit is set the interrupt is asserted as long as the QuadSPI_SR[RBDF] flag is set. 0 No RBDF interrupt will be generated 1 RBDF Interrupt will be generated
15 ABSEIE	AHB Sequence Error Interrupt Enable: Triggered by ABSEF flags of QSPI_FR 0 No ABSEF interrupt will be generated 1 ABSEF interrupt will be generated
14 Reserved	Reserved This field is reserved.
13 Reserved	Reserved This field is reserved.
12 ABOIE	AHB Buffer Overflow Interrupt Enable 0 No ABOF interrupt will be generated 1 ABOF interrupt will be generated
11 IUEIE	IP Command Usage Error Interrupt Enable 0 No IUEF interrupt will be generated 1 IUEF interrupt will be generated
10–8 Reserved	This field is reserved.
7 IPAEIE	IP Command Trigger during AHB Access Error Interrupt Enable 0 No IPAEF interrupt will be generated 1 IPAEF interrupt will be generated
6 IPIEIE	IP Command Trigger during IP Access Error Interrupt Enable 0 No IPIEF interrupt will be generated 1 IPIEF interrupt will be generated
5 Reserved	This field is reserved.
4 IPGEIE	IP Command Trigger during AHB Grant Error Interrupt Enable 0 No IPGEF interrupt will be generated 1 IPGEF interrupt will be generated
3–1 Reserved	This field is reserved. Reserved.

Table continues on the next page...

QuadSPI_RSER field descriptions (continued)

Field	Description
0 TFIE	Transaction Finished Interrupt Enable 0 No TFF interrupt will be generated 1 TFF interrupt will be generated

21.4.2.22 Sequence Suspend Status Register (QuadSPI_SPNDST)

The sequence suspend status register provides information specific to any suspended sequence. An AHB sequence may be suspended when a high priority AHB master makes an access before the AHB sequence completes the data transfer requested.

Address: 20C_0000h base + 168h offset = 20C_0168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_SPNDST field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–9 DATLFT	Data left: Provides information about the amount of data left to be read in the suspended sequence. Valid only when SUSPND is set to 1'b1. Value in terms of 64 bits or 8 bytes
8 Reserved	This field is reserved.
7–6 SPDBUF	Suspended Buffer: Provides the suspended buffer number. Valid only when SUSPND is set to 1'b1
5–1 Reserved	This field is reserved.
0 SUSPND	When set, it signifies that a sequence is in suspended state

21.4.2.23 Sequence Pointer Clear Register (QuadSPI_SPTRCLR)

The sequence pointer clear register provides bits to reset the IP and Buffer sequence pointers. The sequence pointer contains the index of which instruction within the LUT entry is to be executed next. For example, if the LUT entry ends on a JMP_ON_CS value of 2, the index will be stored as 2.

The software should reset the sequence pointers whenever the sequence ID is changed by updating the SEQID field in QSPI_IPCR or QSPI_BFGENCR.

Address: 20C_0000h base + 16Ch offset = 20C_016Ch

QuadSPI_SPTRCLR field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 IPPTRC	IP Pointer Clear: 1: Clears the sequence pointer for IP accesses as defined in QuadSPI_IPCR This is a self-clearing field. This bit should be programmed two times to clear the sequence pointers.
7–1 Reserved	This field is reserved. Reserved.
0 BFPTRC	Buffer Pointer Clear: 1: Clears the sequence pointer for AHB accesses as defined in QuadSPI_BFGENCR. This is a self-clearing field. This bit should be programmed two times to clear the sequence pointers.

21.4.2.24 Serial Flash A1 Top Address (QuadSPI_SFA1AD)

The QSPI_SFA1AD register provides the address mapping for the serial flash A1.The difference between QSPI_SFA1AD[TPADA1] and QSPI_AMBA_BASE defines the size of the memory map for serial flash A1.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 180h offset = 20C_0180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADA1																						Reserved									
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0

* Notes:

- TPADA1 field: See the module configuration for the device specific reset values.

QuadSPI_SFA1AD field descriptions

Field	Description
31–10 TPADA1	Top address for Serial Flash A1. In effect, TPADxx is the first location of the next memory.
Reserved	This field is reserved.

21.4.2.25 Serial Flash A2 Top Address (QuadSPI_SFA2AD)

The QSPI_SFA2AD register provides the address mapping for the serial flash A2.The difference between QSPI_SFA2AD[TPADA2] and QSPI_SFA1AD[TPADA1] defines the size of the memory map for serial flash A2.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 184h offset = 20C_0184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADA2																						Reserved									
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0

* Notes:

- TPADA2 field: See the module configuration for the device specific reset values.

QuadSPI_SFA2AD field descriptions

Field	Description
31–10 TPADA2	Top address for Serial Flash A2. In effect, TPxxAD is the first location of the next memory.
Reserved	This field is reserved.

21.4.2.26 Serial Flash B1 Top Address (QuadSPI_SFB1AD)

The QSPI_SFB1AD register provides the address mapping for the serial flash B1.The difference between QSPI_SFB1AD[TPADB1] and QSPI_SFA2AD[TPADA2] defines the size of the memory map for serial flash B1.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 188h offset = 20C_0188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADB1																						Reserved									
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0

* Notes:

- TPADB1 field: See the module configuration for the device specific reset values.

QuadSPI_SFB1AD field descriptions

Field	Description
31–10 TPADB1	Top address for Serial Flash B1.In effect, TPxxAD is the first location of the next memory.
Reserved	This field is reserved.

21.4.2.27 Serial Flash B2 Top Address (QuadSPI_SFB2AD)

The QSPI_SFB2AD register provides the address mapping for the serial flash B2.The difference between QSPI_SFB2AD[TPADB2] and QSPI_SFB1AD[TPADB1] defines the size of the memory map for serial flash B2.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 18Ch offset = 20C_018Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADB2																						Reserved									
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0

* Notes:
- TPADB2 field: See the module configuration for the device specific reset values.

QuadSPI_SFB2AD field descriptions

Field	Description
31–10 TPADB2	Top address for Serial Flash B2. In effect, TPxxAD is the first location of the next memory.
Reserved	This field is reserved.

21.4.2.28 Data Learn Pattern Register (QuadSPI_DLPR)

The QSPI_DLPR register contains the information of the data to be used for Data Learning.

Write:
- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 20C_0000h base + 190h offset = 20C_0190h

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

QuadSPI_DLPR field descriptions

Field	Description
DLPV	Data Learning Pattern Value: This value is used for data learning in DDR and DQS mode. If programmer wants to do data learn for more than 32 bit than the same value in the register is repeated. Say if 64 bit data learning is requested by any flash and the value of DLPR is aa55_3443 then the 64 bit value will be aa55_3443_aa55_3443. If 8 bit data learning was enabled by programming in seq_operand fields of DATA_LEARN instruction to 1 , the bits [7:0] are used as data_learning pattern. For details, refer to Data Learning

21.4.2.29 RX Buffer Data Register (QuadSPI_RBDR*n*)

The QuadSPI_RBDR registers provide access to the individual entries in the RX Buffer. Refer to Table 21-16 for the byte ordering scheme.

QuadSPI_RBDR0 corresponds to the actual position of the read pointer within the RX Buffer. The number of valid entries available depends from the number of RX Buffer entries implemented and from the number of valid buffer entries available in the RX Buffer.

Example 1, RX Buffer filled completely with 32 words: In this case the address range for valid read access extends from QuadSPI_RBDR0 to QuadSPI_RBDR31.

Example 2, RX Buffer filled with 5 valid words: RX Buffer fill level QuadSPI_RBSR[RDBFL] is 5. In this case an access to QuadSPI_RBDR4 provides the last valid entry.

Any access beyond the range of valid RX Buffer entries provides undefined results.

Address: 20C_0000h base + 200h offset + (4d × i), where i=0d to 31d

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

QuadSPI_RBDR*n* field descriptions

Field	Description
RXDATA	RX Data. The RXDATA field contains the data associated with the related RX Buffer entry. Data format and byte ordering is given in Byte Ordering of Serial Flash Read Data .

21.4.2.30 LUT Key Register (QuadSPI_LUTKEY)

The LUT Key register contains the key to lock and unlock the Look-up-table. Refer to Look-up Table for details.

Write: Anytime

Address: 20C_0000h base + 300h offset = 20C_0300h

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

QuadSPI_LUTKEY field descriptions

Field	Description
KEY	The key to lock or unlock the LUT. The KEY is 0x5AF05AF0. The read value is always 0x5AF05AF0

21.4.2.31 LUT Lock Configuration Register (QuadSPI_LCKCR)

The LUT lock configuration register is used along with QSPI_LUTKEY register to lock or unlock the LUT. This register has to be written immediately after QSPI_LUTKEY register for the lock or unlock operation to be successful. Refer to Look-up Table for details. Setting both the LOCK and UNLOCK bits as "00" or "11" is not allowed.

Write: Just after writing the LUT Key Register

(QSPI_LUTKEY)

Address: 20C_0000h base + 304h offset = 20C_0304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved														UNLOCK	LOCK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

QuadSPI_LCKCR field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1 UNLOCK	Unlocks the LUT when the following two conditions are met: 1. This register is written just after the LUT Key Register (QuadSPI_LUTKEY) 2. The LUT key register was written with 0x5AF05AF0 key
0 LOCK	Locks the LUT when the following condition is met: 1. This register is written just after the LUT Key Register (QuadSPI_LUTKEY) 2. The LUT key register was written with 0x5AF05AF0 key

21.4.2.32 Look-up Table register (QuadSPI_LUT*n*)

A sequence of instruction-operand pairs may be pre-populated in the LUT according to the device connected on board. Each instruction-operand pair is of 16 bits (2 bytes) each. Every sequence pre-programmed by Program Sequence Engine in the LUT is referred to by its index. The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. Reset value of the register shown below is only applicable to LUT2 to LUT63. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

NOTE

The reset values for LUT0 and LUT1 are 0818_0403h and 2400_1C08h, respectively.

Write: Once the LUT is unlocked

Address: 20C_0000h base + 310h offset + (4d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- The reset values for LUT0 and LUT1 are 0818_0403h and 2400_1C08h respectively.

QuadSPI_LUT*n* field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 8 Pads

Table continues on the next page...

QuadSPI_LUT*n* field descriptions (continued)

Field	Description
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 8 Pads
OPRND0	Operand for INSTR0.

21.4.3 Serial Flash Address Assignment

The serial flash address assignment may be modified by writing into Serial Flash A1 Top Address (QuadSPI_SFA1AD) and Serial Flash A2 Top Address (QuadSPI_SFA2AD) for device A and into Serial Flash B1 Top Address (QuadSPI_SFB1AD) and Serial Flash B2 Top Address (QuadSPI_SFB2AD) for device B. The following table shows how different access modes are related to the address specified for the next SFM Command. Note that this address assignment is valid for both IP and AHB commands.

Table 21-7. Serial Flash Address Assignment

Parameter	Function	Access Mode
QSPI_AMBA_BASE ((31:10) - 22 bits)	QuadSPI AHB base address	
TOP_ADDR_MEMA1(TPADA1)	Top address for the external flash A1 (first device of the dual die flash A, or the first of the two independent flashes sharing the IOFA)	Any access to the address space between TOP_ADDR_MEMA1 and QSPI_AMBA_BASE will be routed to **Serial Flash A1**
TOP_ADDR_MEMA2(TPADA2)	Top address for the external flash A2 (second device of the dual die flash A, or the second of the two independent flashes sharing the IOFA).	Any access to the address space between TOP_ADDR_MEMA2 and TOP_ADDR_MEMA1 will be routed to **Serial Flash A2**
TOP_ADDR_MEMB1(TPADB1)	Top address for the external flash B1 (first device of the dual die flash B, or the first of the two independent flashes sharing the IOFB)	Any access to the address space between TOP_ADDR_MEMB1 and TOP_ADDR_MEMA2 will be routed to **Serial Flash B1**
TOP_ADDR_MEMB2(TPADB2)	Top address for the external flash B2 (second device of the dual die flash B or the second of the two independent flashes sharing the IOFB)	Any access to the address space between TOP_ADDR_MEMB2 and TOP_ADDR_MEMB1 will be routed to **Serial Flash A2**

21.5 Flash memory mapped AMBA bus

QSPI_AMBA_BASE defines the address to be used as start address of the serial flash device as defined by the system memory map. Note that this may be a remapping of the physical address of the serial flash in the system. Refer to the system address map. Note that this may be a remapping of the physical address of the serial flash in the system. Refer to the system address map.

Table 21-8. QuadSPI AMBA Bus Memory Map

Address	Register Name
Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A	
QSPI_AMBA_BASE to (TOP_ADDR_MEMA2 - 0x01)	Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A Refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A for details and to Table 21-16 and Table 21-20 for information about the byte ordering.
Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B	
TOP_ADDR_MEMA2 to (TOP_ADDR_MEMB2 - 0x01)	Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B Refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B for details and to Table 21-16 and Table 21-20 for information about the byte ordering.
Parallel Flash Mode	
QSPI_AMBA_BASE to (TOP_ADDR_MEMB2 - 0x01)	Parallel Flash Mode Refer to Parallel Flash Mode for details and to Table 21-19 and Table 21-20 for information about the byte ordering.
AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)	
QSPI_ARDB_BASE to… (32 * 4 Byte) QSPI_ARDB_BASE + 0x0000_01FF	AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31) Refer to Table 21-16 and Table 21-18 for information about the byte ordering.

Note

Any read access to non-implemented addresses will provide undefined results.

In case of single-die flash devices, TOP_ADDR_MEMA2 and TOP_ADDR_MEMB2 should be initialized/programmed to TOP_ADDR_MEMA1 and TOP_ADDR_MEMB1 respectively (in effect, setting the size of these devices to 0). This would ensure that the complete memory map is assigned to only one flash device.

Parallel Flash Mode is valid only for commands related to data read from the serial flash. The first device of flash A has to be paired with the first device of flash B and the second device of

flash A has to be paired with the second device of flash B in parallel mode. Parallel mode is selected via the QSPI_BFGENCR[PAR_EN] bit for all masters in AHB driven mode and via the QSPI_IPCR[PAR_EN] in IP driven mode. In parallel mode, the incoming address (SFAR address in case of IP initiated transactions and the incoming AHB address in case of AHB initiated transactions) is divided by 2 and sent to the two flashes connected in parallel.

Any IP Command other than data read in Parallel Flash Mode will result in the assertion of the QSPI_FR[IUEF] flag and any AHB Command other than data read in Parallel Flash Mode will result in the assertion of the QSPI_FR[ABSEF] flag.

In 'Individual Flash Modes', the 3/4 address bytes (as programmed in the instruction/operand in the sequence) available for the flash address is determined by SFADR [23:0] or SFADR [31:0]as given in the table above.

In 'Parallel Flash Mode', both flashes are read with the same starting address of 3/4 (as programmed in the instruction/ operand in the sequence) bytes in size. This address is derived from SFADR [24:1] or SFADR [31:1] as given in the table above. The LSB of the SFADR field is used to select the appropriate bits of both flash devices to combine the byte corresponding to the selected address.

21.5.1 AHB Bus Access Considerations

It has to be noted that all logic in the QuadSPI module implementing the AHB Bus access is designed to read the content of an external serial flash device. Therefore, the following restrictions apply to the QuadSPI module with respect to accesses to the AHB bus.

- At present, the QuadSPI does not support AHB writes so any write access is answered with the ERROR condition according to the AMBA AHB Specification. No write occurs.
- Any AHB Command resulting in the assertion of the QSPI_FR[ABSEF] flag is answered with the ERROR condition according to the AMBA_AHB specification. The resulting AHB Command is ignored.

- AHB Bus access types fully supported are NONSEQ and BUSY.
- AHB access type SEQ is treated in the same way like NONSEQ. Refer to the AMBA AHB Specification for further details.

21.5.2 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A

Starting with address QSPI_AMBA_BASE the content of the first external serial flash devices is mapped into the address space of the device containing the QuadSPI module. Serial flash address byte address 0x0 corresponds to bus address QSPI_AMBA_BASE with increasing order. Assuming that a dual-die flash is connected on the first set of external pads, the address space is divided into two parts, one for each device of the dual die package. Refer to the following table for the address mapping. The byte ordering for 32 bit access is given in Table 21-16 and for 64 bit read access the byte ordering is given in Table 21-20.

Table 21-9. Memory Mapped Individual Flash Mode - Flash A Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
QSPI_AMBA_BASE + 0x00	QSPI_AMBA_BASE + 0x00	0x00_0000 to 0x00_0003	A1
QSPI_AMBA_BASE + 0x04		0x00_0004 to 0x00_0007	
…	…	…	
TOP_ADDR_MEMA1 - 0x08	TOP_ADDR_MEMA1 - 0x08	(TOP_ADDR_MEMA1- 0x08) to (TOP_ADDR_MEMA1 - 0x04 -0x01)	
TOP_ADDR_MEMA1 - 0x04		(TOP_ADDR_MEMA1 - 0x04) to (TOP_ADDR_MEMA1 - 0x01)	
TOP_ADDR_MEMA1 + 0x00	TOP_ADDR_MEMA1 + 0x00_0000	0x00_0000 to 0x00_0003	A2
TOP_ADDR_MEMA1 + 0x04		0x00_0004 to 0x00_0007	
…..	…	…	
TOP_ADDR_MEMA2 - 0x08	TOP_ADDR_MEMA2 - 0x08	(TOP_ADDR_MEMA2 - 0x08) to (TOP_ADDR_MEMA2 - 0x04 - 0x01)	
TOP_ADDR_MEMA2 - 0x04		(TOP_ADDR_MEMA2 - 0x04) to (TOP_ADDR_MEMA2 - 0x01)	

The available address range depends from the size of the external serial flash device. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

21.5.3 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B

Starting with address TOP_ADDR_MEMA2 the content of the first external serial flash devices is mapped into the address space of the device containing the QuadSPI module. Serial flash address byte address 0x0 corresponds to bus address TOP_ADDR_MEMA2 with increasing order. Assuming that a dual-die flash is connected on the first set of external pads, the address space is divided into two parts, one for each device of the dual die package. Refer the following table for the address mapping. The byte ordering for 32 bit access is given in Table 21-16 and for 64 bit read access the byte ordering is given in Table 21-20.

Table 21-10. Memory Mapped Individual Flash Mode - Flash B Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
TOP_ADDR_MEMA2 + 0x00	TOP_ADDR_MEMA2 + 0x00	0x00_0000 to 0x00_0003	B1
TOP_ADDR_MEMA2 + 0x04		0x00_0004 to 0x00_0007	
…	…	…	
TOP_ADDR_MEMB1 - 0x08	TOP_ADDR_MEMB1 - 0x08	(TOP_ADDR_MEMB1-TOP_ADDR_MEMA2 - 0x08) to (TOP_ADDR_MEMB1 - TOP_ADDR_MEMA2 - 0x04 -0x01)	
TOP_ADDR_MEMB1 - 0x04		(TOP_ADDR_MEMB1-TOP_ADDR_MEMA2 - 0x04) to (TOP_ADDR_MEMB1-TOP_ADDR_MEMA2- 0x01)	
TOP_ADDR_MEMB1 + 0x00	TOP_ADDR_MEMB1 + 0x00_0000	0x00_0000 to 0x00_0003	B2
TOP_ADDR_MEMB1 + 0x04		0x00_0004 to 0x00_0007	
…..	…	…	
TOP_ADDR_MEMB2 - 0x08	TOP_ADDR_MEMA2 - 0x08	(TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1- 0x08) to (TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1 - 0x04 - 0x01)	
TOP_ADDR_MEMB2 - 0x04		(TOP_ADDR_MEMB2-TOP_ADDR_MEMB1 - 0x04) to (TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1 - 0x01)	

The available address range depends from the size of the external serial flash device. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

21.5.4 Parallel Flash Mode

Any of the AHB flexible-buffers can be configured to work in parallel flash mode by programming the QSPI_BFGENCR[PAR_EN] bit to '1'. When parallel mode is set, Flash A1 is paired with Flash B1 and Flash A2 is paired with Flash B2. In parallel mode, software should ensure that the size of Flash A1(A2) is equal to the size of Flash B1(B2).

Reads from any even AHB bus address provides bits [7:4] of both serial flash devices and reads from any odd AHB bus address provides bits [3:0] of both flash devices. Refer to the following table for the address mapping. The byte ordering for 32 bit access is given in Table 21-18 and for 64 bit read access the byte ordering is given in Table 21-20.

Table 21-11. Memory Mapped Parallel Flash Mode Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash A Byte Address	Serial Flash B Byte Address
QSPI_AMBA_BASE + 0x0000_0000	QSPI_AMBA_BASE + 0x00 For details, please refer to Parallel mode and Dual Die Flashes.	0x00_0000 - 0x00_0001	0x00_0000 - 0x00_0001
QSPI_AMBA_BASE + 0x0000_0004		0x00_002 - 0x00_0003	0x00_0002 - 0x00_0003
QSPI_AMBA_BASE + 0x0000_0008	QSPI_AMBA_BASE + 0x08	0x00_0004 - 0x00_0005	0x00_0004 - 0x00_0005
QSPI_AMBA_BASE + 0x0000_000C		0x00_0006 - 0x00_0007	0x00_0006 - 0x00_0007
…	…	…	…
TOP_ADDR_MEMB2 - 0x08	TOP_ADDR_MEMB2 - 0x08	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x08)/2	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x08)/2
TOP_ADDR_MEMB2 - 0x04		(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x04)/2 +0x01	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE- 0x04)/2 + 0x01

The available address range covers 27 address bits, corresponding to 128 MB per flash device. The usable space depends from the size of the external serial flash devices. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

21.5.5 AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)

NOTE

See the System Memory map in this document for the base address of the QSPI AHB RX Data Buffer.

memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	AHB RX Data Buffer register (ARDB0)	32	R/W	0000_0000h	21.5.5.1/ 1913
4	AHB RX Data Buffer register (ARDB1)	32	R/W	0000_0000h	21.5.5.1/ 1913
8	AHB RX Data Buffer register (ARDB2)	32	R/W	0000_0000h	21.5.5.1/ 1913
C	AHB RX Data Buffer register (ARDB3)	32	R/W	0000_0000h	21.5.5.1/ 1913
10	AHB RX Data Buffer register (ARDB4)	32	R/W	0000_0000h	21.5.5.1/ 1913
14	AHB RX Data Buffer register (ARDB5)	32	R/W	0000_0000h	21.5.5.1/ 1913
18	AHB RX Data Buffer register (ARDB6)	32	R/W	0000_0000h	21.5.5.1/ 1913
1C	AHB RX Data Buffer register (ARDB7)	32	R/W	0000_0000h	21.5.5.1/ 1913
20	AHB RX Data Buffer register (ARDB8)	32	R/W	0000_0000h	21.5.5.1/ 1913
24	AHB RX Data Buffer register (ARDB9)	32	R/W	0000_0000h	21.5.5.1/ 1913
28	AHB RX Data Buffer register (ARDB10)	32	R/W	0000_0000h	21.5.5.1/ 1913
2C	AHB RX Data Buffer register (ARDB11)	32	R/W	0000_0000h	21.5.5.1/ 1913
30	AHB RX Data Buffer register (ARDB12)	32	R/W	0000_0000h	21.5.5.1/ 1913
34	AHB RX Data Buffer register (ARDB13)	32	R/W	0000_0000h	21.5.5.1/ 1913
38	AHB RX Data Buffer register (ARDB14)	32	R/W	0000_0000h	21.5.5.1/ 1913
3C	AHB RX Data Buffer register (ARDB15)	32	R/W	0000_0000h	21.5.5.1/ 1913
40	AHB RX Data Buffer register (ARDB16)	32	R/W	0000_0000h	21.5.5.1/ 1913
44	AHB RX Data Buffer register (ARDB17)	32	R/W	0000_0000h	21.5.5.1/ 1913

Table continues on the next page...

memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
48	AHB RX Data Buffer register (ARDB18)	32	R/W	0000_0000h	21.5.5.1/ 1913
4C	AHB RX Data Buffer register (ARDB19)	32	R/W	0000_0000h	21.5.5.1/ 1913
50	AHB RX Data Buffer register (ARDB20)	32	R/W	0000_0000h	21.5.5.1/ 1913
54	AHB RX Data Buffer register (ARDB21)	32	R/W	0000_0000h	21.5.5.1/ 1913
58	AHB RX Data Buffer register (ARDB22)	32	R/W	0000_0000h	21.5.5.1/ 1913
5C	AHB RX Data Buffer register (ARDB23)	32	R/W	0000_0000h	21.5.5.1/ 1913
60	AHB RX Data Buffer register (ARDB24)	32	R/W	0000_0000h	21.5.5.1/ 1913
64	AHB RX Data Buffer register (ARDB25)	32	R/W	0000_0000h	21.5.5.1/ 1913
68	AHB RX Data Buffer register (ARDB26)	32	R/W	0000_0000h	21.5.5.1/ 1913
6C	AHB RX Data Buffer register (ARDB27)	32	R/W	0000_0000h	21.5.5.1/ 1913
70	AHB RX Data Buffer register (ARDB28)	32	R/W	0000_0000h	21.5.5.1/ 1913
74	AHB RX Data Buffer register (ARDB29)	32	R/W	0000_0000h	21.5.5.1/ 1913
78	AHB RX Data Buffer register (ARDB30)	32	R/W	0000_0000h	21.5.5.1/ 1913
7C	AHB RX Data Buffer register (ARDB31)	32	R/W	0000_0000h	21.5.5.1/ 1913

21.5.5.1 AHB RX Data Buffer register (ARDB*n*)

The AHB RX Data Buffer register 0 to 31 can be used to read the buffer content of the RX Buffer from successive addresses. QSPI_ARDB0 corresponds to the RX Buffer register entry corresponding to the current value of the read pointer with increasing order.

The increment of the read pointer depends from the access scheme (DMA or flag-driven). Refer to "Data Transfer from the QuadSPI Module Internal Buffers" section in Flash Read section, RX Buffer, data read via register interface and AHB read, for the description of successive accesses to the RX Buffer content. Refer also to Byte Ordering of Serial Flash Read Data for the byte ordering scheme.

Valid address range accessible in the QSPI_ARDBn range depends from the number of RX Buffer entries implemented and from the number of valid buffer entries available in the RX Buffer.

- Example 1, RX Buffer filled completely with 32 words: In this case the address range for valid read access extends from QSPI_ARDB0 to QSPI_ARDB31.
- Example 2, RX Buffer filled with 5 valid words, RX Buffer fill level QSPI_RBSR[RDBFL] is 5. In this case an access to QSPI_ARDB4 provides the last valid entry.

Address: 0h base + 0h offset + (4d × i), where i=0d to 31d

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

ARDB*n* field descriptions

Field	Description
ARXD	ARDB provided RX Buffer Data. Byte order (endianness) is identical to the RX Buffer Data Registers.

21.6 Interrupt Signals

The interrupt request lines of the QuadSPI module are mapped to the internal flags according to the following table.

Table 21-12. Assignment of Interrupt Request Lines

IRQ/DMA line	QSPI_FR Flag	Interrupt Description
ipi_int_tfff	TBFF	TX Buffer Fill
ipi_int_tcf	TFF	Peripheral Command Transaction Finished
ipi_int_rfdf	RBDF	RX Buffer Drain
ipi_int_overrun		Buffer Overflow/Underrun Error Logical OR from:
	RBOF	RX Buffer Overrun
	TBUF	TX Buffer Underrun
	ABOF	AHB Buffer Overflow
ipi_int_cerr		Serial Flash Command Error Logical OR from:
	IPAEF	Peripheral access while AHB busy Error
	IPIEF	Peripheral Command could not be triggered Error
	IPGEF	Peripheral access while AHB Grant Error

Table continues on the next page...

Table 21-12. Assignment of Interrupt Request Lines (continued)

IRQ/DMA line	QSPI_FR Flag	Interrupt Description
ipi_int_ored	DLPFF, TBFF, TFF, ILLINE, RBDF, RBOF, TBUF, ABSEF, ABOF, IPAEF, IPIEF, IPGEF	Logical OR from all the QSPI_FR flags mentioned

21.7 Functional Description

This section provides the functional information of the QuadSPI module.

21.7.1 Serial Flash Access Schemes

The following access schemes are possible, depending on the serial flash devices attached to the QuadSPI module.

Table 21-13. Access Schemes for Serial Flash Data Access

Access Scheme	One Flash Device on Port A	One Flash Device on Port B	Two identical Flash Devices connected on Port A and Port B
Individual Flash Mode: Access to Flash A	**Yes**	**N/a**	**Yes**
Individual Flash Mode: Access to Flash B	**N/a**	**Yes**	**Yes**
Parallel Flash Mode: Read from Flash A and Flash B	**N/a**	**N/a**	**Yes**

Note

If two flash devices are accessed in Parallel Flash Mode, they are accessed with identical control signals. Special alignment on per-flash basis is **not** possible. It is within the responsibility of the application to ensure that the identical signals are applicable to both flash devices.

In Parallel Flash Mode, both external serial flash devices appear logically as one single memory doubled in size with respect to one individual flash device.

If two different flash devices are attached, they can be operated only in Individual Flash Mode.

In the Parallel Flash Mode, only data read commands are supported. Any other IP Command will result in an error condition signaled by the assertion of the QSPI_FR[IUEF] flag and any other AHB Command will result in the assertion of the QSPI_FR[ABSEF] flag.

In the Individual Flash Mode, all supported commands are available.

Unless explicitly noted, all the following descriptions relate to the Individual Flash Mode.

21.7.2 Normal Mode

This mode is used to allow communication with an external serial flash device. Compared to the standard SPI protocol, this communication method uses up to 4 bidirectional data lines operating at high data rates. The communication to the external serial flash device consists of an instruction code and optional address, mode, dummy and data transfers. The flexible programmable core engine described below is immune to a wide variety of command/protocol differences in the serial flash devices provided by various flash vendors.

21.7.2.1 Programmable Sequence Engine

The core of the QuadSPI module is a programmable sequence engine that works on "instruction-operand" pairs. The core controller executes each programmed instruction sequentially. The complete list of instructions and the corresponding operands is given in the following table.

Table 21-14. Instruction set

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
CMD	6'd1	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	8 bit command value	Provide the serial flash with operand on the number of pads specified
ADDR	6'd2		Number of address bits to be sent (for example, 8'd24 => 24 address bits required)	Provide the serial flash with address cycles according to the operand on the number of pads specified. The actual address to be provided will be derived from the incoming address in case of AHB initiated transactions and the value of SFAR in case of IPS initiated transactions .

Table continues on the next page...

Table 21-14. Instruction set (continued)

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
DUMMY	6'd3		Number of dummy clock cycles (should be <= 64 cycles)	Provide the serial flash with dummy cycles as per the operand. The PAD information defines the number of pads in input mode. (for example, one pad implies that pad 1 is not driven, rest all are driven)
MODE	6'd4		8 bit mode value	Provide the serial flash with 8 bit operand on the number of pads specified
MODE2	6'd5	N=2'd{0,1}	2 bit mode value	Provide the serial flash with 2 bit operand on the number of pads[1] specified
MODE4	6'd6	N=2'd{0,1,2}	4 bit mode value	Provide the serial flash with 4 bit operand on the number of pads[2] specified
READ	6'd7	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	Read data size in bytes (the user's application should ensure that data size is a multiple of 8 bytes)	Read data from flash on the number of pads specified. The data size may be overwritten by writing to the ADATSZ field of the QSPI_BUFxCR registers for AHB initiated transactions and IDATSZ field of IP Configuration Register (QuadSPI_IPCR) for IP initiated transactions.
WRITE	6'd8		Write data size in bytes	Write data on number of pads sepcified. The data size may be overwritten by writing to the IDATSZ field of IP Configuration Register (QuadSPI_IPCR) register
JMP_ON_CS	6'd9	NA	Instruction number	Every time the chip select (CS) is deasserted, jump to the instruction pointed to by the operand. This instruction allows the programmer to specify the behavior of the controller when a new read transaction is initiated following a CS deassertion.
ADDR_DDR	6'd10	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	Number of address bits to be sent (for example, 8'd24 => 24 address bits required)	Provide the serial flash with address cycles according to the operand on the number of pads specified at each clock edge of serial flash clock. The actual address to be provided will be derived from the incoming address in case of AHB initiated transactions and the value of QSPI_SFAR in case of IPS initiated transactions .
MODE_DDR	6'd11		8 bit mode value	Provide the serial flash with 8 bit operand on the number of pads specified at each clock edge of serial flash.
MODE2_DDR	6'd12	N=2'd{0}	2 bit mode value	Provide the serial flash with 2 bit operand on the number of pads specified at each clock edge of serial flash[3]
MODE4_DDR	6'd13	N=2'd{0,1}	4 bit mode value	Provide the serial flash with 4 bit operand on the number of pads specified at each clock edge of serial flash[4].
READ_DDR	6'd14	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads	Read data size in bytes (the user's application should ensure that data size is in multiple of 8 bytes)	Read data from flash on the number of pads specified at each clock edge of serial flash. The data size may be overwritten by writing to the ADATSZ field of the QSPI_BUFxCR registers for AHB initiated transactions and IDATSZ field of IP Configuration Register (QuadSPI_IPCR) for IP initiated transactions

Table continues on the next page...

Table 21-14. Instruction set (continued)

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
WRITE_DDR	6'd15	2'd3- Eight pads	Write data size in bytes	Write data on the number of pads specified at each clock edge of serial flash. The data size may be overwritten by writing to the IDATSZ field of IP Configuration Register (QuadSPI_IPCR) register
DATA_LEARN	6'd16		Number of data in bytes to be used for data learning. Say if operand is 1 than 8 bit data_learning is enabled.	Finds the correct sampling point in case of only DDR operations. When this instruction is encountered, the QSPI_SMPR[DDRSMP] values are ignored and the controller finds the correct sampling point on its own by data learning. But this feature of sampling point is valid for only DDR modes. If DQS mode is enabled, then data learn instruction just matches the incoming data from flash and if it does not matches the QSPI_DLPV then Flag in QSPI_FR[DLPFF] is set.
STOP	8'd0	NA	NA	Stop execution; deassert CS

1. For a one pad instruction, MODE2 will take 2 serial flash clock cycles on the flash interface.
2. For a one pad instruction, MODE4 will take 4 serial flash clock cycles on the flash interface. For a 4 pad instruction, MODE4 will take 1 serial flash clock cycle on the flash interface.
3. For a one pad instruction, MODE2_DDR will take 1 serial flash clock cycle on the flash interface.
4. For a one pad instruction, MODE4_DDR will take 2 serial flash clock cycles on the flash interface. For a 4 pad instruction MODE4_DDR will take half a cycle on the serial flash interface.

The programmable sequence engine allows the user to configure the QuadSPI module according to the serial flash connected on board. The flexible structure is easily adaptable to new command/protocol changes from different vendors.

21.7.2.2 Flexible AHB buffers

In order to reduce the latency of the reads for AHB masters, the data read from the serial flash is buffered in flexible AHB buffers. There are four such flexible buffers. The size of each of these buffers is configurable with the minimum size being 0 Bytes and maximum size being the size of the complete buffer instantiated.The size of buffer 0 is defined as being from 0 to QSPI_BUF0IND. The Size of buffer 1 is from QSPI_BUF0IND to QSPI_BUF1IND, buffer2 is from QSPI_BUF1IND to QSPI_BUF2IND and buffer 3 is from QSPI_BUF2IND to the size of the complete buffer, which is given in the chip-specific QuadSPI information.

Each flexible AHB buffer is associated with the following

1. An AHB master. Optionally, buffer3 may be configured as an "all master" buffer by setting the QSPI_BUF3CR[ALLMST] bit. When buffer3 is configured in such a way, any access from a master not associated with any other buffer is routed to buffer3.

2. A datasize field representing the amount of data to be fetched from the flash on every "missed" access.

The master port number of every incoming request is checked and the data is returned/ fetched into the corresponding associated buffer. Every "missed" access to the buffer causes the controller to clear the buffer and fetch QSPI_BUFxCR[ADATSZ] amount of data from the serial flash.As such, there is no benefit in configuring a buffer size of greater than ADATSZ, as the locations greater than ADATSZ will never be used. For any AHB access, the sequence pointed to by the QSPI_BFGENCR[SEQID] field is used for the flash transaction initiated. The data is returned to the master as soon as the requested amount is read from the serial flash. The controller however, continues to prefetch the rest of the data in anticipation of a next consecutive request. Figure 21-3 shows the flexible AHB buffers.

The QSPI_BFGENCR[SEQID] field points to an index of the LUT. Refer to Look-up Table for details.

Figure 21-3. Flexible AHB Buffers

Buffer0 may optionally be configured to be associated with a high priority master by setting the QSPU_BUF0CR[HP_EN] bit. An access by a high priority master suspends any ongoing prefetch to any of the other buffers. The ongoing prefetch is suspended and

the high priority master is serviced first. Once the high priority masters access completes, the suspended transaction is resumed (before any other AHB access is entertained). The status of the suspended buffer can be read from Sequence Suspend Status Register (QuadSPI_SPNDST).

21.7.2.3 Suspend-Abort Mechanism

Any low priority AHB access can be suspended by a high priority AHB master request. The ongoing transaction is suspended at 64 bit boundary. The suspended transaction is restarted after the high priority master is served and the high priority transaction including data prefetch is completed. While a transaction is in suspended state, it may be aborted if a transaction by the same suspended master is made to a location which is different from the location of the suspended transaction.

Any ongoing transaction is aborted if a request from the same master arrives for a location other than the location at which the transaction is going on. The abort can happen at any point of time.

21.7.2.4 Look-up Table

The Look-up-table or LUT consists of a number of pre-programmed sequences. Each sequence is basically a sequence of instruction-operand pairs which when executed sequentially generates a valid serial flash transaction. Each sequence can have a maximum of 8 instruction-operand pairs. The LUT can hold a maximum of 16 sequences. The figure below shows the basic structure of the sequence in the LUT.

Figure 21-4. LUT and sequence structure

At reset, the index 0 of the look-up-table (LUT[0..3]) is programmed with a basic read sequence as given in Table 21-15. After reset the complete LUT may be reprogrammed according to the device connected on board.In order to protect its contents during a code runover the LUT may be locked, after which a write to the LUT will not be successful until it has been unlocked again.The key for locking or unlocking the LUT is **0x5AF05AF0**. The process for locking and un-locking the LUT is as follows:

Locking the LUT

1. Write the key **(0x5AF05AF0)** in to the LUT Key Register (QuadSPI_LUTKEY).
2. Write 0b01 to the LUT Lock Configuration Register (QuadSPI_LCKCR). Note that this IPS transaction should immediately follow the above IPS transaction (no other IPS transaction can be issued in between). A successful write into this register locks the LUT.

Unlocking the LUT

1. Write the key **(0x5AF05AF0)** into the LUT Key Register (QuadSPI_LUTKEY)
2. Write 0b10 to the LUT Lock Configuration Register (QuadSPI_LCKCR). Note that this IPS transaction should immediately follow the above IPS transaction (no other IPS transaction can be issued in between). A successful write into this register unlocks the LUT.

The lock status of the LUT can be read from QSPI_LCKCR[UNLOCK] and QSPI_LCKCR[LOCK] bit.

Some example sequences are defined in Example Sequences. The reset sequence at LUT index 0 is given in the following table.

Table 21-15. Reset sequence

Instruction	Pad	Operand	Comment
CMD	0x00	0x03	Read Data byte command on one pad
ADDR	0x00	0x18	24 Addr bits to be sent on one pad
READ	0x00	0x08	Read 64 bits
JMP_ON_CS	0x00	0x00	Jump to instruction 0 (CMD)

21.7.2.5 Issuing Serial Flash Memory (SFM) Commands

Each access to the external device follows the same sequence:

1. The user must pre-populate the LUT with the serial flash command sequences that are required for the flash device being used.
2. The QuadSPI module starts executing the instructions in the sequence one by one. The transaction starts and the status bit QSPI_SR[BUSY] is set.
3. Communication with the external serial flash device is started and the transaction is executed.
4. When the transaction is finished (all transmit- and receive operations with the external serial flash device are finished) the status bit QSPI_SR[BUSY] is reset. In case of an IP Command the QSPI_FR[TFF] flag is asserted.

Further details are given in below in Flash Programming and Flash Read.

You can trigger the processing of SFM commands in the QuadSPI module in one of the following ways:

- **Using IP commands**

  For IP Commands the required components need to be written into the following registers:

  - Write the serial flash address to be used by the instruction into QSPI_SFAR, refer to Serial Flash Address Register (QSPI_SFAR). For IP Commands not related to specific addresses, the base address of the related flash need to be

programmed.For example, for an instruction which does not require an address (i.e. write enable instruction) the SFAR should be programmed with the base address of the memory the command is to be sent to.

- Write the sequence ID and data size details in the IP Configuration Register (QSPI_IPCR).
- Note that the write into the QSPI_IPCR[SEQID] field must be the last step of the sequence. It is possible to combine all fields of the QSPI_IPCR into one single write. Refer to IP Configuration Register (QSPI_IPCR) for details.

Note that there are some conditions where no IP Command is executed after writing the QSPI_IPCR[SEQID] field and the write operation itself is ignored. They are described in Command Arbitration.

- **Using AHB commands**

  Any AHB memory mapped access is routed to one of the buffers depending on the master port number of the request. If the access is a "miss", a new serial flash transaction is started. The transaction is based on the sequence pointed to by the BFGENCR[SEQID] field as described in Flexible AHB buffers.

  An AHB access is termed memory mapped when the access is to the memory mapped serial flashes, as described in Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A and Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B.

  Again the possible error conditions are described in Command Arbitration .

21.7.2.6 Flash Programming

In all cases the memory sector to be written needs to be erased first. The programming sequence itself is then initiated in the following way:

1. Check QuadSPI_SR[BUSY] is de-asserted or '0'. Check that the TX buffer is empty. If it is required to discard the data present in the TX buffer (QSPI_SR[TXEDA]) bit is set, then the TX buffer must be cleared by writing '1' into the QSPI_MCR[CLR_TXF] bit.
2. Program the address related to the command in the QSPI_SFAR register.
3. Provide initial data for the program command into the circular buffer via register TX Buffer Data Register (QSPI_TBDR) . At least four word of data must be written into the TX Buffer up to a maximum of 32.

4. Program the QSPI_IPCR register to trigger the command. The QSPI_IPCR[SEQID] should point to an index of the LUT which has the flash program sequence pre-programmed.The IDATSZ field should be set to denote the size of the write.
5. Depending on the amount of data required, step 3 must be repeated until all the required data have been written into the QSPI_TBDR register.The QSPI_SR[TXFULL] can be used to check if the buffer is ready to receive more data. At any time, the QSPI_TBSR[TRCTR] field can be read to check how many words have been written actually into the TX Buffer.

Upon writing the QSPI_IPCR[SEQID] field (refer to step 4) the QuadSPI module will start to execute the programmed sequence. It is the responsibility of the software to ensure that a correct sequence is programmed into the LUT in accordance with the flash memory connected to the module. The data is fetched from the TX Buffer. It consists of **32** entries of 32-bits and is organized as a circular FIFO, whose read pointer is incremented by four after each fetch. When all data are transmitted, the QuadSPI module will return from 'busy' to 'idle'. However, this is not true for the external device since the internal programming is still ongoing. It is up to the user to monitor the relevant status information available from the serial flash device and to ensure that the programming is finished properly.

21.7.2.7 Flash Read

Host access to the data stored in the external serial flash device is done in two steps. First, the data must be read into the internal buffers and in the second step these internal buffers can be read by the host.

1. **Reading Serial Flash Data into the QuadSPI Module Internal Buffers**

   A read access to the external serial flash device can be triggered in two different ways:

   - **IP Command Read**: For **reading flash data into the RX Buffer** the user must provide the correct sequence ID in the QuadSPI_IPCR[SEQID] register. The sequence ID points to a sequence in the LUT. It is the responsibility of the software to ensure that a correct read sequence is programmed in the LUT in accordance with the serial flash device connected on board. The user should program the Serial Flash Address Register (QSPI_SFAR) and the IP Configuration Register (QSPI_IPCR) registers. All available read commands supported by the external serial flash are possible.

     Optionally it is possible to clear the RX Buffer pointer prior to triggering the IP Command by writing a 1 into the QuadSPI_MCR[CLR_RXF] field.

From these inputs, the complete transaction is built when the QSPI_IPCR[SEQID] field is written. The transaction related to the read access starts and the requested number of bytes is fetched from the external serial flash device into the RX Buffer. Since the read access is triggered by an IP command, the IP_ACC status bit and the BUSY bit are both set (both are located in the Status Register (QSPI_SR)). A count of the number of entries currently in the Rx Buffer can be obtained from QSPI_RBSR[RDBFL].

The communication with the external serial flash is stopped when the specified number of bytes has been read (successful completion of the transaction).

- **AHB Command Read**: For **reading flash data into the AHB Buffer** the user must set up a read access by a master to the address range in the system memory map which the external serial flash devices are mapped to. The user should also program the buffer registers corresponding to the AHB master initiating the request, this is depends on the configuration of the QSPI_RBCT[RXBRD]. The user should provide the correct sequence ID into the buffer generic configuration register (QSPI_BFGENCR). It is the responsibility of the software to ensure that a correct read sequence in programmed in the LUT in accordance with the serial flash device connected on board. Flash device selection and access mode are determined by the address accessed in the AHB address space associated to the QuadSPI module (refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A, Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B and Parallel Flash Mode.

  On each AHB read access to the memory mapped area the valid data in the AHB Buffer is checked against the address requested in the actual read. When the AHB read request can't be served from the content of the AHB Buffer, the buffer is flushed and the sequence pointed to by the sequence ID is executed by the controller. The requested number of buffer entries defined in the QSPI_BUFxCR[ADATSZ] field is then fetched from the external serial flash device into the internal AHB Buffer. Since the read access is triggered via the AHB bus, the QSPI_SR[AHB_ACC] status bit is set driving in turn the QSPI_SR[BUSY] bit until the transaction is finished. The communication with the external serial flash is stopped when the specified number of entries has been filled.

2. **Data Transfer from the QuadSPI Module Internal Buffers**

The data read out from the external serial flash device by the QuadSPI module is stored in the internal buffers.The means of accessing the data from the buffer differs depending on which buffer the data has been loaded to. Refer to Block Diagram for details about the two available buffers, the RX Buffer and the AHB Buffer, in the QuadSPI module:

Figure 21-5. QuadSPI memory map

- The RX Buffer is implemented as FIFO of depth 32 entries of 4 bytes. Its content is accessible in two different address areas both referring to the identical data and the same physical memory.

  In the IPS address space in the area associated to QSPI_RBDR0 to QSPI_RBDR31

  In the AHB address space in the area associated to QSPI_ARDB0 to QSPI_ARDB31. Two successive entries are accessed with one single 64 bit AHB read operation.

RX Buffer operation can be summarized as follows: The QSPI_RBCT[WMRK] field determines at which fill level the RXWE bit is asserted and how many entries are removed from the RX Buffer on each Buffer POP operation. So the QSPI_SR[RXWE] bit indicates that the configured number of data entries is available in the RX Buffer and the QSPI_RBSR[RDBFL] field indicates how

many valid entries are available in total. Note that the first entry (QSPI_RBDR0 or QSPI_ARDB0) always corresponds to the first valid entry in the RX Buffer. The software needs to manage the number of valid data bytes itself.

Further details can be found in RX Buffer Data Register (QuadSPI_RBDR*n*) and in AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31).

- **Flag-based Data Read of the RX Buffer** is done by polling the QSPI_SR[RXWE] bit. When it is asserted the valid entries can be read either via the IPS address space (QSPI_RBDRn) or the AHB address space (QSPI_ARDBn). A Buffer POP operation must be triggered by the application by writing a 1 into the QSPI_FR[RBDF] bit - this automatically updates the FIFO to point to the next entry as defined by RBCT[WMRK]. For example, if WMRK is set to 3, then the buffer will discard 16 bytes of data.
- **DMA controlled Data Read of the RX Buffer** is done by using the DMA module.The application must ensure that the DMA controller of the related device is programmed appropriately like it is described in DMA Usage.

  DMA controlled read out is triggered fully automatically by the assertion of the QSPI_SR[RXWE] bit. The related Buffer POP operation is also handled completely inside the QuadSPI module. Like in the case above, accessing the RX Buffer content either on QSPI_RBDRn or QSPI_ARDBn related addresses is equivalent.

- **AHB Buffer data read via memory mapped access:** This kind of access is done by reading one of the addresses assigned to the external serial flash device(s) within the range given in Table 21-8 table *under the condition that the data requested are already present in the AHB Buffer or it is currently being read from the serial flash device by the instruction in progress.* If this is not the case a memory mapped AHB command read is triggered as described above. If the requested data is already available in the AHB Buffer they are provided directly to the host.

  When AHB access are made to the flash memory mapped address, the data will be fetched and returned to the AHB interface. Till the data is being fetched the AHB interface would be stalled. As soon as the data from the requested address has been read by the QuadSPI module the AHB read access is served. So it is possible to run sequential reads from the AHB buffer at arbitrary speed without the need to monitor any information about the availability of the data. Nevertheless this access scheme stalls the AHB bus for the time required to read the data from the serial flash device. A better way is (when it is known the

access is sequential) to have a prefetch enabled (by programming the ADATSZ field) such that before the next sequential AHB access come, the data is already fetched into the buffer.

As long as the host restricts its accesses to the data already in the buffer and the data currently fetched from the serial flash, it is possible to run the host read from the AHB Buffer in parallel to the serial flash read into the AHB Buffer.

21.7.2.8 Byte Ordering of Serial Flash Read Data

In this paragraph the byte ordering of the serial flash data is given. The basic scheme is that the **first** byte read out of the serial flash device - which is addressed by the QSPI_SFAR[SFADR] field - corresponds to bit position QSPI_RBDR0[31:24] register for IP Command read. In contrast to that for AHB Command read the bytes are always positioned according to the byte ordering of the AHB bus.

- **Byte Ordering in Individual Flash Mode**

  The following table gives the byte ordering scheme of how the byte oriented data space of the serial flash device is mapped into one single 32 bit entry of the RX Buffer or the AHB Buffer. The table is valid within the following context:

  - Flash A or Flash B in Individual Flash Mode
  - All AHB data read commands with access size of 32 bit

Table 21-16. Byte Ordering in Individual Flash Mode

Serial Flash Byte Numbering	3	2	1	0
Buffer Entry Bit Position [31:0] (32 Bit data width)	[31:24]	[23:16]	[15:8]	[7:0]

Note

For IP Commands the read size can be given in number of bytes. If this number is not a multiple of 4, then the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

For AHB Commands, reads, starting from an address not aligned to 32 bit boundaries, the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- **Byte Ordering in Parallel Flash Mode**

In Parallel Flash Mode each byte is combined out of 2 half bytes which are read in parallel from the two serial flash devices. The following tables shows how the flash content is separated into the half bytes and how the half bytes are assembled to the content of the QSPI_RBDR0 register.

Table 21-17. Serial Flash Device Half Byte Ordering

Serial Flash Device Byte #	Flash A Bit Position		Flash B Bit Position	
	[7:4]	[3:0]	[7:4]	[3:0]
0	fah0	fal0	fbh0	fbl0
1	fah1	fal1	fbh1	fbl1
2	fah2	fal2	fbh2	fbl2
3	fah3	fal3	fbh3	fbl3
4	fah4	fal4	fbh4	fbl4
5	fah5	fal5	fbh5	fbl5
6	fah6	fal6	fbh6	fbl6
7	fah7	fal7	fbh7	fbl7
8	fah8	fal8	fbh8	fbl8

The table entry naming reflects the half byte positioning in the serial flash devices:

- **<fa>**h0 means **Flash A**, <fb>h0 means Flash B.
- fa**<h>**0 means half byte in **high position**, fa<l>o means half byte in low position.
- fah**<0>** means **physical byte address 0** in the serial flash device, fal<1> means physical byte address 1 in the serial flash device.

Table 21-18. Byte Ordering in Parallel Flash Mode - RX Buffer

QSPI_SFAR[SFADR] set to 0x000_0000

Table continues on the next page...

Table 21-18. Byte Ordering in Parallel Flash Mode - RX Buffer (continued)

QSPI_RBDR0 **QSPI_ARDB0**	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
QSPI_RBDR1 **QSPI_ARDB1**	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2
QSPI_SFAR[SFADR] set to 0x000_0001								
QSPI_RBDR0 **QSPI_ARDB0**	fal2	fbl2	fah2	fbh2	fal1	fbl1	fah1	fbh1
QSPI_RBDR1 **QSPI_ARDB1**	fal4	fbl4	fah4	fbh4	fal3	fbl3	fah3	fbh3

Note

For IP Commands the read size can be given in number of bytes. If this number is not a multiple of 4 the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

Table 21-19. Byte Ordering in Parallel Flash Mode - AHB Buffer

AHB Address **(32 Bit Access)**	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
AHB Address 0x800_0004 **(32 Bit Access)**	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2

Note

For AHB Command read starting from an address not aligned to 32 bit boundaries or AHB access size smaller than 32 bit the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- **Buffer Entry Ordering for 64 Bit Read Access**

For read access via the AHB interface 64 bit access is possible. Each 64 bit access reads 2 32 bit entries simultaneously. The ordering of these 32 bit entries within the 64 bit word is given in the following table.

Table 21-20. 64 Bit Read Access Buffer Entry Ordering

AHB Read Data Bit Position [63:0]	[63:32]	[31:0]
Buffer Entry #	Odd (1, 3, 5, …)	Even (0, 2, 4, …)

21.7.2.9 Normal Mode Interrupt and DMA Requests

The QuadSPI module has different flags that can only generate interrupt requests and one flag that can generate interrupt as well as DMA requests. The following table lists the eight conditions. Note that the flags mentioned in the table are related to the Flag Register (QSPI_FR).

Table 21-21. Interrupt and DMA Request Conditions

Condition	Flag(QSPI_FR)	DMA
Data Learn patttern Failure	DLPFF	-
TX Buffer Fill	TBFF	-
TX Buffer Underrun	TBUF	-
Illegal Instruction Error	ILLINE	-
RX Buffer Drain	RBDF	X
RX Buffer Overflow	RBOF	-
AHB Buffer Overflow	ABOF	-
AHB Sequence Error	ABSEF	-
IP Command Usage Error	IUEF	-
IP Command Trigger during AHB Access Error	IPAEF	-
IP Command Trigger could not be executed Error	IPIEF	-
IP Access during AHB Grant Error	IPGEF	-
IP Command related Transaction Finished	TFF	-

Each condition has a flag bit in the Flag Register (QSPI_FR) and a Request Enable bit in the DMA Request Select and Enable Register (QSPI_RSER) . The RX Buffer Drain Flag (RBDF) has separate enable bits for generating IRQ and DMA requests. Note that not all flags have an individual IRQ line. Check the devices Interrupt Vector Table for more details.

- Transmit Buffer Fill Interrupt Request:

The Transmit Buffer Fill IRQ indicates that the TX Buffer can accept new data. It is asserted if the QSPI_FR[TBFF] flag is asserted and if the corresponding enable bit (QSIP_RSER[TBFIE]) is set. Refer to TX Buffer Operation, for details about the assertion of the QSPI_FR[TBFF] flag.

- Receive Buffer Drain Interrupt or DMA Request:

  The Receive Buffer Drain IRQ derived from the QSPI_FR[RBDF] flag indicates that the RX Buffer of the QuadSPI module has data available from the serial flash device to be read by the host. It remains set as long as the QSPI_RBSR[RXWE] bit is set. The QSPI_RSER[RBDIE] bit enables the related IRQ.

  Aside from the IRQ it is possible to handle RX Buffer drain by DMA. If the QSPI_RSER[RBDDE] bit is set, a DMA request will be triggered when the RX Buffer contains more than QSPI_RBCT[WMRK] valid entries.The application must set the environment appropriately (for example, the DMA controller) for the DMA transfers.

- Buffer Overflow/Underrun Interrupt Request:

  The Buffer Overflow/Underrun IRQ is a combination of the following flags (all located in the QSPI_FR register with the related enable bits in the QSPI_RSER register):

  - TBUF - TX Buffer Underrun, enabled by TBUIE
  - RBOF - RX Buffer Overflow, enabled by RBOIE
  - ABOF - AHB Buffer Overflow, enabled by ABOIE

    The Transmit Buffer Underrun indicates that an underrun condition in the TX Buffer has occurred. It is generated when a write instruction is triggered whilst the Tx Buffer is empty and the QSPI_RSER[TBUIE] bit is set.

    The Receive Buffer Overflow indicates that an overflow condition in the RX Buffer has occurred. It is generated when the RX Buffer is full, an additional read transfer attempts to write into the RX Buffer and the QSPI_RSER[RBOIE] bit is set.

    The AHB Buffer Overflow indicates that an overflow condition in the AHB Buffer has occurred. It is generated when the AHB Buffer is full, an additional read transfer attempts to write into the AHB Buffer and the QSPI_RSER[ABOIE] bit is set.

    The data from the transfers that generated the individual overflow conditions is ignored.

- Serial Flash Command Error Interrupt Request

If the IPAEF, IPIEF, IPGEF or IUEF flags in the QSPI_FR are set, and the related interrupt enable bits in the QSPI_RSER are also set, then an interrupt is requested.

- Transaction Finished Interrupt Request

  The IP Command Transaction Finished IRQ indicates the completion of the current IP Command.It is triggered by the QSPI_FR[TFF] flag and is masked by the QSPI_RSER[TFIE] bit.

21.7.2.10 TX Buffer Operation

The TX Buffer provides the data used for page programming. For proper operation it is required to provide at least four entry in the TX Buffer prior to starting the execution of the page programming command. The application must ensure that the required number of data bytes is written into the TX Buffer fast enough as long as the command is executed without a TX Buffer overflow or underrun.

The QuadSPI module sets the QSPI_FR[TBFF] flag so long as the TX Buffer is not full and can accept more data.

When the QuadSPI module tries to pull data out of an empty TX Buffer the TX Buffer underrun is signaled by the QSPI_FR[TBUF] flag.The TX buffer underrun flag is also asserted when TX buffer contains less than 128 bits of data and QuadSPI module tries to pull out data from it. The current IP Command leading to the underrun condition is continued until the specified number of bytes has been sent to the serial flash device, in the underrun condition when QuadSPI module tries to pull out data of empty TX buffer, the data transferred is all F's i.e. once the underrun flag is set under this condition, it will return F's until the required number of bytes are not sent. This has been done to ensure that the software need not to erase whole sector after underrun, just reprogramming from failure point will serve the purpose. When this Sequence Command is finished, the QSPI_FR[TBFF] flag is asserted indicating that the Tx Buffer is ready to be written again.

The TX Buffer overflow isn't signaled explicitly, but the TX Buffer fill level can be monitored by the QSPI_TBSR[TRBFL] field.

Refer to TX Buffer Status Register (QuadSPI_TBSR) and Flag Register (QuadSPI_FR) for details about the TX Buffer related registers.

21.7.2.11 Address scheme

Earlier serial flash memories supported only 24-bit address space hence restricting the maximum memory size of the serial flash as 16 MB. The new memory specification supports two types of 32-bit addressing mode in addition to legacy 24-bit address mode.

- **Extended Address Mode**

  In this mode, the legacy 24-bit commands are converted to accept 32-bit address commands. The flash memory needs to be configured for 32-bit address mode. Also, while programming the LUT sequence in QuadSPI for 32-bit mode, the ADDR and ADDR_DDR command should be programmed with 8'd32 as the operand value. By default, the QuadSPI is in 24-bit legacy address mode. Each of the memory vendors have a different way of enabling this mode (Refer to the memory specification from memory vendors). For example, the command B7h sent to Macronix flash will enable it for 32-bit address mode.

- **Extended Address register**

  In this mode, the upper 8-bit of the 32-bit address is provided by the Extended address register in the memory itself. The memory provides a specific register which is updated according to the address to be accessed. This effectively converts the legacy 24-bit address command into 32-bit address commands. The memories greater in size than 16 MB, consists of banks of 16 MB. The 8-bit written in the extended address register effectively enables a bank. For example in Spansion memory, when the extended address register is updated with a value of 0x01 with the help of the command 17h, it will open Bank1 of the memory. The consequent 24-bit address commands will lead to Bank1. The extended address register needs to be update with the respective value for access to other banks. This effectively converts the legacy 24-bit address command into 32-bit address commands.

21.8 Initialization/Application Information

This section provides the initialization and application information of the QuadSPI module.

21.8.1 Power Up and Reset

Note that the serial flash devices connected to the QuadSPI module may require special voltage characteristics of their inputs during power up or reset. It is the responsibility of the application to ensure this.

21.8.2 Available Status/Flag Information

This paragraph gives an overview of the different status and flag information available and their interdependencies for different use cases. Related registers are QSPI_SR and QSPI_FR. Refer to the related descriptions how to set up the QuadSPI module appropriately.

21.8.2.1 IP Commands

Refer to IP Configuration Register (QuadSPI_IPCR) for additional details not explicitly covered in this paragraph.

- **IP Commands - Normal Operation**

  Writing the QSPI_IPCR[SEQID] field triggers the execution of a new IP Command. Given that this is a legal command the QSPI_SR[IPACC] and the QSPI_SR[BUSY] bits are asserted simultaneously, immediately after the execution is started.

  When the instruction on the serial flash device has been finished these bits are de-asserted and the QSPI_FR[TFF] flag is set.

- **IP Commands - Error Situations**

  Refer to Table 21-22 below.

21.8.2.2 AHB Commands

Refer to Section 1, Reading Serial Flash Data into the QuadSPI Module, in Flash Read for additional details not explicitly covered in this paragraph.

- **AHB Commands - Normal Operation**

  Memory mapped read access to a serial flash address not contained in the AHB Buffer, triggers the execution of an AHB Command. Given that this is a legal command the QSPI_SR[AHBACC] and the QSPI_SR[BUSY] bits are asserted simultaneously immediately after the execution is started. When the instruction on the serial flash device has been finished these bits are de-asserted.

- **IP Commands - Error Situations**

  Refer to Table 21-22 below.

21.8.2.3 Overview of Error Flags

The following table gives an overview of the different error flags in the QSPI_FR register and additional error-related details.

Table 21-22. Overview of QSPI_FR Error Flags

Error Category	Error Flag in QSPI_FR	Command Execution on Serial Flash Device TFF Behavior (in case of IP commands only)	Description
AHB Error Flag	ABSEF	Flash transaction is aborted	AHB sequence contains • WRITE instruction • WRITE_DDR instruction
AHB Error Flag	ABOF	Flash transaction continues until it finishes	Set when the module tried to push data into the AHB buffer that exceeded the size of the AHB buffer. Only occurs due to wrong programming of the QSPI_BUFxCR[ADATSZ].
Miscellaneous Error Flag	DLPFF	Flash transaction continues until it finishes	Set when DATA_LEARN instruction was encountered in a sequence but no sampling point was found for the data learning pattern.
Miscellaneous Error Flag	ILLINE	Flash transaction aborted	Illegal instruction Error Flag – Set when an illegal instruction is encountered by the controller in any of the sequences.
Command Arbitration Error	IPIEF	TFF not asserted in conjunction with that command	IP Command Error - caused when IP access is currently in progress (IP_ACC set) and • write attempt to QSPI_IPCR register. • write attempt to QSPI_SFAR register. • write attempt to QSPI_RBCT register.
Command Arbitration Error	IPAEF		• AHB Command already running, another IP Command could not be executed. • AHB Command already running, write attempt to QSPI_IPCR[SEQID] field.
Command Arbitration Error	IPGEF		• Exclusive access to the serial flash granted for AHB Commands, write attempt to QSPI_IPCR[SEQID] field.
IP Command Error	IUEF	—	• IP Command Usage Error

Table continues on the next page...

Table 21-22. Overview of QSPI_FR Error Flags (continued)

Error Category	Error Flag in QSPI_FR	Command Execution on Serial Flash Device TFF Behavior (in case of IP commands only)	Description
Buffer Related Error	RBOF	TFF is asserted on completion	• RX Buffer Overrun
Buffer Related Error	TBUF		• TX Buffer Underrun

Note that only the buffer related errors are related to a transaction on the external serial flash. All the other errors do not trigger an actual transaction.

21.8.2.4 IP Bus and AHB Access Command Collisions

There are two flags related to this topic, the QSPI_FR[IPAEF] and QSPI_FR[IPIEF]. Refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for a description of the flags and Command Arbitration , for details about possible command collisions.

21.8.3 Exclusive Access to Serial Flash for AHB Commands

It is possible that several masters need to access the serial flash device connected to the QuadSPI module separately, one master by triggering IP Commands and reading the RX Buffer (via RBDR*n* register) and the other masters by triggering AHB Commands (via ARDB*n* Registers). These two set of buffer (RBDR and ARDB Buffer) points to the same physical buffer.Refer to Figure 21-5 To avoid command collisions resulting in excessive latencies the QuadSPI module implements a request-handshake mechanism between the master triggering AHB Commands and the QuadSPI module allowing this specific master to request exclusive access to the serial flash device for AHB Commands. If this exclusive access is granted the execution of IP Commands is blocked. This resolves command collisions and excessive times where the AHB interface may be blocked.

If this capability is used in the device there is additional status and flag information available related to this mechanism. The QSPI_SR[AHBGNT] bit reflects the module-internal state that the exclusive access mentioned above is granted, any attempt to trigger an IP Command is rejected and results in the assertion of the QSPI_FR[IPGEF] flag. Refer to the descriptions of the related bit and flag for details.

It is within the responsibility of the application to set up the master using this mechanism appropriately, if used incorrectly no IP Commands at all can be triggered.

Two different cases can be distinguished:

21.8.3.1 RX Buffer Read via QSPI_ARDB Registers

In this case all masters share the AHB bus for RX Buffer as well as for AHB Buffer read. In this case the access to the AHB interface by the master triggering AHB Commands must be deferred until any pending IP Command has been finished **and** the RX Buffer readout has been finished as well. The QSPI ARDB Buffers access the Rx buffer i.e the data from the Rx Buffer is returned and no data from AHB Buffer is touched. This is the conservative use case, corresponding to the reset value 0 of the QSPI_RBCT[RXBRD] bit.

In this case the QSPI_SR[AHBGNT] bit is asserted not earlier than any running IP Command has been finished (QSPI_SR[IP_ACC] is 0), the RX Buffer has been read out completely (QSPI_RBSR[RDBFL] equal to 0) or no DMA read is pending (QSPI_SR[RXDMA] equal to 0 and Rx Buffer readout is via AHB(QSPI_RBCT[RXBRD]) equal to 1.

21.8.3.2 RX Buffer Read via QSPI_RBDR Registers

This is the preferred use case as an access to the AHB buffer (memory mapped flash) does not interfere with any IPS access to read the RBDR buffer. It is not possible that a pending AHB bus access triggered by an AHB Command stalls the AHB bus and blocks the RX Buffer readout since the RX Buffer is read via the IP bus based registers QSPI_RBDR0 to QSPI_RBDR31.

For this case it is recommended to program the QSPI_RBCT[RXBRD] bit to 1. The QSPI_SR[AHBGNT] bit is asserted immediately after any running IP Command has been finished (QSPI_SR[IP_ACC] is 0), the RX Buffer has been read out completely (QSPI_RBSR[RDBFL] equal to 0) or no DMA read is pending (QSPI_SR[RXDMA] equal to 0, allowing the master triggering AHB Commands to trigger AHB Commands as soon as possible without the need to wait for the RX Buffer readout to be finished.

21.8.4 Command Arbitration

In case of overlapping commands, the arbitration scheme is described in the following paragraphs under the assumption that the priority mechanism described in Exclusive Access to Serial Flash for AHB Commands is **not** used:

- During the execution of an IP Command, the running IP Command can't be terminated by issuing another IP Command or AHB Command. The QSPI_FR[IPIEF] flag is asserted when the host tries to write into the QSPI_IPCR register. When the host triggers an AHB Command (refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for details), this command is stalled until the currently running IP Command is finished.
- During the execution of an AHB Command, the running AHB Command can't be terminated by issuing an IP Command. The command is ignored and the QSPI_FR[IPAEF] flag is asserted. Refer to Flag Register (QuadSPI_FR) for the description of these flags.

  When another AHB Command is triggered the address of the memory mapped access is considered. If the requested address is currently read from the serial flash device, the running command is continued. If this is not the case the currently running command is terminated and another AHB Command related to the requested address is executed. Refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for further details.

In case of coinciding commands the IP Command is triggered and the AHB Command is stalled until the IP Command has been finished (QSPI_SR[IP_ACC] has been deasserted).

The IP Commands ignored in case of command collision will not result in the assertion of the QSPI_FR[TFF] flag.

21.8.5 Flash Device Selection

Regardless of the SFM Command (IP or AHB) the access mode is selected by specifying the 32 bit address value for the following SFM Command.

For IP Commands the access mode is selected with the address programmed into the QSPI_SFAR register. Refer to Serial Flash Address Register (QuadSPI_SFAR) for details.

For AHB Commands the access mode is determined by the memory mapped address which is accessed Refer to Flash memory mapped AMBA bus for details.

21.8.6 DMA Usage

For the complete description of the DMA module refer to the related DMA Controller chapter. In this paragraph only the details specific to the DMA usage related to the QuadSPI module are given.

21.8.6.1 DMA Usage in Normal Mode

21.8.6.1.1 Bandwidth considerations

Careful consideration of the throughput rate of the entire chain (serial flash -> AHB bus / IP Bus -> DMA controller) involved in the read data process is essential for proper operation. Such analysis must take into account not only the data rate provided by the serial flash but also the data rate of the AHB bus and the performance of the DMA controller in reading data from the RX buffer.

Two figures must match for proper operation, that means that the data rate provided by the serial flash device must not exceed the average RX Buffer readout data rate. Otherwise, the longer this state persists, a RX Buffer overflow will result.

AHB Bus Side (data read):

The total number of bus cycles for each DMA Minor Loop completion is added from the following components:

- Overhead for each minor loop, given by DMA controller: Assume 10 cycles
- Overhead due to clock domain crossing: Assume 2 cycles
- Number of bus clock cycles required for 8 bytes (64 bit read size): Assume 2 cycles (read/write sequence of DMA controller)

Note that the size of the minor loop is determined by the size of the QSPI_RBCT[WMRK] field, therefore the overhead given above distributes among (QSPI_RBCT[WMRK]+1)/2 read accesses of 64 bit each.

The following table gives some examples for typical use cases:

Table 21-23. Access Duration Examples - Bus Clock Side

QSPI_RBCT[WMRK]	Number of Bytes per DMA Loop [1]	Number of Bus Clock cycles for DMA Minor Loop	Time Duration of DMA Minor Loop for 120Mhz Bus clock Frequency
0	4	12+2 = 14	~117ns

Table continues on the next page...

Table 21-23. Access Duration Examples - Bus Clock Side (continued)

QSPI_RBCT[WMRK]	Number of Bytes per DMA Loop [1]	Number of Bus Clock cycles for DMA Minor Loop	Time Duration of DMA Minor Loop for 120Mhz Bus clock Frequency
1	8	12+2 = 14	~117ns
3	16	12+4 = 16	~133ns
7	32	12+8 = 20	~167ns
11	48	12+12 = 24	~200ns

1. DMA Loop means one Minor Loop Completion which is equivalent to one.

NOTE

The table figure represents ideal scenario, actual performance will depend on how the system is integrated.

Serial Flash Device Side (data read):

The number of serial flash cycles can be determined in the following way:

- Number of serial flash clock cycles required to read 4 bytes, corresponding to one RX Buffer entry (setup of command and address not considered): 2 cycles for Quad DDR mode instructions in Parallel Flash Mode, 2 cycles for Octal DDR mode (Hyperflash) in individual flash mode, 4 cycles for Quad (SDR) mode instruction in parallel flash mode, or Dual IO DDR mode instruction in parallel flash mode, 8 cycles for Quad Mode (SDR) instructions in Individual Flash Mode etc.
- Overhead due to clock domain crossing : 1 cycle.

The following table lists the number of clock cycles required to read the data from the serial flash corresponding to the different settings of the QSPI_RBCT[WMRK] field:

Table 21-24. Access Duration Examples - Serial Flash side

QSPI_RBCT[WMRK] setting	Num Bytes per DMA Loop [1]	Num SCKFx for 60MHz SCKFx			Time duration of Flash data readout for 60MHz SCKFx (~16.6ns period)		
		IFM [2] Quad	IFM Quad DDR	PFM [3] Quad DDR	IFM Quad	IFM Quad DDR	PFM Quad DDR
0	4	9	5	3	~150ns	~83ns	~50ns
1	8	17	9	5	~282ns	~150ns	~83ns
3	16	33	17	9	~548ns	~282ns	~150ns
7	32	65	33	17	~1079ns	~548ns	~282ns
11	48	97	49	25	~1610ns	~813ns	~415ns

1. DMA Loop means one Minor loop completion which is equivalent to one Major Loop iteration.
2. Individual flash mode.

3. Parallel flash mode.

Table 21-25. Access Duration Examples - Serial Flash side

QSPI_RBCT[WMRK] setting	Num Bytes per DMA Loop [1]	Num SCKFx for 60MHz SCKFx			Time duration of Flash data readout for 60MHz SCKFx (~16.6ns period)		
		IFM [2] Quad	IFM Quad DDR	PFM [3] Quad DDR	IFM Quad	IFM Quad DDR	PFM Quad DDR
0	4	9	5	3	~150ns	~83ns	~50ns
1	8	17	9	5	~282ns	~150ns	~83ns
3	16	33	17	9	~548ns	~282ns	~150ns
7	32	65	33	17	~1079ns	~548ns	~282ns
11	48	97	49	25	~1610ns	~813ns	~415ns

1. DMA Loop means one Minor loop completion which is equivalent to one Major Loop iteration.
2. Individual flash mode.
3. Parallel flash mode.

From the examples given in the two tables above, it can be seen that depending on the relationship between the Bus clock and Serial flash clock frequencies, there are settings possible where the serial flash provides the read data faster than the AHB bus can read out the RX buffer. In the above tables, it is the case of PFM Quad DDR mode with Watermark up to 3 and other cases. In these cases, the RX buffer data keeps accumulating over time and will eventually overflow. To avoid RX Buffer overflow, the data transaction size should be small enough.

A complementary example would be when the watermark is set to be too high. In such a case, the time taken by the DMA to read out the RX buffer entries should be smaller than the time taken by the controller to push in the remaining entries in the buffer.

NOTE

The tables mentioned above are only examples which must be correlated with the DMA in the system.

21.8.7 Parallel mode

QuadSPI can access two flashes in parallel. This increases the throughput of the QuadSPI by two times. Only read operations are allowed in parallel mode. In case a write transaction is initiated in parallel mode, QSPI_FR[IUEF] is set. When dual die flashes are accessed in parallel mode, it is mandatory for flash A1 to be of the same size as B1 and A2 to be of the same size as B2. The following figure shows how QuadSPI maps the incoming addresses to the different flashes connected on board.

Figure 21-6. Flash addressing

An example programming for parallel mode access is given below (flash sizes are assumed to be 256MB):

- QSPI_AMBA_BASE - 0x10000000
- QSPI_SFA1AD[TPADA1] - 0x20000000
- QSPI_SFA2AD[TPADA2] - 0x30000000
- QSPI_SFB1AD[TPADB1] - 0x40000000
- QSPI_SFB2AD[TPADB2] - 0x50000000

In order to access the first location of A1/B1 pair, the incoming address should be 0x10000000. QSPI_AMBA_BASE is subtracted from this address and the result is divided by two. Therefore, address provided to flash A1 and B1

Flash Address = (Memory mapped address - QSPI_AMBA_BASE)/2

For Memory Mapped address:

- 0x10000000, flash address: 0x0 (Or, the first address of flash A1 and B1)
- 0x10000004, flash address: 0x2
- 0x10000008, flash address: 0x4 etc.

Similarly, in order to access the first location of A2/B2 pair, the incoming address should be 0x30000000.

Flash Address = (Memory mapped address - SFA2AD)/2

For Memory Mapped address:

- 0x30000000, flash address: 0x0 (Or, the first address of flash A2 and B2)
- 0x30000004, flash address: 0x2
- 0x30000008, flash address: 0x4 etc.

Figure 21-7. Memory map - Serial and Parallel

21.9 Byte Ordering - Endianness

QuadSPI provides support for swapping the flash read/write data based on the configuration of the QSPI_MCR[END_CFG]. By default the data is always returned in 64 bit LE format on the AHB bus and 32 bit LE format on the IPS interface when read via the RX buffer and written in 32 bit LE format when written via the TX buffer.

The table(QSPI_MCR[END_CFG]) below shows the complete bit ordering. BE signifies Big Endian which means the high order bits of the associated data vectors are associated with low order address positions. LE signifies Little Endian which means the lower order bits of the associated data vectors are associated with low order address positions.Refer to figure Figure 21-5

Table 21-26. QSPI_MCR[END_CFG]

00	64 bit BE
01	32 bit LE
10	32 bit BE
11	64 bit LE

The tables below (Byte ordering configuration in AHB) and (Byte ordering configuration in IPS) show how this configuration is implemented in QSPI AHB and IPS interfaces respectively. B in the table signifies Byte and the index 1-8 refers to the byte position i.e. 1 refer to bits[7:0], 8 refer to bits[63:56] and so on.

Table 21-27. Byte ordering configuration in AHB

64 bit BE	B1	B2	B3	B4	B5	B6	B7	B8
64 bit LE	B8	B7	B6	B5	B4	B3	B2	B1
32 bit BE	B5	B6	B7	B8	B1	B2	B3	B4
32 bit LE	B4	B3	B2	B1	B8	B7	B6	B5

Table 21-28. Byte ordering configuration in IPS

32BE	B1	B2	B3	B4
32LE	B4	B3	B2	B1

The examples below show the byte ordering in 64 bit BE configuration for AHB Buffer and 32 bit BE for TX/RX Buffer:

21.9.1 Programming Flash Data

CPU write instructions to the QSPI_TBDR register like

- Write QSPI_TBDR -> 0x01_02_03_04
- Write QSPI_TBDR -> 0x05_06_07_08

result in the following content of the TX Buffer:

Table 21-29. Example of QuadSPI TX Buffer

TX Buffer Entry	Content
0	32'h01_02_03_04
1	32'h05_06_07_08

Programming the TX Buffer into the external serial flash device results in the following byte order to be sent to the serial flash:

- 01…02…03…04…05…06…07…08

21.9.2 Reading Flash Data into the RX Buffer

Reading the content from the same address provides the following sequence of bytes, identical to the write case:

- 01…02…03…04…05…06…07…08

This results in the RX Buffer filled with:

Table 21-30. Resulting RX Buffer Content

RX Buffer Entry	Content
0	32'h01_02_03_04
1	32'h05_06_07_08

21.9.2.1 Readout of the RX Buffer via QSPI_RBDRn

The RX Buffer content appears at CPU read access via the Peripheral bus interface in the following order:

- Read QSPI_RBDR0 <- 0x01_02_03_04
- Read QSPI_RBDR1 <- 0x05_06_07_08

21.9.2.2 Readout of the RX Buffer via ARDBn

The RX Buffer content appears at read access on the AMBA AHB interface at the QuadSPI module boundary:

- (1a): 32 Bit Access: Read QSPI_ARDB0 <- 0x01_02_03_04
- (2a): 32 Bit Access: Read QSPI_ARDB1 <- 0x05_06_07_08
- (1b/2b): 64 Bit Access: Read QSPI_ARDB0 <- 0x01_02_03_04_05_06_07_08

21.9.3 Reading Flash Data into the AHB Buffer

Reading the content from the same address as it was written to provides the following sequence of bytes, identical to the write case:

- 01…02…03…04…05…06…07…08

This results in the AHB Buffer filled with:

Table 21-31. Resulting AHB Buffer Content

AHB Buffer Entry	Content
0	64'h01_02_03_04_05_06_07_08

21.9.3.1 Readout of the AHB Buffer via Memory Mapped Read

The AHB Buffer content appears at read access on the AMBA AHB interface at the QuadSPI module boundary:

- (1a): 32 Bit Read Access: <- 0x01_02_03_04
- (2a): 32 Bit Read Access: <- 0x05_06_07_08
- (1/2): 64 Bit Read Access: <- 0x01_02_03_04_05_06_07_08

21.10 Driving Flash Control Signals in Single and Dual Mode

In single and dual mode the serial flash devices which can connect to the QuadSPI module expect additional control signals on the inputs which are connected to IOFA[3], IOFA[2], IOFB[3] and IOFB[2] in quad mode. For easy interfacing the outputs IOFA[3:2] for Flash A and the IOFB[3:2] for Flash B are driven to the logic state given by the configuration bits QSPI_MCR[ISD3FA], QSPI_MCR[ISD2FA], QSPI_MCR[ISD3FB] and QSPI_MCR[ISD2FB].

These outputs are driven all the time to the logic level programmed in the QSPI_MCR register except the time when quad commands of the serial flash are executed. Refer to the specification of the related serial flash device for details about the inactive level.

21.11 Serial Flash Devices

Several different vendors make flash devices with a QuadSPI interface. At present there is no set standard for the QuadSPI instruction set. Most common commands currently have the same instruction code for all vendors, however some commands are unique to specific vendors. Some example sequences are provided below.

21.11.1 Example Sequences

This section provides the example sequences of the QuadSPI module.

Table 21-32. Exit 4 x I/O Read Enhance Performance Mode (XIP) (Macronix) and Read Status

INSTR	PAD	OPERAND	COMMENT
CMD	0x0	0xEB	4xIO Read Command
ADDR	0x2	0x18	24 Bit address to be send on 4 pads
MODE	0x2	0x00	2 mode cycles (exit XIP)
DUMMY	0x0	0x04	4 dummy cycles
READ	0x2	0x08	Read 64 bits
CMD	0x0	0x05	Read Status register
READ	0x0	0x01	Status register data
STOP	0x0	0x00	STOP, Instruction over

21.11.1.1 Read Command (Spansion Hyperflash)

This section provides the read command sequences (Spansion Hyperflash) of the QuadSPI module.

Table 21-33. Read Command (Spansion Hyperflash)

INSTR	PAD	OPERAND	COMMENT
CMD_DDR	0x3	0xA0	Read command with continuous burst type
ADDR_DDR	0x3	0x18	24 bit row address

Table continues on the next page...

Table 21-33. Read Command (Spansion Hyperflash) (continued)

CADDR_DDR	0x3	0x10	16 bit column address with lower 3 bits valid rest 0
DUMMY	0x3	0x0F	15 dummy cycles
READ_DDR	0x3	0x4	32 bit data read on 8 pads
STOP	0x3	0x00	STOP, Instruction over

Hyperflash is a word addressable flash i.e. each address accesses a word wide (2 bytes) data value, the software should ensure that when Hyperflash is connected to the controller, the QSPI_SFACR [WA] bit must be set. If this bit is set the controller remaps a byte addressable access to a word addressable access.

21.11.1.2 Read Status Register(Spansion Hyperflash)

This section provides the read status register of the QuadSPI module.

Table 21-34. Read Status register (Spansion Hyperflash)

INSTR	INSTR SEQUENCE	PAD	OPERAND	COMMENT
CMD_DDR	Read Pre Comman	0x3	0x00	Write command with wrapped burst type.
ADDR_DDR		0x3	0x18	24 bit row address(0000AAh)
CADDR_DDR		0x3	0x10	16 bit column address with lower 3 bits valid rest 0(0005h),treated as command
CMD_DDR		0x3	0x00	Write command with wrapped burst type
CMD_DDR		0x3	0x70	Write data to be sent to flash as pre-command
CMD_DDR	Command phase (fourth/final chip select phase)	0x3	0xA0	Read command with continuous burst type
ADDR_DDR		0x3	0x18	24 bit row address
CADDR_DDR		0x3	0x10	16 bit column address with lower 3 bits valid rest 0
DUMMY		0x3	0x0F	15 dummy cycles
READ_DDR		0x3	0x4	32 bit data read on 8 pads
STOP		0x3	0x00	STOP, Instruction over

21.11.1.3 Word Program (Spansion Hyperflash)

This section provides the Word Program (Spansion Hyperflash) of the QuadSPI module.

Table 21-35. Word Program (Spansion Hyperflash)

INSTR	INSTR SEQUENCE	PAD	OPERAND	COMMENT
CMD_DDR	Unlock Sequence 1 (first chip select phase)	0x3	0x00	Write command with wrapped burst type
CMD_DDR		0x3	0x00	8 bit address 00h treated as command
CMD_DDR		0x3	0x00	8 bit address 00h treated as command
CMD_DDR		0x3	0xAA	8 bit address AAh treated as command
CADDR_DDR		0x3	0x10	16 bit column address with lower 3 bits valid rest 0(0005h),treated as command
CMD_DDR		0x3	0x00	Write command with wrapped burst type.
CMD_DDR		0x3	0xAA	Write data to be sent to flash as pre-command.
CMD_DDR	Unlock Sequence 2 (second chip select phase)	0x3	0x00	Write command with wrapped burst type
CMD_DDR		0x3	0x00	8 bit address 00h treated as command
CMD_DDR		0x3	0x00	8 bit address 00h treated as command
CMD_DDR		0x3	0x55	8 bit address 55h treated as command
CADDR_DDR		0x3	0x10	16 bit column address with lower 3 bits valid rest 0(0002h),treated as command
CMD_DDR		0x3	0x00	Write command with wrapped burst type
CMD_DDR		0x3	0x55	Write data to be sent to flash as pre-command
CMD_DDR	Program setup phase (third chip select phase)	0x3	0x00	Write command with wrapped burst type
CMD_DDR		0x3	0x00	8 bit address 00h treated as command
CMD_DDR		0x3	0x00	8 bit address 00h treated as command
CMD_DDR		0x3	0xAA	8 bit address AAh treated as command
CADDR_DDR		0x3	0x10	16 bit column address with lower 3 bits valid rest 0(0005h),treated as command

Table continues on the next page...

Table 21-35. Word Program (Spansion Hyperflash) (continued)

CMD_DDR		0x3	0x00	Write command with wrapped burst type
CMD_DDR		0x3	0xA0	Write data to be sent to flash as pre-command
CMD_DDR	Command phase (fourth/final chip select phase)	0x3	0x00	Write command with wrapped burst type
ADDR_DDR		0x3	0x18	24 bit row address
CADDR_DDR		0x3	0x10	16 bit column address with lower 3 bits valid rest 0
WRITE_DDR		0x3	0x2	2 bytes data written on 8 pads (D1D2)
STOP		0x3	0x00	STOP, Instruction over

21.11.1.4 Fast Read Sequence (Macronix/Numonyx/Spansion/ Winbond)

The following table shows the fast read sequence for Macronix/Numonyx/Spansion/ Winbond flashes.

Table 21-36. Fast Read sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x0B	Fast Read command = 0x0B
ADDR	0x0	0x18	24 Addr bits to be sent on one pad
DUMMY	0x0	0x08	8 Dummy cycles
READ	0x0	0x04	Read 32 Bits on one pad
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

21.11.1.5 Fast Dual I/O DT Read Sequence (Macronix)

The following table shows the Fast Dual I/O DT read sequence for Macronix flashes.

Table 21-37. Fast Dual I/O DT Read sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xBD	Fast Dual I/O DT read command = 0xBD
ADDR_DDR	0x1	0x18	24 Addr bits to be sent on 2 pads in DDR mode
MODE4_DDR	0x1	0x00	P2=P0 or P3=P1 is necessary. Refer to Macronix datasheet for details. One clock cycle for mode.

Table continues on the next page...

Table 21-37. Fast Dual I/O DT Read sequence (continued)

Instruction	Pad	Operand	Comment
DUMMY	0x0	0x06	6 Dummy cycles
READ_DDR	0x1	0x04	Read 32 Bits on 2 pads in DDR mode
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

21.11.1.6 Fast Read Quad Output (Winbond)

The following table shows the Fast read quad output sequence for Winbond memories

Table 21-38. Fast Read Quad output sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x6B	Fast read quad output command = 0x6B
ADDR	0x0	0x18	24 Addr bits to be sent on 1 pad
DUMMY	0x0	0x08	8 Dummy cycles
READ	0x2	0x04	Read 32 Bits on 4 pads
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

21.11.1.7 4 x I/O Read Enhance Performance Mode (XIP) (Macronix)

The following table shows the 4 x I/O Read Enhance Performance Mode for Macronix flashes. The enhanced performance mode is also known as XIP mode.

Table 21-39. Fast Read Quad output sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xEB	4xI/O Read command = 0xEB
ADDR	0x2	0x18	24 Addr bits to be sent on 4 pads
MODE	0x2	0xA5	2 mode cycles
DUMMY	0x0	0x04	4 Dummy cycles
READ	0x2	0x04	Read 32 Bits on 4 pads
JMP_ON_CS	0x0	0x01	Jump to instruction 1 (ADDR)

When in XIP mode the software should ensure that all the flashes connected to the controller are in XIP mode. As a part of initializing the controller, all the flashes may be enabled with XIP by carrying out dummy reads.

21.11.1.8 Dual Command Page Program (Numonyx)

The following table shows the Dual command page program sequence for Numonyx flashes.

Table 21-40. Dual Command Page Program sequence

Instruction	Pad	Operand	Comment
CMD	0x1	0x02	Dual command page program = 0x02 on 2 pads
ADDR	0x1	0x18	24 Addr bits to be sent on 2 pads
WRITE	0x1	0x20	Write 32 Bytes on 2 pads
STOP	0x0	0x00	STOP, Instruction over

21.11.1.9 Sector Erase (Macronix/Spansion/Numonyx)

The following table shows the Sector erase sequence for Macronix/Spansion/Numonyx flashes

Table 21-41. Sector Erase sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xD8	Sector erase command = 0xD8
ADDR	0x0	0x18	24 Addr bits to be sent on 1 pad
STOP	0x0	0x00	STOP, Instruction over

21.11.1.10 Read Status Register (Macronix/Spansion/Numonyx/Winbond)

The following table shows the Read status register sequence for Macronix/Spansion/Numonyx/Winbond flashes.

Table 21-42. Read Status Register Sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x05	Read status register command = 0x05
READ	0x0	0x01	Read status register data
STOP	0x0	0x00	STOP, Instruction over

21.11.2 Dual Die Flashes

Certain serial flash vendors provide dual-die packages which are essentially two devices (dies) stacked within the same package to increase the memory capacity of a single package. These two devices within a package share the same data and clock pins, but have individual Chip Selects. QuadSPI controller provides support for two dual-die packages to be connected simultaneously. The figure below shows the two dual-die packages and the naming conventions used in this document. For simplicity, the data pins are shown to be unidirectional.

Figure 21-8. Dual-die support

Since the two devices within one package share the same i/o pads, they cannot function in parallel mode. Software should ensure that when QuadSPI is configured in parallel mode the two selected flash devices are from different dual-die packages.

21.11.3 Boot initialization sequence

The following are the recommended sequence of steps for booting from QuadSPI:

- System out of reset and flash available (300us)
- Clocks still at very low frequency. Clock tree configured, I/O pins configured. First request sent to QuadSPI for address 0x0 of flash.

- The reset command sequence in QuadSPI has 0x03 (basic read command) which is applicable to all flashes at < 50MHz serial flash clock
- The first few bytes of data is read from the flash which contains the following information:
  - The total sizes of all the flashes connected on board
  - Whether DDR mode supported
  - Frequency of DDR operation
  - Continuous mode entry sequence
  - 24bit or 32bit addressing (assuming 24bit for first accesses)
- All the serial flashes are configured
  - Quad Mode enabled
  - Dummy reads to enter into XIP
- QuadSPI is configured
  - Parallel enable set
  - LUT configured for highest performance reads
  - DDR mode enabled (if applicable)
  - Buffers configured
- Serial flash clock frequency increased.
- Boot reads happen in parallel, DDR enabled, quad output mode @66MHz.

21.11.4 Serial Flash Clock Frequency Limitations

Certain commands of some serial flash devices are limited in the frequency applied to the serial flash device on command execution. In order to support these commands without having to recalibrate the module clocks, the the serial flash device clock can be divided by 2 (half speed) by setting the QSPI_SMPR[HSENA] bit. The SCLK will return to full speed once the the QSPI_SMPR[HSENA] bit is cleared.

21.12 Sampling of Serial Flash Input Data

21.12.1 Basic Description

QuadSPI is used to read data from the serial flash device. Depending on the actual implementation, there is a delay between the internal clocking in the QuadSPI module and the external serial flash device. Refer to the following figure for an overview of this scheme.

Figure 21-9. Serial Flash Sampling Clock Overview

The rising edge of the internal reference clock is taken as timing reference for the data output of the serial flash. After a time of $t_{Del,total}$ the data arrives at the internal sampling stage of the QuadSPI module. According to the Serial Flash Sampling Clock Overview figure, the following parts of the delay chain contribute to $t_{Del,total}$:

1. Output delay of the serial flash clock output of the device containing the QuadSPI module
2. Wire delay of application/PCB from the device containing the QuadSPI module to the external serial flash device
3. Clock to data out delay of the external serial flash device, including input and output delays
4. Wire delay of application/PCB from the external serial flash device to the device containing the QuadSPI module
5. Device delay corresponding to the input data

NOTE

The amount of total delay $t_{Del,total}$ is specific to the characteristics of the actual implementation. Also, the serial flash device clock (SCK) is inverted with respect to the QuadSPI internal reference clock.

21.12.2 Supported read modes

The table below provides an overview of QuadSPI read modes.

Table 21-43. QuadSPI read modes

Read modes			QuadSPI_MCR[DDR_EN]	QuadSPI_MCR[DQS_EN]	Data learning support	For more information
SDR mode	Internal sampling (N/1, I/1)		0	0	No	Internal sampling
	DQS sampling method	Internally generated DQS	0	1	No	Internally generated DQS
		External DQS	0	1	No	External DQS
DDR mode	Internal sampling (4x method)		1	0	Yes	Internal sampling (4x sampling method)
	DQS sampling method	Internally generated DQS	1	1	Yes	Internally generated DQS
		External DQS	1	1	Yes	External DQS

21.12.2.1 SDR mode

Most flash memories operate in single data rate (SDR) mode. In SDR mode, the data is transferred only on one edge of the clock signal. The SDR serial flash memories sample the incoming data on the rising edge of serial flash clock and drive the output data on the falling edge of the serial flash clock.

21.12.2.1.1 Internal sampling

QuadSPI uses different edges of the internal reference clock for sampling the input data in SDR mode.

Figure 21-10. Internal sampling in SDR mode

The possible points in time for sampling incoming data are denoted as N/1, I/1, N/2 and I/2 above. The sampling point relevant for the internal sampling is configured in the QSPI_SMPR register. Refer to Sampling Register (QuadSPI_SMPR) for details. The following table gives an overview of the available configurations for the commands running at regular (full) speed:

Table 21-44. Sampling Configuration

Sampling Point	Description	Delay [FSDLY] [HSDLY]	Phase [FSPHS] [HSPHS]	QSPI_SMPR for Full Speed Setting[1]
N/1	sampling with non-inverted clock, 1 sample delay	0	0	0x0000000x
I/1	sampling with inverted clock, 1 sample delay	0	1	0x0000002x
N/2	sampling with non-inverted clock, 2 samples delay	1	0	0x0000004x
I/2	sampling with inverted clock, 2 samples delay	1	1	0x0000006x

1. 'x' is not considered here

Depending on the actual delay and the serial flash clock frequency, the appropriate sampling point can be chosen. The following remarks should be considered when selecting the appropriate setting:

- Theoretically there should be two settings possible to capture the correct data, since the serial flash output is valid for 1 clock cycle, disregarding rise and fall times and timing uncertainties.

- Depending on the timing uncertainties, it may turn out in actual applications that only one possible sample position remains. This is subject to careful consideration depending on the actual implementation.
- The delay $t_{Del,total}$ is an absolute size to shift the point in time when the serial flash data get valid at the QuadSPI input.
- For decreasing frequency of the serial flash clock the distance between the edges increases. So for large differences in the frequency the required setting may change.
- For commands running at half of the regular serial flash clock (QSPI_SMPR[HSENA] bit set) the sampling point must be figured separately to allow for the compensation of the absolute shift in time with respect to the sample-relative setting in the QSPI_SPMPR register.

21.12.2.1.2 DQS sampling method

Data sampling in SDR mode can be supported using the DQS sampling method. Refer to Data Strobe (DQS) sampling method for more details.

21.12.2.2 DDR Mode

The increasing requirement of improved throughput has introduced the double data rate (DDR) mode. In DDR mode, the data is transferred on both the rising and falling edges of the serial flash clock. The DDR serial flashes sample as well as drive the data on both rising and falling edges of serial flash clock.

21.12.2.2.1 Internal sampling (4x sampling method)

When the serial flash memories function in DDR mode, the data is valid for only half a clock cycle. This, along with the fact that the time for which the data is actually valid is smaller than half a clock cycle, requires that we provide closely spaced sampling points. The QuadSPI module provides a mechanism to sample the incoming data at multiple sampling points provided by a 4x serial flash clock in DDR mode. The figure below shows the different sampling points as configured by QuadSPI_SMPR[DDRSMP]. The FSDLY/FSPHS and HSDLY/HSPHS bits are ignored for DDR instructions.

Figure 21-11. 4x sampling edges in DDR mode

Software should ensure that the correct sampling value is configured in the QuadSPI_SMPR[DDRSMP] register.

NOTE

Higher frequency can be achieved using data learning. For details, refer to Data Learning.

21.12.2.2.2 DQS sampling method

Data sampling in DDR mode can be supported using DQS method. Refer to Data Strobe (DQS) sampling method for more details.

A higher frequency can be achieved using Data learning. For details refer to Data Learning.

21.12.3 Data Strobe (DQS) sampling method

21.12.3.1 Basic Description

In DQS mode, the data strobe signal (DQS/RWDS) is used to sample the read data. Here, both DQS and the data sent by the flash move in the same direction, so it is relatively easier to achieve at higher frequencies.

When using DQS for SDR reads, QuadSPI internally samples the incoming data on rising edge of the strobe signal. FSPHS=0 only.

The figure below shows sampling read data in SDR mode using DQS.

Figure 21-12. Data Strobe functionality in SDR mode

When using DQS for DDR reads, QuadSPI internally samples the incoming data on both the edges of the strobe signal. Refer to the figure below for more detail.

Figure 21-13. Data Strobe functionality in DDR mode

21.12.3.2 Internally generated DQS

In this mode, the internal reference clock is fed as a strobe to QuadSPI for read data sampling. The data strobe must be generated in a way such that the data is correctly sampled by the QuadSPI module. This internally generated data strobe signal can be used by QuadSPI to capture the data in:

- SDR mode
- DDR mode

Refer to QuadSPI chip-specific information for additional details about internally generated DQS.

21.12.3.3 External DQS

In serial flash memories supporting DQS, the data strobe signal is an output from the flash device that indicates when data is being transferred from the flash to the host controller. The data is then captured by the controller on:

- Only one edge (either rising or falling edge) of DQS signal in SDR mode
- Both rising/falling edge of the DQS signal in DDR mode

21.12.4 Data Learning

21.12.4.1 Basic description

"Data learning" is used to manage varying data valid windows from flash memory as well as any variations in the chip based on PVT conditions in DDR mode. QuadSPI provides this feature via the DATA_LEARN instruction for all flash memories, irrespective of whether the flash supports it. The DATA_LEARN instruction accepts an operand that defines the number of bits of the known pattern for which the data learning has to be done. The known pattern is provided in the QSPI_DLPR register.

QuadSPI supports the data learning methods below:

- 4x sampling method
- Data Strobe (DQS) sampling method

The following sections explain data learning in greater detail.

21.12.4.2 4x sampling method

Automatic data learning is supported in 4x sampling mode. To start data learning:

1. Certain flash memories provide data learning as a feature for DDR data reads.

   For flash memories that support data learning, configure a known pattern in the Data Learn register inside flash. The pattern should be chosen to have multiple low and high transitions in the data bus. Set up a QuadSPI DDR data read sequence, inserting a DATA_LEARN instruction before the READ_DDR instruction. The flash will itself return the data learning pattern in between dummy cycles in every read command. Each I/O will output the same DLP value for every clock edge.

   For flash memories that *do not* support data learning, configure a known location in flash memory with a known pattern. The pattern should not have 0x00 or 0xFF in it. The pattern should be chosen to have multiple low and high transitions in the data bus. Set up a QuadSPI DDR data read sequence, inserting a DATA_LEARN instruction before the READ_DDR instruction. The address in the QuadSPI_SFAR register should point to the known location with the pattern. Configure the known pattern in the QuadSPI_DLPR [DLPV] field. Refer to Table 21-45 for details on how the data has to be ordered in memory for correct operation.

2. Select a sampling point using QuadSPI_SMPR[DDRSMP].

3. Initiate the read via a peripheral transaction.
4. QuadSPI reads the data from the flash. It then encounters the DATA_LEARN instruction and samples the incoming data on all 8 possible sampling points for all the data bus bits.
5. The sample point at which data matches the known pattern is reported in QuadSPI_SR[DLPSMP].
6. If data doesn't match at any sample point, QuadSPI_FR[DLPFF] is set.
7. If the correct sampling point is located, QuadSPI uses the new sampling point to sample the read data following the DATA_LEARN instruction.

This feature may be used to auto-calibrate the sampling point for DDR reads. This auto-calibration may be triggered at fixed intervals, or depending on a change in PVT conditions.

The following figure shows data learning with 4x DDR sampling:

Figure 21-14. Data learning with 4x sampling method

21.12.4.3 Data Strobe (DQS) sampling method

Semi-automatic data learning is supported in DQS sampling method. To start data learning:

1. Certain flash memories provide data learning as a feature for DDR data reads.

   For flash memories that support data learning, configure a known pattern in the Data Learn register inside flash. The pattern should be chosen to have multiple low and high transitions in the data bus. Set up a QuadSPI DDR data read sequence, inserting a DATA_LEARN instruction before the READ_DDR instruction. The flash will itself return the data learning pattern in between dummy cycles in every read command. Each I/O will output the same DLP value for every clock edge.

For flash memories that *do not* support data learning, configure a known location in flash memory with a known pattern. The pattern should not have 0x00 or 0xFF in it. The pattern should be chosen to have multiple low and high transitions in the data bus. Set up a QuadSPI DDR data read sequence, inserting a DATA_LEARN instruction before the READ_DDR instruction. The address in the QSPI_SFAR register should point to the known location with the pattern. Configure the known pattern in the QSPI_DLPR[DLPV] field. Refer to Table 21-45 for details on how the data has to be ordered in memory for correct operation.

2. Select a sampling point. See chip-specific QuadSPI information for selection of the sampling point.
3. Initiate the read via a peripheral transaction.
4. QuadSPI reads the data from the flash. It then encounters the DATA_LEARN instruction and samples the incoming data on both edges (rising and falling) of the DQS.
5. If the data from the flash doesn't match the data learning pattern, the QuadSPI_FR[DLPFF] flag is set.
6. No sampling point is reported automatically by the QuadSPI module.
7. These steps (2-7) are repeated with varying VT conditions for a particular process until the QuadSPI_FR[DLPFF] is set. The sampling points where no QuadSPI_FR[DLPFF] is set signify valid setting.
8. In the case of multiple valid settings, software should choose the middle point.
9. The sampling point should be fixed with the above chosen point for the next READ transactions.

NOTE

It must be ensured that QuadSPI is not accessed during calibration.

NOTE

Ensure that there is no latency in between DQS edges while doing data learning using DQS sampling method.

This feature may be used to auto-calibrate the sampling point for DDR reads. This auto-calibration may be triggered at fixed intervals, or depending on a change in PVT conditions.

The following figure shows data learning in DQS mode:

Figure 21-15. Data learning with DQS sampling method

21.12.4.4 Example: programming the data learning pattern in flash memory

If QSPI_DLPR[DLPV] is set to 0x43, QuadSPI tries to match this sample on every incoming data line for single, dual, quad, or octal configurations. The table below shows the data programmed in flash with endianness taken into consideration.

Table 21-45. Programming the Data Learning Pattern in flash memory

Endianness	Single IO	Dual IO	Quad IO	Octal IO
BE	0x43	0x300F	0x0F0000FF	0x00FF00000000FFFF
LE	0x43	0X0F30	0XFF00000F	0XFFFF0000000000FF

21.13 Data Input Hold Requirement of Flash

In DDR mode, the data is valid only for half clock cycle. It is difficult to meet the data input hold time requirement of flash. QuadSPI internally delays the data sent to flash so that it will be easy to meet the hold requirement. The QSPI_FLSHCR[TDH] is used for this purpose. If QSPI_FLSHCR[TDH] is configured to 0, the data sent to flash is aligned

with the internal reference clock of QuadSPI, if it is 1, then, the data is aligned to 2x internal reference half clock, if it is 2, then, the data is aligned to 4x internal reference half clock.

Figure 21-16. Data Hold

Chapter 22
SATA 3.0

22.1 The SATA module as implemented on the chip

This section provides details about how the SATA module is integrated into this chip.

The chip implements two SATA modules, SATA1 and SATA2.

Table 22-1. SATA module as implemented on chip

Module name	Module base address
SATA1	0320_0000h
SATA2	0321_0000h

22.2 Advanced host controller interface overview

The SATA 3.0 advanced host controller interface (AHCI) is a high-performance SATA solution that delivers comprehensive and fully-compliant generation 3 (1.5 Gb/s - 6.0 Gb/s) serial ATA capabilities, in accordance with the serial ATA revision 3.0 of Serial ATA International Organization.

SATA consists of an AHCI command layer (with a descriptor-based DMA controller) designed to operate in a system that supports command queuing. It supports FIS-based switching for improved performance in systems that employ port multipliers. Under the command layer, SATA contains the transport and link layers, as outlined in the SATA standard.

Figure 22-1. SATA controller layer overview

22.3 SATA features summary

The SATA controller includes the following features:

- Complies with the serial ATA 3.0 specification and the AHCI 1.3.1 specification
- Contains a high-speed descriptor-based DMA controller
- Supports the following:
  - Speeds of 1.5 Gb/s (first-generation SATA), 3 Gb/s (second-generation SATA), and 6 Gb/s (third-generation SATA)
  - FIS-based switching
  - Native command queuing (NCQ) commands
  - Port multiplier operation
  - Asynchronous notification
  - SATA Vendor BIST mode

22.4 SATA AHCI register descriptions

The SATA controller is compliant to AHCI 1.3.1 programming model. The SATA memory map includes only those registers which have implementation differences from AHCI Specification. For complete registers list, refer to the SATA AHCI Specification.

The registers are divided into global registers and port control registers, as shown in the table below.

Table 22-2. Register group

Start address	End address	Description
00h	2Bh	Generic host controller
A0h	FFh	Vendor-specific registers
100h	17Fh	Port 0 port control registers

22.4.1 SATA memory map

SATA base address: 320_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	HBA capabilities register (CAP)	32	RO	E537_FF80h
4h	Global HBA control register (GHC)	32	RW	8000_0000h
10h	AHCI version register (VS)	32	RO	0001_0301h
14h	Command completion coalescing control register (CCC_CTL)	32	RW	0001_0110h
24h	HBA capabilities extended register (CAP2)	32	RO	0000_000Ch
A4h	Port config register (PCFG)	32	RW	0032_2002h
A8h	Port Phy1Cfg register (PPCFG)	32	RW	A001_FFFEh
ACh	Port Phy2Cfg register (PP2C)	32	RW	2818_4D1Bh
B0h	Port Phy3Cfg register (PP3C)	32	RW	0E08_1906h
B4h	Port Phy4Cfg register (PP4C)	32	RW	064A_0813h
B8h	Port Phy5Cfg register (PP5C)	32	RW	3FFC_96A4h
BCh	AXI cache control register (AXICC)	32	RW	0010_0010h
C0h	Port AXICfg register (PAXIC)	32	RW	0041_0102h
C4h	AXI PROT control register (AXIPC)	32	RW	0000_0000h
C8h	Port TransCfg register (PTC)	32	RW	0800_0020h
D0h	Port LinkCfg register (PLC)	32	RW	3800_FF34h
D4h	Port LinkCfg1 register (PLC1)	32	RW	0000_0000h
D8h	Port LinkCfg2 register (PLC2)	32	RW	0000_0000h
E0h	Port LinkStatus1 register (PLS1)	32	RW	0000_0000h
E4h	Port CmdConfig register (PCMDC)	32	RW	0000_0000h
E8h	Port PhyControl status register (PPCS)	32	RW	F800_0002h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
F0h	Timer control register (TCR)	32	RW	0000_0100h
100h	Port x command list base address register (PxCLB)	32	RW	0000_0000h
104h	Port x command list base address upper 32-bit register (PxCLBU)	32	RW	0000_0000h
108h	Port x FIS base address register (PxFB)	32	RW	0000_0000h
10Ch	Port x FIS base address upper 32-bit register (PxFBU)	32	RW	0000_0000h
110h	Port x interrupt status register (PxIS)	32	RW	0000_0000h
118h	Port x command and status register (PxCMD)	32	RW	0040_0000h
128h	Port x SATA status register (PxSSTS)	32	RO	0000_0000h
12Ch	Port x SATA control register (PxSCTL)	32	RW	0000_0300h
130h	Port x SATA error register (PxSERR)	32	RW	0000_0000h
138h	Port x command issue register (PxCI)	32	RW	0000_0000h
140h	Port x FIS-based switching control register (PxFBS)	32	RW	0000_4000h
170h	Port 0 BIST error register (PBERR)	32	RW	0000_0003h

22.4.2 HBA capabilities register (CAP)

22.4.2.1 Offset

Register	Offset
CAP	0h

22.4.2.2 Function

This register indicates basic capabilities of HBA to the driver software.

22.4.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	S64A	SNCQ	SSNTF	SMPS	SSS	SALP	SAL	SCLO	ISS				Reserved	SAM	SPM	FBSS
W																
Reset	1	1	1	0	0	1	0	1	0	0	1	1	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PMD	SSC	PSC	NCS					CCCS	EMS	SXS	NP				
W																
Reset	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

22.4.2.4 Fields

Field	Function
31 S64A	Supports 64-bit addressing Indicates whether the HBA can access 64-bit data structures. 0b - Tthe upper 32-bit bits of the port DMA descriptor, the PRD base, and each PRD entry are read-only and treated as 0 by the HBA. 1b - The HBA makes the upper 32-bit bits of the port DMA descriptor, the PRD base, and each PRD entry read/write.
30 SNCQ	Supports native command queuing Indicates whether the HBA supports SATA native command queuing. 0b - The native command queuing is not supported and software should not issue any native command queuing commands. 1b - The HBA handles DMA Setup FISes natively and the auto-activate optimization through that FIS.
29 SSNTF	Supports SNotification register 0b - The HBA does not support the PxSNTF[SNotification] register and its associated functionality. Refer to Section 10.11.1 of the SATA AHCI Specification for more details. Asynchronous notification with a directly attached device is always supported. 1b - The HBA supports the PxSNTF[SNotification] register and its associated functionality.
28 SMPS	Supports mechanical presence switch 0b - This function is not supported. This value is loaded by the BIOS prior to OS initialization. 1b - The HBA supports mechanical presence switches on its ports for use in hot plug operations.
27 SSS	Supports staggered spin-up 0b - This function is not supported. This value is loaded by the BIOS prior to OS initiallization. 1b - The HBA supports staggered spin-up on its ports, for use in balancing power spikes.
26 SALP	Supports aggressive link power management 0b - This function is not supported and software treats the PxCMD[ALPE] and PxCMD[ASP] bits as reserved. 1b - The HBA supports auto-generating link requests to the partial or slumber states if there are no commands to process.

Table continues on the next page...

Field	Function
25 SAL	Supports activity LED 0b - This function is not supported. 1b - The HBA supports a single activity indication output pin. This pin can be connected to an LED on the platform to indicate device activity on any drive.
24 SCLO	Supports command list override 0b - The HBA is not capable of clearing the PxTFD[BSY] and PxTFD[DRQ] bits in the status register to issue a software reset if these bits are still set from a previous operation. 1b - The HBA supports the PxCMD[CLO] bit and its associated function.
23-20 ISS	Interface speed support This bit indicates the maximum speed that the HBA can support on its ports. These encodings match the system software programmable PxSCTL[DET] and PxSCTL[SPD] bits. The settings not shown below are reserved. 0001b - Gen 1 (1.5 Gbps) 0010b - Gen 2 (3 Gbps) 0011b - Gen 3 (6 Gbps)
19 —	Reserved
18 SAM	Supports AHCI mode only The SATA controller may optionally support AHCI access mechanisms only. 0b - In addition to the native AHCI mechanism (through ABAR), the SATA controller implements a legacy, task-file based register interface such as SFF-8038i. 1b - The SATA controller does not implement a legacy, task-file based register interface.
17 SPM	Supports port multiplier This bit indicates whether the HBA supports a port multiplier. 0b - A port multiplier is not supported and a port multiplier may not be attached to this HBA. 1b - A port multiplier using command-based switching is supported and FIS-based switching may be supported.
16 FBSS	FIS-based switching supported This bit must be set if the SPM bit is set. 0b - The HBA does not support FIS-based switching. 1b - The HBA supports port multiplier FIS-based switching.
15 PMD	PIO multiple DRQ block In AHCI 1.2 HBAs, this bit is set to 1. 0b - The HBA only supports single DRQ block data transfers for the PIO command protocol. 1b - The HBA supports multiple DRQ block data transfers for the PIO command protocol.
14 SSC	Slumber state capable This bit indicates whether the HBA supports transitions to the slumber state. 0b - Software must neither allow the HBA to initiate transitions to the slumber state through agressive link power management nor the PxCMD[ICC] and PxSCTL[IPM] bits in each port must be programmed to disallow device initiated slumber requests. 1b - The HBA and device initiated slumber requests are supported.
13 PSC	Partial state capable This bit indicates whether the HBA can support transitions to the partial state. 0b - Software must neither allow the HBA to initiate transitions to the partial state through agressive link power management nor the PxCMD[ICC] and PxSCTL[IPM] bits in each port must be programmed to disallow device initiated partial requests. 1b - The HBA and device initiated partial requests are supported.
12-8 NCS	Number of command slots

Table continues on the next page...

Field	Function
	This bit provides value indicating the number of command slots per port supported by this HBA. A minimum of 1 and maximum of 32 slots per port can be supported. The same number of command slots is available on each implemented port.
7 CCCS	Command completion coalescing supported 0b - The HBA does not support command completion coalescing and the CCC_CTL and CCC_PORTS global HBA registers are not implemented. 1b - The HBA supports command completion coalescing as defined in Section 11 of the SATA AHCI Specification. When command completion coalescing is supported, the HBA has implemented the CCC_CTL and the CCC_PORTS global HBA registers.
6 EMS	Enclosure management supported 0b - The HBA does not support enclosure management and the EM_LOC and EM_CTL global HBA registers are not implemented. 1b - The HBA supports enclosure management as defined in Section 12 of the SATA AHCI Specification. When enclosure management is supported, the HBA has implemented the EM_LOC and EM_CTL global HBA registers.
5 SXS	Supports external SATA If this bit is set to 1, software may refer to the PxCMD[ESP] bit to determine whether a specific port has its signal connector externally accessible as a signal only connector (i.e. power is not part of that connector). 0b - The HBA has no SATA port that has a signal only connector externally accessible. 1b - This bit indicates that the HBA has one or more SATA ports that has a signal only connector that is externally accessible (for example, eSATA connector).
4-0 NP	Number of ports This bit provides value indicating the maximum number of ports supported by the HBA silicon. A maximum of 32 ports can be supported. A value of 0h, indicating one port, is the minimum requirement. **NOTE:** The number of ports indicated in this BIT may be more than the number of ports indicated in the PI register.

22.4.3 Global HBA control register (GHC)

22.4.3.1 Offset

Register	Offset
GHC	4h

22.4.3.2 Function

This register controls various global actions of the HBA.

22.4.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	AE	Reserved														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													MRSM	I E	HR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.3.4 Fields

Field	Function
31 AE	AHCI enable This is used by the HBA that supports both legacy mechanisms (such as, SFF-8038i) and AHCI to know when the HBA is running under the AHCI driver. When set, software only communicates with the HBA using AHCI. When cleared, software only communicates with the HBA using legacy mechanisms. Software sets this bit to 1 before accessing other AHCI registers. The implementation of this bit is dependent upon the value of the CAP[SAM] bit. If CAP[SAM] is 0, then GHC[AE] is read-write with a reset value of 0. If CAP[SAM] is 1, then GHC[AE] is read-only with a reset value of 1. 0b - FISes are not posted to memory and no commands are sent through AHCI mechanisms. 1b - Communication to the HBA happens through AHCI mechanisms.
30-3 —	Reserved
2 MRSM	MSI revert to single message This bit is set to 1 only when the following conditions hold: • MC[MSIE] = 1 (MSI is enabled) • MC[MMC] > 0 (multiple messages requested) • MC[MME] > 0 (more than one message allocated) • MC[MME] != MC[MMC] (messages allocated not equal to number requested) When this bit is set to 1, single MSI mode operation is in use and software is responsible for clearing bits in the IS register to clear interrupts. This bit is cleared to ‘0’ by hardware when any of the above four conditions stated is false. This bit is also cleared to 0 when MC[MSIE] = 1 and MC[MME] = 0. In this case, the hardware has been programmed to use single MSI mode, and is not reverting to that mode. 0b - The HBA has not reverted to single MSI mode (i.e. hardware is already in single MSI mode, software has allocated the number of messages requested, or hardware is sharing interrupt vectors if MC[MME] < MC[MMC]). The HBA may revert to single MSI mode when the number of vectors allocated by the host is less than the number requested. 1b - The HBA requested more than one MSI vector but has reverted to using the first vector only.
1 IE	Interrupt enable This bit enables interrupts from the HBA.

Table continues on the next page...

Field	Function
	0b - All interrupt sources from all ports are disabled (default) 1b - Interrupts are enabled.
0 HR	HBA reset When set by software, this bit causes an internal reset of the HBA. All state machines that relate to data transfers and queuing return to an idle condition, and all ports are re-initialized through COMRESET (if staggered spin-up is not supported). If staggered spin-up is supported, then it is the responsibility of software to spin-up each port after the reset has completed. When the HBA has performed the reset action, it resets this bit to 0. A software write of 0 has no effect.

22.4.4 AHCI version register (VS)

22.4.4.1 Offset

Register	Offset
VS	10h

22.4.4.2 Function

This register indicates the major and minor version of the AHCI specification that the HBA implementation supports. The upper two bytes represent the major version number and the lower two bytes represent the minor version number.

22.4.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MJR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MNR															
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

22.4.4.4 Fields

Field	Function
31-16 MJR	Major version number This bit indicates that the major version is 1.
15-0 MNR	Minor version number This bit indicates that the minor version is “3.1”.

22.4.5 Command completion coalescing control register (CCC_CTL)

22.4.5.1 Offset

Register	Offset
CCC_CTL	14h

22.4.5.2 Function

This register configures the command completion coalescing feature for the entire HBA.

22.4.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CC								INT					Reserved		EN
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

22.4.5.4 Fields

Field	Function
31-16 TV	Timeout value The timeout value is specified in 1 ms intervals. The timer accuracy must be within 5%. hCccTimer is loaded with this timeout value. hCccTimer is only decremented when commands are outstanding on selected ports, as defined in Section 11.2 of the SATA AHCI Specification. The HBA signals a CCC interrupt when hCccTimer has decremented to 0. hCccTimer is reset to the timeout value on the assertion of each CCC interrupt. A timeout value of 0 is reserved.
15-8 CC	Command completions This bit specifies the number of command completions that are necessary to cause a CCC interrupt. The HBA has an internal command completion counter and hCccComplete. hCccComplete is incremented by one each time a selected port has a command completion. When hCccComplete is equal to the command completions value, a CCC interrupt is signaled. The internal command completion counter is reset to 0 on the assertion of each CCC interrupt. A value of 0 for this field disables CCC interrupts being generated based on the number of commands completed, which means CCC interrupts are only generated based on the timer in this case.
7-3 INT	Interrupt This bit specifies the interrupt used by the CCC feature. This interrupt must be marked as unused in the ports implemented (PI) register by setting the corresponding bit to 0. Therefore, the CCC interrupt corresponds to the interrupt for an unimplemented port on the controller. When a CCC interrupt occurs, the IS[IPS[INT]] bit is asserted to 1. This field also specifies the interrupt vector used for MSI.
2-1 —	Reserved
0 EN	Enable Software changes only the contents of the TV and CC fields when EN is cleared. When this bit is set, any updated values for the TV and CC fields take effect. 0b - The command completion coalescing feature is disabled and no CCC interrupts are generated. 1b - The command completion coalescing feature is enabled and CCC interrupts may be generated based on timeout or command completion conditions.

22.4.6 HBA capabilities extended register (CAP2)

22.4.6.1 Offset

Register	Offset
CAP2	24h

22.4.6.2 Function

This register indicates capabilities of the HBA to driver software.

22.4.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													APST	NVMP	BOH
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

22.4.6.4 Fields

Field	Function
31-3 —	Reserved
2 APST	Automatic partial to slumber transitions 0b - Automatic partial to slumber transitions are not supported. 1b - The HBA supports automatic partial to slumber transitions.
1 NVMP	NVMHCI present 0b - The HBA does not support NVMHCI. 1b - The HBA includes support for NVMHCI and the registers at offset 60h-9Fh are valid.
0 BOH	BIOS/OS handoff 0b - The HBA does not support BIOS/OS handoff and the BOHC global HBA register is not implemented. 1b - The HBA supports the BIOS/OS handoff mechanism defined in Section 10.6 of the SATA AHCI Specification and implements the BOHC global HBA register.

22.4.7 Port config register (PCFG)

22.4.7.1 Offset

Register	Offset
PCFG	A4h

22.4.7.2 Function

This register is used to program and configure the controller port.

22.4.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									TPSS						
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	TPRS			Reserved			Reserved	Reserved		PAD					
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0

22.4.7.4 Fields

Field	Function
31-23 —	Reserved
22-16 TPSS	Microsecond timer post scaler
15 —	Reserved
14-12 TPRS	Microsecond timer pre scaler
11-9 —	Reserved
8 —	Reserved
7-6 —	Reserved
5-0 PAD	Port address. The settings not defined below are reserved. 000010b - Address configuration/status register set

22.4.8 Port Phy1Cfg register (PPCFG)

22.4.8.1 Offset

Register	Offset
PPCFG	A8h

22.4.8.2 Function

This register controls the configuration of the PHY control layer 1 for port 0. This register holds configuration values for both the OOB generator and OOB detector for each of the two OOB pulses, COMINIT/COMRESET and COMWAKE. The separate values allow multiple configurations of the clocks for running either the OOB generator or detector. The PHY control layer implements the OOB generation logic synchronous TXCLK, which is a 75 MHz input clock to the host controller. The OOB detector logic works synchronous to PMCLK, which is a fixed clock of 150 MHz and used solely for the detector clock.

22.4.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	ESDF	ERSN	PSS	PSSO	STB	PBPNA	PBCE	PBPE	PBPS			FPR	Reserved	SNR	SNM	TTA
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TTA															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

22.4.8.4 Fields

Field	Function
31 ESDF	Enable signal detect filter When set, a single de-assertion of signal detect or the coreRxDataValid, when the PHY Init state machine is in the PHYReady state, causes the state machine to exit the PHYReady state and return to a PhyNotReady state. This results in the OOB and speed negotiation running again.
30 ERSN	Enable reset speed negotiation When set, the PHY control layer enables only a single speed on the RX path during speed negotiation. This speed is determined as the fastest support for the first round falling to the lowest speed for the final round. Each round of speed negotiation is terminated by the host issuing a COMRESET and rerunning OOB before beginning the next round of speed negotiation as detailed in reset speed negotiation (RSN).
29 PSS	PhyControl SerDes slumber This bit selects SerDes slumber CMU during link slumber. When this bit is set and the controller enters slumber, an extra control signal is applied to the SerDes to slumber the clock block within SerDes. This yields an extra power savings that is SerDes specific.
28 PSSO	PhyControl select SerDes OOB This bit selects SerDes OOB or internally decoded OOB signalling as input. 0b - Select SerDes decoded OOB signalling 1b - Select internally decoded OOB signalling
27 STB	Status bit This bit provides the status of the Gen fixed clocks parameter. This bit indicates if the PHY control layer is running from a fixed frequency clock or a variable clock derived from the TX clock of the SerDes.
26 PBPNA	PhyControl BIST pattern no aligns Setting this bit causes the PHY control pattern generator to transmit each pattern continuously.
25 PBCE	PhyControl BIST clear error When a pattern mismatch occurs, this bit needs to be set and then negated to clear the error. 0b - Pattern match error bit is not cleared 1b - Clears the pattern match error bit
24 PBPE	PhyControl BIST pattern enable This bit controls enabling/disabling the PHY control test pattern generation. For more information, refer to PhyControl BIST modes 0b - Disables the PHY control test pattern generation 1b - Enables the PHY control test pattern generation
23-21 PBPS	PhyControl BIST pattern select The settings not defined below are reserved. 000b - LBP (generator clock only) 001b - LFTP 010b - MFTP 011b - HFTP 100b - PRBS pattern 101b - BIST pattern (default)
20 FPR	Force PHY Ready This bit determines how PHY Ready is driven. 0b - Normal operation mode 1b - FrcPhyRdy: In this mode, the OOB and speed negotiation states of the PHY Init state machine are bypassed. The PHY Init state machine directly enters PHY Ready after reset. The Tx buffer IDLE control is forced off.

Table continues on the next page...

Field	Function
19 —	Reserved
18 SNR	Speed negotiation rate When this bit is set set to 1, the speed negotiation runs only at the rate programmed in the PxSCTL[SPD] bit.
17 SNM	Speed negotiation method If the SATA host controller supports Gen3 speed, this bit must be set to 1. If the SATA host controller is limited to Gen2 or Gen1 speed, this bit may use the default value 0. 0b - The speed negotiation runs starting at the fastest supported speed down to the Gen1 speed. 1b - The speed negotiation runs starting at the Gen1 speed up to the fastest supported speed.
16-0 TTA	It determines the time period. This bit determines the time period that the controller transmits and waits for ALIGNp during speed negotiation. This value is derived for the PMCLK period.

22.4.9 Port Phy2Cfg register (PP2C)

22.4.9.1 Offset

Register	Offset
PP2C	ACh

22.4.9.2 Function

This register controls the configuration of the PHY control OOB timing for the COMINIT parameters for port 0. For more info on OOB timing setup, refer to PHY control configuration register OOB timing setup.

Table 22-3. OOB specifications

Interval	Specification value (ns)	OOB generator TXCLK (75 MHz)	OOB detector PMCLK (150 MHz)
Negate minimum	≥ 525.0	28h	-
Gap nominal	= 320.0	18h	-
Gap maximum	≤ 525.0	-	4Dh
Gap minimum[1]	≥ 200.0	-	1Fh

1. While calculating the CIBGMN value, add 25 ns to the specification value.

22.4.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CINMP								CIBGN							
Reset	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CIBGMX								CIBGMN							
Reset	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	1

22.4.9.4 Fields

Field	Function
31-24 CINMP	COMINIT negate minimum period OOB function: generator clock
23-16 CIBGN	COMINIT burst gap nominal OOB function: generator clock
15-8 CIBGMX	COMINIT burst gap maximum OOB function: detector clock
7-0 CIBGMN	COMINIT burst gap minimum OOB function: detector clock

22.4.10 Port Phy3Cfg register (PP3C)

22.4.10.1 Offset

Register	Offset
PP3C	B0h

22.4.10.2 Function

This register controls the configuration of the PHY control OOB timing for the COMWAKE parameters for port 0. For more info on OOB timing setup, refer to PHY control configuration register OOB timing setup.

The following table provides the OOB specifications for the chip.

Table 22-4. OOB specifications

Interval	Specification value (ns)	OOB generator TXCLK (75 MHz)	OOB detector PMCLK (150 MHz)
Negate minimum	≥ 175.0	0Eh	-
Gap nominal	= 106.6	08h	-
Gap maximum[1]	≤ 175.0	-	15h
Gap minimum	≥ 35.0	-	09h

1. While calculating the CWBGMX value, subtract 25 ns from the specification value.

22.4.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CWNMP								CWBGN							
Reset	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CWBGMX								CWBGMN							
Reset	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0

22.4.10.4 Fields

Field	Function
31-24 CWNMP	COMWAKE negate minimum period OOB function: generator clock
23-16 CWBGN	COMWAKE burst gap nominal OOB function: generator clock
15-8 CWBGMX	COMWAKE burst gap maximum OOB function: detector clock

Table continues on the next page...

Field	Function
7-0 CWBGMN	COMWAKE burst gap minimum OOB function: detector clock

22.4.11 Port Phy4Cfg register (PP4C)

22.4.11.1 Offset

Register	Offset
PP4C	B4h

22.4.11.2 Function

This register controls the configuration of the PHY control burst timing for the COM parameters for port 0. For more info on OOB timing setup, refer to PHY control configuration register OOB timing setup.

The following table provides the OOB specifications for the chip.

Table 22-5. OOB specifications

Interval	Specification value (ns)	OOB generator TXCLK (75 MHz)	OOB detector PMCLK (150 MHz)
Signal failure detection	-	-	4Ah
Burst nominal	= 106.6	08h	-
Burst maximum	≤ 109.9 non 113.33	-	0Fh

22.4.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PTST								SFD							
Reset	0	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BNM								BMX							
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1

22.4.11.4 Fields

Field	Function
31-24 PTST	Partial to slumber timer This bit provides a value that specifies the delay that the controller should apply while in partial before entering slumber. The value is based on the system clock divided by 128, total delay = (system clock period) * PTST * 128.
23-16 SFD	Signal failure detection OOB function: detector clock If the signal detection de-asserts for a time greater than this, the OOB detector determines this as a line idle and causes the PhyInit state machine to exit the PHY Ready state.
15-8 BNM	COM burst nominal OOB function: generator clock
7-0 BMX	COM burst maximum OOB function: detector clock

22.4.12 Port Phy5Cfg register (PP5C)

22.4.12.1 Offset

Register	Offset
PP5C	B8h

22.4.12.2 Function

This register controls the configuration of the PHY control retry interval timing for port 0. For more info on OOB timing setup, refer to PHY control configuration register OOB timing setup.

Table 22-6. OOB specifications

Interval	Specification value	Calculated value for PMCLK (150 MHz)
Rate change timer	≥ 54.2 µs / 4	03FFh
Retry interval	≥ 10.0 ms	0C96A8h

22.4.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RCT												RIT			
Reset	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RIT															
Reset	1	0	0	1	0	1	1	0	1	0	1	0	0	1	0	0

22.4.12.4 Fields

Field	Function
31-20 RCT	Rate change timer This bit provides a value based on the 54.2 µs for which a SATA device transmits at a fixed rate ALIGNp after OOB has completed. For a fast SerDes, this value is recommended to be 54.2 µs / 4. OOB function: detector clock
19-0 RIT	Retry interval timer The calculated value is divided by two and the lower digit of precision is not needed. OOB function: detector clock

22.4.13 AXI cache control register (AXICC)

22.4.13.1 Offset

Register	Offset
AXICC	BCh

22.4.13.2 Function

This register controls the value of the AWCACHE and ARCACHE used to distinguish each address operation on the address bus.

AWCACHE/ARCACHE control (Non AHCI standard)

Both AWCACHE and ARCACHE can be controlled during header, command, and status of the data phase AXI bus operations. The values driven onto these signals are controlled by the AXI cache control register for operations related to the header, physical region description, command, and status operations. The values driven onto these signals during a data operation can be further controlled from the relevant PRDT entry as shown in Table 22-15. Note that this control is not an AHCI standard.

The following table summarizes each of the possible attributes that can be set through the cache bits.

Table 22-7. Attributes list

ARCACHE[3:0]/AWCACHE[3:0]				Transaction attributes
WA	RA	C	B	
0	0	0	0	Non-cacheable and non-bufferable
0	0	0	1	Bufferable only
0	0	1	0	Cacheable, but do not allocate
0	0	1	1	Cacheable and bufferable, but do not allocate
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Cacheable write-through, allocate on reads only

Table continues on the next page...

Table 22-7. Attributes list (continued)

ARCACHE[3:0]/AWCACHE[3:0]				Transaction attributes
0	1	1	1	Cacheable write-back, allocate on reads only
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Cacheable write-through, allocate on writes only
1	0	1	1	Cacheable write-back, allocate on writes only
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Cacheable write-through, allocate on both reads and writes
1	1	1	1	Cacheable write-back, allocate on both reads and writes

22.4.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		EARC	EAWC	AWCF				AWCD				AWCFD			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ARCP				ARCH				ARCF				ARCA			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

22.4.13.4 Fields

Field	Function
31-30 —	Reserved
29	Enable ARCACHE

Table continues on the next page...

Field	Function
EARC	This bit provides the control from the PRDT entries used during the data phase of this operation (global control).
28 EAWC	Enable AWCACHE This bit provides the control from the PRDT entries used during the data phase of this operation (global control).
27-24 AWCF	Address write cache FIS This bit provides the value driven onto AWCACHE when the AXI master is posting a status FIS write address to the memory controller.
23-20 AWCD	Address write cache data This bit provides the value driven onto AWCACHE when the AXI master is posting a data burst write address to the memory controller when the data burst is not the final burst in the transfer.
19-16 AWCFD	Address write cache final data This bit provides the value driven onto AWCACHE when the AXI master is posting a data burst write address to the memory controller when the data burst is the final burst in the transfer.
15-12 ARCP	Address read cache PRD This bit provides the value driven onto ARCACHE when the AXI master is posting a PRD read address to the memory controller.
11-8 ARCH	Address read cache header This bit provides the value driven onto ARCACHE when the AXI master is posting a header or physical region descriptor read address to the memory controller.
7-4 ARCF	Address read cache FIS This bit provides the value driven onto ARCACHE when the AXI master is posting a command FIS read address to the memory controller.
3-0 ARCA	Address read cache ATAPI This bit provides the value driven onto ARCACHE when the AXI master is posting a data burst.

22.4.14 Port AXICfg register (PAXIC)

22.4.14.1 Offset

Register	Offset
PAXIC	C0h

22.4.14.2 Function

This register controls the configuration of the AXI bus operation for port 0. The AXIPE and ADBW bits can only be configured through access to port 0 configuration space.

22.4.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved			ENPE	Reserved		AAO	ECM	OTL				MARIDD			
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	MARID				MAWIDD				MAWID				Reserved		Reserved	
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0

22.4.14.4 Fields

Field	Function
31-29 —	Reserved
28 ENPE	Enable non-zero 4 MB PRD entries
27-26 —	Reserved
25 AAO	Allow address overwrite This bit allows the PxCLB[9:0] and PxFB[7:0] bits to be overwritten.
24 ECM	Enable the context management This bit enables the context management system in the memory.
23-20 OTL	Outstanding transfer limit This bit limits the maximum number of outstanding transfers supported. • For the 264 DWord transport layer implementation, this can be programmed between 1 and 16 • For the 136 DWord transport layer implementation, this can be programmed between 1 and 8 • For the 72 DWord transport layer implementation, this can be programmed between 1 and 4 The settings not defined below are reserved. 0000b - Transfers limited by the transport layer FIFO space/fill level.
19-16 MARIDD	Memory address read ID This bit provides memory address read ID for data transfers.
15-12 MARID	Memory address read ID This bit provides memory address read ID for non data transfers.

Table continues on the next page...

Field	Function
11-8 MAWIDD	Memory address write ID This bit provides memory address write ID for data transfers.
7-4 MAWID	Memory address write ID This bit provides memory address write ID for non data transfers.
3-2 —	Reserved
1-0 —	Reserved

22.4.15 AXI PROT control register (AXIPC)

22.4.15.1 Offset

Register	Offset
AXIPC	C4h

22.4.15.2 Function

This register controls the value of AWPROT and ARPROT that are used to distinguish each address operation on the address bus. The following table summarizes each of the possible attributes that can be set through the PROT bits.

Table 22-8. Attributes table

ARPROT[2:0]/AWPROT[2:0]	Protection level
[0]	1 : privileged access
	0 : normal access
[1]	1 : non-secure access
	0 : secure access
[2]	1 : instruction access
	0 : data access

AWPROT/ARPROT control (Non AHCI standard)

Both AWPROT and ARPROT can be controlled during header, command, and status of the data phase AXI bus operations. The values driven onto these signals are controlled by the AXI PROT control register (AXIPC) for operations related to the header, physical region description, command, and status operations. The values driven onto these signals during a data operation can be further controlled from the relevant PRD entry as shown in Table 22-15

22.4.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		EARP	EAWP	Reserved	AWPF			Reserved	AWPD			Reserved	AWPFD		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ARPP			Reserved	ARPH			Reserved	ARPF			Reserved	ARPD		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.15.4 Fields

Field	Function
31-30 —	Reserved
29 EARP	Enable ARPROT This bit provides the control from the PRDT entries used during the data phase of this operation.
28 EAWP	Enable AWPROT This bit provides the control from the PRDT entries used during the data phase of this operation.
27 —	Reserved
26-24 AWPF	Address write PROT FIS This bit provides the value driven onto AWPROT when the AXI master is posting a status FIS write address to the memory controller.
23 —	Reserved
22-20	Address write PROT data

Table continues on the next page...

Field	Function
AWPD	This bit provides the value driven onto AWPROT when the AXI master is posting a Data burst write address to the memory controller when the data burst is not the final burst in the transfer.
19 —	Reserved
18-16 AWPFD	Address write PROT final data This bit provides the value driven onto AWPROT when the AXI master is posting a Data burst write address to the memory controller when the data burst is the final burst in the transfer.
15 —	Reserved
14-12 ARPP	Address read PROT PRD This bit provides the value driven onto ARPROT when the AXI master is posting a PRD read address to the memory controller.
11 —	Reserved
10-8 ARPH	Address read PROT header This bit provides the value driven onto ARPROT when the AXI master is posting a header read address to the memory controller.
7 —	Reserved
6-4 ARPF	Address read PROT FIS This bit provides the value driven onto ARPROT when the AXI master is posting a command FIS read address to the memory controller.
3 —	Reserved
2-0 ARPD	Address read PROT data This bit provides the value driven onto ARPROT when the AXI master is posting a data burst read address to the memory controller.

22.4.16 Port TransCfg register (PTC)

22.4.16.1 Offset

Register	Offset
PTC	C8h

22.4.16.2 Function

This register controls the configuration of the transport layer for port 0.

22.4.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						ITM	ENBD	Reserved	RXWM						
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

22.4.16.4 Fields

Field	Function
31-10 —	Reserved
9 ITM	Initialize transport memories 0b - The machine runs writing 0 to all locations within the memory 1b - Causes the memory initialize hardware state machine to enter reset
8 ENBD	Enable back down When a port multiplier is attached and an attempt to send a command to an attached drive results in three consecutive retries due to R_ERR receptions, the command removes for the transport layer and returns to the pending queue. This is to avoid unnecessary retries owing to the device trying to send a set device bits FIS while the host is trying to send a command. This feature improves performance as it allows the controller to queue any pending commands to the other drives.
7 —	Reserved
6-0 RXWM	RxWaterMark This bit sets the minimum number of free location within the RX FIFO before the watermark is exceeded. The transport layer in turn instructs the link layer to transmit HOLDS to the transmitting end. **NOTE:** It can take some time for the HOLDs to get to the other end and in the interim period there must be enough room in the FIFO to absorb all data that could arrive. An initial value of 7'h29 is recommended.

22.4.17 Port LinkCfg register (PLC)

22.4.17.1 Offset

Register	Offset
PLC	D0h

22.4.17.2 Function

This register controls the configuration of the link layer for port 0.

22.4.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PMPRA					POE	PRT									
Reset	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AIR								EPNRT	S4A	RXSE	TXSE	TXPJ	TXC	RXBC	TXBC
Reset	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0	0

22.4.17.4 Fields

Field	Function
31-27 PMPRA	Power management primitive rate acknowledge This bit determines the number of PMACK primitives sent when a power management state transition is requested by the host.
26 POE	Primitive override enable When set, this bit enables the replacement of a single primitive, as specified by override primitive/CD, when the link layer state machine is in the Prim override state. This bit must be set to enable this feature.
25-16 PRT	PHY Ready timer

Table continues on the next page...

Field	Function
	This bit specifies the timeout value of the PHY Ready timer. If EnPhyReadyTimeOut is set, the link layer counts down on every rising edge of TX clock, as long as PHY Ready is de-asserted. When the counter reaches zero, a PhyReset is issued to the PHY to try and re-establish communications with the far-end. The timer is initially loaded with a value equal to the concatenation of { PHY Ready Timer, 9'h000}.
15-8 AIR	ALIGN insertion rate The SATA AHCI Specification requires that the link layer sends a pair of ALIGN primitives at least every 254 DWords of data. This is achieved by setting ALIGN insertion rate to "11111111". However, for test purposes it is possible to send ALIGNs at a higher rate. This can be achieved by setting ALIGN insertion rate to a lower value i.e. (ALIGN insertion rate-1). DWords are sent by the link layer between each set of ALIGN primitive pairs. **NOTE:** If the S4A bit is set, the ALIGN insertion rate should not be set to four or less. If S4A is not set, the ALIGN insertion rate should not be set to two or less.
7 EPNRT	Enable PHY not ready timer If PHY Ready is de-asserted for a long time as specified by PRT bit, then this bit, when asserted, enables the link layer to re-issue a PhyReset, thereby re-initiating OOB.
6 S4A	Send 4 Aligns If this bit is asserted, four ALIGN primitives are transmitted at the specified rate, instead of the normal two ALIGN primitives.
5 RXSE	Rx scramble enable If this bit is asserted, the de-scrambling of the receive data is enabled as per the SATA AHCI Specification.
4 TXSE	Tx scramble enable If this bit is asserted, the scrambling of the transmit data is enabled as per the SATA AHCI Specification.
3 TXPJ	Tx Prim junk If this bit is de-asserted, the scrambled junk data is sent after a CONT primitive, as per the SATA AHCI Specification. If this bit is asserted, then the single character 32'hDEADBEEF is sent continuously to aid debug.
2 TXC	Tx CONT If this bit is asserted, the transmission of CONT primitives is enabled. If de-asserted, then long sequences of repeated primitives can be sent by the link layer.
1 RXBC	Rx bad CRC When a rising edge is detected on this bit, it causes a bad CRC to be detected for the current frame. This bit has must be set to enable this feature.
0 TXBC	Tx bad CRC A bad CRC (inverted value of the correct CRC) value is transmitted for one FIS only by the link layer when a rising edge is detected on this signal. This bit must be set to enable this feature.

22.4.18 Port LinkCfg1 register (PLC1)

22.4.18.1 Offset

Register	Offset
PLC1	D4h

22.4.18.2 Function

This register controls the configuration of the link layer for port 0.

22.4.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									CD	POS					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.18.4 Fields

Field	Function
31-7 —	Reserved
6 CD	Data character or primitive This bit specifies whether the data used during the primitive override should be a data character or a primitive. For example, if CD = 1, Prim override state = L_SendEOF and override primitive = WTRM, then a WTRM primitive is inserted into the data stream instead of an EOF (whenever a rising edge is seen on PLC[POE]. If CD = 0, then a normal data character (as specified by PLC2[OP]) is inserted into the data stream instead of the EOF.
5-0 POS	Primitive override state This bit is used in the primitive override debug functionality. When the link layer detects a positive edge on PLC[POE], it overrides the next primitive that would be inserted during the PLC1[POS], with the data specified by the PLC2[OP] and PLC1[CD] configuration bits.

22.4.19 Port LinkCfg2 register (PLC2)

22.4.19.1 Offset

Register	Offset
PLC2	D8h

22.4.19.2 Function

This register controls the configuration of the link layer for port 0.

22.4.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	OP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.19.4 Fields

Field	Function
31-0 OP	Override primitive This bit specifies the data to be used in the primitive override debug functionality that is described in the definition of Port LinkCfg1 register (PLC1).

22.4.20 Port LinkStatus1 register (PLS1)

22.4.20.1 Offset

Register	Offset
PLS1	E0h

22.4.20.2 Function

This register indicates the status of the link layer for port 0. This register acts as an accumulator for the SerDes errors. Each counter can be cleared by writing 8'hFF to the individual byte.

22.4.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	KCEC								PIEC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CEC								DEC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.20.4 Fields

Field	Function
31-24 KCEC	Kchar error count This bit provides the number of DWords that have been received from the PHY, where one or more control character errors have been detected. A value of 255 indicates an error count of 255 or more as this counter does not wrap around to zero. The count value is updated with its current value each time the LinkStatus1 register is read.
23-16 PIEC	PHY internal error count This bit provides the number of DWords that have been received from the PHY, where one or more internal errors have been detected. A value of 255 indicates an error count of 255 or more as this counter does not wrap around to zero. The count value is updated with its current value each time the LinkStatus1 register is read.
15-8 CEC	Code error count

Table continues on the next page...

Field	Function
	This bit provides the number of DWords that have been received from the PHY, where one or more code errors have been detected. A value of 255 indicates an error count of 255 or more as this counter does not wrap around to zero. The count value is updated with its current value each time the LinkStatus1 register is read.
7-0 DEC	Disparity error count This bit provides the number of DWords that have been received from the PHY, where one or more disparity errors have been detected. A value of 255 indicates an error count of 255 or more as this counter does not wrap around to zero. The count value is updated with its current value each time the LinkStatus1 register is read.

22.4.21 Port CmdConfig register (PCMDC)

22.4.21.1 Offset

Register	Offset
PCMDC	E4h

22.4.21.2 Function

This register controls the operation of the command layer for port 0.

22.4.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		TSVIE	TSVI	TSVT											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSVT				Reserved										ETLL	ETLLB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.21.4 Fields

Field	Function
31-30 —	Reserved
29 TSVIE	Trustzone slave ID violation interrupt enable The interrupt is seen on bit 3 of the interrupt status register in the general register group. Refer to the SATA AHCI Specification for more details.
28 TSVI	Trustzone slave ID violation interrupt
27-12 TSVT	Trustzone Slave ID of violating transaction When the slave port rejects a read or write as slave error due to security violation, this register records the AXI ID of the violating transaction.
11-2 —	Reserved
1 ETLL	Enable transport layer loopback
0 ETLLB	Enable transport layer loopback in the BIST-L mode This bit must be set to 1 before performing the BIST-L test.

22.4.22 Port PhyControl status register (PPCS)

22.4.22.1 Offset

Register	Offset
PPCS	E8h

22.4.22.2 Function

This register controls the operation of the status of the phy control layer for port 0.

22.4.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PHYCE		PHYDE		PHYKC	PHYD										
W																
Reset	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHYD					CCAC	CCA					PCTRLS				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

22.4.22.4 Fields

Field	Function
31-30 PHYCE	Current 2-bit code error This bit provides snapshot within the PHY control layer.
29-28 PHYDE	Current 2-bit disparity error This bit provides snapshot within the PHY control layer.
27 PHYKC	Current 1-bit K character This bit provides snapshot within the PHY control layer.
26-11 PHYD	Current 16-bit data This bit provides snapshot within the PHY control layer.
10 CCAC	Comma alignment has changed
9-5 CCA	Current comma alignment
4-0 PCTRLS	PHY control state

22.4.23 Timer control register (TCR)

22.4.23.1 Offset

Register	Offset
TCR	F0h

22.4.23.2 Function

This register controls the operation of the timer prescaler that configures a 10 µs pulse generator used to control the operation of the slumber timer. This pulse generator is used in port 0.

22.4.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TPS												
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

22.4.23.4 Fields

Field	Function
31-13 —	Reserved
12-0 TPS	Timer preScalar value The system clock is divided by the ratio to generate a 10 µs clock pulse to run the port layers.

22.4.24 Port x command list base address register (PxCLB)

22.4.24.1 Offset

Register	Offset
PxCLB	100h

22.4.24.2 Function

This bit indicates the 32-bit base physical address for the command list for this port.

22.4.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CLB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLB						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.24.4 Fields

Field	Function
31-10 CLB	Command list base address This bit indicates the 32-bit base physical address for the command list for this port. This base is used when fetching commands to execute. The structure pointed to by this address range is 1 KB in length. This address must be 1 KB aligned as indicated by bits [9:0] being read only.
9-0 —	Reserved

22.4.25 Port x command list base address upper 32-bit register (PxCLBU)

22.4.25.1 Offset

Register	Offset
PxCLBU	104h

22.4.25.2 Function

Indicates the upper 32-bits for the command list base physical address for this port. This base is used when fetching commands to execute. This register shall be read only ‘0’ for HBAs that do not support 64-bit addressing.

22.4.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CLBU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CLBU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.25.4 Fields

Field	Function
31-0 CLBU	Command list base address upper This bit indicates the upper 32 bits for the command list base physical address for this port. This base is used when fetching commands to execute. This bit is read only '0' for HBAs that do not support 64-bit addressing.

22.4.26 Port x FIS base address register (PxFB)

22.4.26.1 Offset

Register	Offset
PxFB	108h

22.4.26.2 Function

Indicates the 32-bit base physical address for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256-byte aligned as indicated by bits 07:00 being read only.

22.4.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FB								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.26.4 Fields

Field	Function
31-8 FB	FIS base address This bit indicates the 32-bit base physical address for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256 bytes aligned as indicated by bits [7:0] being read only. When FIS-based switching is in use, this structure is 4 KB in length and the address is 4 KB aligned.
7-0 —	Reserved

22.4.27 Port x FIS base address upper 32-bit register (PxFBU)

22.4.27.1 Offset

Register	Offset
PxFBU	10Ch

22.4.27.2 Function

Indicates the upper 32-bits for the received FIS base physical address for this port. This register shall be read only ‘0’ for HBAs that do not support 64-bit addressing.

22.4.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FBU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FBU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.27.4 Fields

Field	Function
31-0 FBU	FIS base address upper This bit indicates the upper 32 bits for the received FIS base physical address for this port. This register is read only '0' for HBAs that do not support 64-bit addressing.

22.4.28 Port x interrupt status register (PxIS)

22.4.28.1 Offset

Register	Offset
PxIS	110h

22.4.28.2 Function

Port x interrupt status register

22.4.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CPDS	TFES	HBFS	HBDS	IFS	NTFS	Reserved	OFS	IPMS	PRCS	Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DMPS	PCS	DPS	UFS	SDBS	DSS	PSS	DHRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.28.4 Fields

Field	Function
31 CPDS	Cold port detect status When set, a device status has changed as detected by the cold presence detect logic. This bit can either be set due to a non-connected port receiving a device or a connected port having its device removed. This bit is only valid if the port supports cold presence detect as indicated by PxCMD[CPD] set to 1.
30 TFES	Task file error status This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set.
29 HBFS	Host bus fatal error status This bit indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer. In PCI, such an indication would be a target or master abort.
28	Host bus data error status

Table continues on the next page...

Field	Function
HBDS	This bit indicates that the HBA encountered a data error (uncorrectable ECC/parity) when reading from or writing to system memory.
27 IFS	Interface fatal error status This bit indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. Refer to Section 6.1.2 of the SATA AHCI Specification fore more details.
26 NTFS	Interface non-fatal error status This bit indicates that the HBA encountered an error on the SATA interface but was able to continue operation. Refer to Section 6.1.2 of the SATA AHCI Specification for more details.
25 —	Reserved
24 OFS	Overflow status This bit indicates that the HBA received more bytes from a device than was specified in the PRD table for the command.
23 IPMS	Incorrect port multiplier status This bit indicates that the HBA received a FIS from a device that did not have a command outstanding. This bit is set during enumeration of devices on a port multiplier due to the normal port multiplier enumeration process. It is recommended that IPMS only be used after enumeration is complete on the port multiplier. IPMS is not set when an asynchronous notification is received (a set device bits FIS with the notification 'N' bit set to 1).
22 PRCS	PhyRdy change status When set, this bit indicates the internal PhyRdy signal changed state. This bit reflects the state of PxSERR[DIAG[N]]. To clear this bit, software must clear PxSERR[DIAG[N]] to 0.
21-8 —	Reserved
7 DMPS	Device mechanical presence status When set, this bit indicates that a mechanical presence switch associated with this port has been opened or closed, which may lead to a change in the connection state of the device. This bit is only valid if both CAP[SMPS] and PxCMD[MPSP] are set to 1.
6 PCS	Port connect change status This bit reflects the state of PxSERR[DIAG[X]]. This bit is only cleared when PxSERR[DIAG[X]] is cleared. 0b - No change in current connect status 1b - Change in current connect status
5 DPS	Descriptor processed A PRD with the 'I' bit set has transferred all of its data. Refer to Section 5.4.2 of the SATA AHCI Specification for more details.
4 UFS	Unknown FIS interrupt When set, this bit indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to 0 by software clearing the PxSERR[DIAG[F]] bit to 0. **NOTE:** This bit does not directly reflect the PxSERR[DIAG[F]] bit. PxSERR[DIAG[F]] is set immediately when an unknown FIS is detected, whereas this bit is set when that FIS is posted to memory. Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync.
3 SDBS	Set device bits interrupt A set device bits FIS has been received with the 'I' bit set and has been copied into system memory.

Table continues on the next page...

Field	Function
2 DSS	DMA setup FIS interrupt A DMA setup FIS has been received with the 'I' bit set and has been copied into system memory.
1 PSS	PIO setup FIS interrupt A PIO setup FIS has been received with the 'I' bit set, it has been copied into system memory, and the data related to that FIS has been transferred. This bit is set even if the data transfer resulted in an error.
0 DHRS	Device to host register FIS interrupt A D2H register FIS has been received with the 'I' bit set, and has been copied into system memory.

22.4.29 Port x command and status register (PxCMD)

22.4.29.1 Offset

Register	Offset
PxCMD	118h

22.4.29.2 Function

When set to ‘1’, indicates that this port supports Port Multiplier FIS-based switching. When cleared to ‘0’, indicates that this port does not support FIS-based switching.

22.4.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ICC				AS P	ALP E	DLAE	ATAPI	APST E	FBSC P	ES P	CPD	MPSP	HPCP	PMA	CP S
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CR	F R	MPS S	CCS					Reserved			FR E	CLO	POD	SUD	S T
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.29.4 Fields

Field	Function
31-28 ICC	Interface communication control This bit is used to control power management states of the interface. If the Link layer is currently in the L_IDLE state, writes to this field causes the HBA to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writes to this field has no effect. When system software writes a non-reserved value other than No-Op (0h), the HBA performs the action and update this bit back to Idle (0h). If software writes to this bit to change the state to a state the link is already in (i.e. interface is in the active state and a request is made to go to the active state), the HBA takes no action and returns this bit to Idle. If the interface is in a low power state and software wants to transition to a different low power state, software must first bring the link to active and then initiate the transition to the desired low power state. The settings not defined below are reserved. 0000b - No-Op/Idle: When software reads this value, it indicates the HBA is ready to accept a new interface control command, although the transition to the previously selected state may not yet have occurred. 0001b - Active: Causes the HBA to request a transition of the interface into the active state. 0010b - Partial: Causes the HBA to request a transition of the interface to the partial state. The SATA device may reject the request and the interface remains in its current state. 0110b - Slumber: Causes the HBA to request a transition of the interface to the slumber state. The SATA device may reject the request and the interface must remain in its current state.
27 ASP	Aggressive slumber/partial If CAP[SALP] is cleared to 0, software treats this bit as reserved. Refer to Section 8.3.1.3 of the SATA AHCI Specification for more details. 0b - When ALPE is set, the HBA aggressively enters the partial state when it clears the PxCI register and the PxSACT register is cleared or when it clears the PxSACT register and the PxCI register is cleared. 1b - When ALPE is set, the HBA aggressively enters the slumber state when it clears the PxCI register and the PxSACT register is cleared or when it clears the PxSACT register and the PxCI register is cleared.
26 ALPE	Aggressive link power management enable Software only sets this bit to 1 if CAP[SALP] is set to 1. If CAP[SALP] is cleared to 0, software treats this bit as reserved. Refer to Section 8.3.1.3 of the SATA AHCI Specification for more details. 0b - The aggressive link power management is disabled 1b - The HBA aggressively enters a lower link power state (partial or slumber) based upon the setting of the ASP bit.
25 DLAE	Drive LED on ATAPI enable Refer to Section 10.11 of the SATA AHCI Specification for more details. **NOTE:** The controller does not support ATAPI functionality. 0b - The HBA only drives the LED pin active for commands if PxCMD[ATAPI] is set to 0. 1b - The HBA drives the LED pin active for commands regardless of the state of PxCMD[ATAPI].
24 ATAPI	Device is ATAPI This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. Refer to Section 10.11 of the SATA AHCI Specification for more details. **NOTE:** The controller does not support ATAPI functionality. 0b - The connected device is not an ATAPI device 1b - The connected device is an ATAPI device.

Table continues on the next page...

Field	Function
23 APSTE	Automatic partial to slumber transitions enabled Software only sets this bit to 1 if CAP2[APST] is set to 1. If CAP2[APST] is cleared to 0, software treats this bit as reserved. 0b - The HBA does not perform automatic partial to slumber transitions. 1b - The HBA performs automatic partial to slumber transitions.
22 FBSCP	FIS-based switching capable port This bit may only be set to 1 if both CAP[SPM] and CAP[FBSS] are set to 1. 0b - This port does not support FIS-based switching. 1b - This port supports port multiplier FIS-based switching.
21 ESP	External SATA port When this bit is set to 1, the CAP[SXS] bit is set to 1. ESP is mutually exclusive with the HPCP bit in this register. If ESP is set to 1, then the port may experience hot plug events. 0b - This port's signal connector is not externally accessible on a signal only connector. 1b - This port's signal connector is externally accessible on a signal only connector (e.g. eSATA connector).
20 CPD	Cold presence detection When this bit is set to 1, PxCMD[HPCP] must also be set to 1. 0b - Platform does not support cold presence detection on this port. 1b - Platform supports cold presence detection on this port.
19 MPSP	Mechanical presence switch attached to port When this bit is set to 1, PxCMD[HPCP] should also be set to 1. 0b - The platform does not support a mechanical presence switch attached to this port. 1b - The platform supports the mechanical presence switch attached to this port.
18 HPCP	Hot plug capable port HPCP is mutually exclusive with the ESP bit in this register. 0b - Indicates that this port's signal and power connectors are not externally accessible through a joint signal and power connector. 1b - Indicates that this port's signal and power connectors are externally accessible through a joint signal and power connector for blindmate device hot plug.
17 PMA	Port multiplier attached This bit is read/write for HBAs that support a port multiplier (CAP[SPM] = 1). This bit is read-only for HBAs that do not support a port multiplier (CAP[SPM] = 0). Software is responsible for detecting whether a port multiplier is present; hardware does not auto-detect the presence of a port Multiplier. Software only sets this bit to 1 when PxCMD[ST] is cleared to 0. 0b - A port multiplier is not attached to the HBA for this port. 1b - A port multiplier is attached to the HBA for this port.
16 CPS	Cold presence state The CPS bit reports whether a device is currently detected on this port through cold presence detection. 0b - The HBA detects through cold presence that there is no device attached to this port. 1b - The HBA detects through cold presence that a device is attached to this port.
15 CR	Command list running When set, the command list DMA engine for the port is running. Refer to AHCI state machine in Section 5.3.2 of the SATA AHCI Specification for more details.
14 FR	FIS receive running When set, the FIS receive DMA engine for the port is running. Refer to Section 10.3.2 of the SATA AHCI Specification for more details.
13 MPSS	Mechanical presence switch state

Table continues on the next page...

Field	Function
	This bit reports the state of a mechanical presence switch attached to this port. If CAP[SMPS] is set to 0, this bit is cleared to 0. Software uses this bit only if both CAP[SMPS] and PxCMD[MPSP] are set to 1. 0b - If CAP[SMPS] is set to 1, the mechanical presence switch is closed. 1b - If CAP[SMPS] is set to 1, the mechanical presence switch is open.
12-8 CCS	Current command slot This bit is valid when PxCMD[ST] is set to 1 and is set to the command slot value of the command that is currently being issued by the HBA. When PxCMD[ST] transitions from 1 to 0, this bit resets to 0. After PxCMD[ST] transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is PxCMD[CCS] + 1. For example, after the HBA has issued its first command, if CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1.
7-5 —	Reserved
4 FRE	FIS receive enable System software must not set this bit until PxFB (and PxFBU) have been programmed with a valid pointer to the FIS receive area. If software wishes to move the base, this bit must first be cleared, and software must wait for the FR bit in this register to be cleared. Refer to Section 10.3.2 of the SATA AHCI Specification for more details. 0b - Received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence, and no FISes are posted to the FIS receive area. 1b - The HBA posts received FISes into the FIS receive area pointed to by PxFB (and for 64-bit HBAs, PxFBU).
3 CLO	Command list override This bit is only set to 1 immediately prior to setting the PxCMD[ST] bit to 1 from a previous value of 0. Setting this bit to 1 at any other time is not supported and results in indeterminate behavior. Software must wait for CLO to be cleared to 0 before setting PxCMD[ST] to 1. 0b - The HBA sets this bit to 0 when PxTFD[STS[BSY]] and PxTFD[STS[DRQ]] have been cleared to 0. A write to this register with a value of 0 has no effect. 1b - Causes PxTFD[STS[BSY]] and PxTFD[STS[DRQ]] to be cleared to 0. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD[STS] register. The HBA sets this bit to 0 when PxTFD[STS[BSY]] and PxTFD[STS[DRQ]] have been cleared to 0. A write to this register with a value of 0 has no effect. This bit is only set to 1 immediately prior to setting the PxCMD[ST] bit to 1 from a previous value of 0. Setting this bit to 1 at any other time is not supported and results in indeterminate behavior. Software must wait for CLO to be cleared to 0 before setting PxCMD[ST] to 1.
2 POD	Power-on device This bit is read/write for HBAs that support cold presence detection on this port as indicated by PxCMD[CPD] set to 1. This bit is read only '1' for HBAs that do not support cold presence detect. When set, the HBA sets the state of a pin on the HBA to 1 so that it may be used to provide power to a cold-presence detectable port.
1 SUD	Spin-up device This bit is read/write for HBAs that support staggered spin-up through CAP[SSS]. This bit is read only '1' for HBAs that do not support staggered spin-up. On an edge detect from 0 to 1, the HBA starts a COMRESET initialization sequence to the device. Clearing this bit to 0 does not cause any OOB signal to be sent on the interface. When this bit is cleared to 0 and PxSCTL[DET] = 0h, the HBA enters listen mode as detailed in Section 10.10.1 of the SATA AHCI Specification.
0 ST	Start When this bit is changed from 0 to 1, the HBA starts processing the command list at entry 0. When this bit is changed from 1 to 0, the PxCI register is cleared by the HBA upon the HBA putting the controller into an idle state. This bit is only set to 1 by software after PxCMD[FRE] has been set to 1. Refer to Section 10.3.1 of the SATA AHCI Specification for more details.

Field	Function
	0b - The HBA does not process the command list. 1b - The HBA processes the command list.

22.4.30 Port x SATA status register (PxSSTS)

22.4.30.1 Offset

Register	Offset
PxSSTS	128h

22.4.30.2 Function

This register conveys the current state of the interface and host. The HBA updates it continuously and asynchronously. When the HBA transmits a COMRESET to the device, this register is updated to its reset values.

22.4.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				IPM				SPD				DET			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.30.4 Fields

Field	Function
31-12	Reserved

Table continues on the next page...

Field	Function
—	
11-8 IPM	Interface power management This bit indicates the current interface state. The settings not defined below are reserved. 0000b - Device not present or communication not established 0001b - Interface in active state 0010b - Interface in partial power management state 0110b - Interface in slumber power management state
7-4 SPD	Current interface speed This bit indicates the negotiated interface communication speed. The settings not defined below are reserved. 0000b - Device not present or communication not established 0001b - Generation 1 communication rate negotiated 0010b - Generation 2 communication rate negotiated 0011b - Generation 3 communication rate negotiated
3-0 DET	Device detection This bit indicates the interface device detection and PHY state. The means by which the implementation determines device presence may be vendor specific. However, device presence must always be indicated anytime a COMINIT signal is received. The settings not defined below are reserved. 0000b - No device detected and PHY communication not established 0001b - Device presence detected but PHY communication not established 0011b - Device presence detected and PHY communication established 0100b - PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode

22.4.31 Port x SATA control register (PxSCTL)

22.4.31.1 Offset

Register	Offset
PxSCTL	12Ch

22.4.31.2 Function

The register is used by software to control SATA capabilities. Writes to this register result in an action being taken by the host adapter or interface. Reads from the register return the last value written to it.

22.4.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				IPM				SPD				DET			
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

22.4.31.4 Fields

Field	Function
31-12 —	Reserved
11-8 IPM	Interface power management transitions allowed Indicates which power states the HBA is allowed to transition to. If an interface power management state is disabled, the HBA is not allowed to initiate that state and the HBA must $PMNAK_P$ any request from the device to enter that state. The settings not defined below are reserved. 0000b - No interface restrictions 0001b - Transitions to the partial state is disabled 0010b - Transitions to the slumber state is disabled 0011b - Transitions to both partial and slumber states are disabled
7-4 SPD	Speed allowed This bit indicates the highest allowable speed of the interface. The settings not defined below are reserved. 0000b - No speed negotiation restrictions 0001b - Limit speed negotiation to Generation 1 communication rate 0010b - Limit speed negotiation to a rate not greater than Generation 2 communication rate 0011b - Limit speed negotiation to a rate not greater than Generation 3 communication rate
3-0 DET	Device detection initialization This bit controls the HBA's device detection and interface initialization. This bit may only be modified when PxCMD[ST] is 0. Changing this bit while the PxCMD[ST] bit is set to 1 results in an undefined behavior. When PxCMD[ST] is set to 1, this field should have a value of 0h. **NOTE:** It is permissible to implement any of the SATA defined behaviors for transmission of COMRESET when DET = 1h. The settings not defined below are reserved. 0000b - No device detection or initialization action requested 0001b - Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. While this field is 1h, COMRESET is transmitted on the interface. Software should leave the DET field set to 1h for a minimum of 1 ms to ensure that a COMRESET is sent on the interface. 0100b - Disable the SATA interface and put PHY in offline mode

22.4.32 Port x SATA error register (PxSERR)

22.4.32.1 Offset

Register	Offset
PxSERR	130h

22.4.32.2 Function

This register contains the diagnostics (DIAG) and error (ERR) bits that contains diagnostic error information and error information.

22.4.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DIAG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.32.4 Fields

Field	Function
31-16 DIAG	Diagnostic This bit contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. **Table 22-9. Diagnostic error information** Bit: 31:27 — Description: Reserved

Table continues on the next page...

Field	Function
	Table 22-9. Diagnostic error information (continued)

Bit	Description
26	Exchanged (X): When set to 1, this bit indicates that a change in device presence has been detected since the last time this bit was cleared. The means by which the implementation determines that the device presence has changed is vendor specific. This bit must always be set to 1 anytime a COMINIT signal is received. This bit is reflected in the PxIS[PCS] bit.
25	Unknown FIS Type (F): Indicates that one or more FISes were received by the transport layer with good CRC, but had a type field that was not recognized.
24	Transport state transition error (T): Indicates that an error has occurred in the transition from one state to another within the transport layer since the last time this bit was cleared.
23	Link Sequence Error (S): Indicates that one or more link state machine error conditions was encountered. The link layer state machine defines the conditions under which the link layer detects an erroneous transition.
22	Handshake Error (H): Indicates that one or more R_ERR handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.
21	CRC Error (C): Indicates that one or more CRC errors occurred with the link layer.
20	Disparity Error (D): This field is not used by AHCI.
19	10B to 8B Decode Error (B): Indicates that one or more 10B to 8B decoding errors occurred.
18	Comm Wake (W): Indicates that a Comm Wake signal was detected by the PHY.
17	PHY Internal Error (I): Indicates that the PHY detected some internal error.
16	PhyRdy Change (N): Indicates that the PhyRdy signal changed state. This bit is reflected in the PxIS[PRCS] bit.

15-0 ERR	Error

Field	Function
	This bit contains error information for use by host software in determining the appropriate response to the error condition.

Table 22-10. Error information

Bit	Description
15:12	Reserved
11	Internal Error (E): The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. The internal error may include a master or target abort when attempting to access system memory, an elasticity buffer overflow, a primitive mis-alignment, a synchronization FIFO overflow, and other internal error conditions. Typically when an internal error occurs, a non-fatal or fatal status bit in the PxIS register is also set to give software guidance on the recovery mechanism required.
10	Protocol Error (P): A violation of the SATA protocol was detected.
9	Persistent Communication or Data Integrity Error (C): A communication error, which was not recovered, occurred that is expected to be persistent. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.
8	Transient Data Integrity Error (T): A data integrity error occurred that was not recovered by the interface.
7:2	Reserved
1	Recovered Communications Error (M): Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of PHY synchronization, or from other causes and may be derived from the PhyNRdy signal between the PHY and Link layers.
0	Recovered Data Integrity Error (I): A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action.

22.4.33 Port x command issue register (PxCI)

22.4.33.1 Offset

Register	Offset
PxCI	138h

22.4.33.2 Function

Port x command issue register

22.4.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.33.4 Fields

Field	Function
31-0 CI	Command issued This bit is bit significant. Each bit corresponds to a command slot, where bit 0 corresponds to command slot 0. This bit is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device. When the HBA receives a FIS which clears the BSY, DRQ, and ERR bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field must only be set to 1 by software when PxCMD[ST] is set to 1. This field is also cleared when PxCMD[ST] is written from a 1 to a 0 by software.

22.4.34 Port x FIS-based switching control register (PxFBS)

22.4.34.1 Offset

Register	Offset
PxFBS	140h

22.4.34.2 Function

This register controls and obtains the status for port multiplier FIS-based switching.

22.4.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												DWE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADO				DEV				Reserved					SDE	DEC	EN
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

22.4.34.4 Fields

Field	Function
31-20 —	Reserved
19-16 DWE	Device with error Set by hardware to the value of the port multiplier port number of the device that experienced a fatal error condition. This bit is only valid when PxFBS[SDE] is set to 1.
15-12 ADO	Active device optimization This bit exposes the number of active devices that the FIS-based switching implementation has been optimized for. When there are more devices active than indicated in this bit, throughput of concurrent traffic may degrade. For optimal performance, software should limit the number of active devices based on this value. The minimum value for this bit should be 2h, indicating that at least two devices may be active with high performance maintained.
11-8 DEV	Device To issue Set by software to the port multiplier port value of the next command to issue. This field enables hardware to know the port the command is to be issued to without fetching the command header.

Table continues on the next page...

Field	Function
	Software must not issue commands to multiple port multiplier ports on the same write of the PxCI register.
7-3 —	Reserved
2 SDE	Single device error This bit is cleared on PxFBS[DEC] being set to 1 or on PxCMD[ST] being cleared to 0. 0b - When a fatal error condition has occurred, the error applies to the entire port. To clear the error, PxCMD[ST] must be cleared to 0 by software. 1b - When a fatal error condition has occurred, hardware believes the error is localized to one device such that software's first error recovery step should be to utilize the PxFBS[DEC] functionality.
1 DEC	Device error clear Software only sets this bit to 1 if PxFBS[EN] is set to 1 and PxFBS[SDE] is set to 1. 0b - When hardware has completed error recovery actions, hardware clears the bit to 0. Write to 0 by software has no effect. 1b - The HBA clears the device-specific error condition and the HBA flushes any commands outstanding for the device that experienced the error, including clearing the PxCI and PxSACT bits for that device to 0.
0 EN	Enable Software only changes the value of the EN bit when PxCMD[ST] is cleared to 0. 0b - FIS-based switching is not being used. 1b - A port multiplier is attached and the HBA uses FIS-based switching to communicate with it.

22.4.35 Port 0 BIST error register (PBERR)

22.4.35.1 Offset

Register	Offset
PBERR	170h

22.4.35.2 Function

The BIST operation as programmed by SATA must report the operation as passing or failing. This register reflects the status of the BIST operation running in the link layer.

22.4.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														BEOS	BERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

22.4.35.4 Fields

Field	Function
31-2 —	Reserved
1 BEOS	BIST error one shot bit 0b - BIST operation passed 1b - BIST operation failed
0 BERR	BIST error 0b - BIST operation passing 1b - BIST operation failing

22.5 Command layer

The operation of the command layer is defined by the AHCI specification.

22.5.1 Local port context management

When the AHCI controller is connected to a port multiplier supporting FIS-based switching, a local context store can be enabled to avoid the process of lookup of the related memory addressing for data transactions. This feature facilitates quick context switching and allows the AHCI to operate with multiple devices in a seamless manner. Each context stores information about the memory address and SATA block address of any executing command on the first four ports of a port multiplier.

22.5.2 Vendor-specific BIST operation

As part of the host self-diagnostic operation, a vendor-specific BIST mode is supported. This mode, in conjunction with a SerDes that supports serial loop-back, allows for the test of the host controller operation. When programmed, the host fetches a command from memory loop that commands FIS through the transport and the link layers and then posts the payload to the receive FIS area for checking. This mode exercises the following paths.

- DMA controller FIS transmission
- Command layer FIS transmission
- Transport layer TX FIFO FIS transmission
- Link layer FIS transmission
- PHY modes
- Link layer FIS reception
- Transport layer Rx FIFO and FIS reception
- Command layer FIS reception
- Host DMA controller FIS reception

To run this self-test, the software needs to perform the following operation:

1. Build the vendor-specific header and descriptor as detailed out in the Vendor Specific BIST Header and Descriptor. For more information, refer Table 22-12.
2. Place the PHY in the loop-back mode. For more information, refer to SerDes module chapter for the PHY loop.
3. Set PPCFG[FPR] to 1. Issue this command to the host controller.

When the host controller indicates the command has completed, the software examines the contexts of the command descriptors stats field. If the preprogrammed values are present, it considers the test passed.Vendor-specific BIST header, descriptor, and register settings are described in the subsequent sections.

22.5.2.1 Command list structure

To support the vendor BIST operation, the command header structure has been modified. The DW0[11] bit has been designated as VBIST. Setting this bit indicates that the associated command is to be used as the payload for a vendor BIST operation.

Figure 22-2. Command list structure

This table describes the fields in the command header.

Table 22-11. DW 0 - Description information

Bit	Description
31:12	Per the AHCI specification
11	VBIST Setting this bit indicates that the associated command is to be used as payload for a vendor BIST operation.
10:0	Per the AHCI specification

Table 22-12. Vendor BIST command header

DWord number	Hexadecimal value
DW0	32’h0000_0805
DW1	32’h0000_0000
DW2	System Address (usage dependent)
DW3	System Address (usage dependent)

22.5.2.2 Vendor BIST command descriptor command FIS

The A and B values can be filled in by the user according to the pattern that needs to be tested.

Table 22-13. Command descriptor command

DWord number	Hexadecimal value
DW0	32'h0001_0058
DW1	32'hAAAA_A034
DW2	32'hBBBB_B034
DW3	Reserved - 32’h0000_0000
DW4	Reserved - 32’h0000_0000
DW5	Reserved - 32’h0000_0000
DW6	Reserved - 32’h0000_0000
DW7	Reserved - 32’h0000_0000

22.5.2.3 Receive FIS area reg D2H RFIS structure

The table below provides the receive FIS area reg D2H RFIS structure.

Table 22-14. Receive FIS area reg D2H RFIS structure

DWord number	Hexadecimal Value	Hexadecimal Value
DW0	32'hAAAA_A034	32'hBBBB_B034
DW1	32'hBBBB_B034	32'hAAAA_A034
DW2	32'hAAAA_A034	32'hBBBB_B034
DW3	32'hBBBB_B034	32'hAAAA_A034
DW4	32'hAAAA_A034	32'hBBBB_B034

22.6 PhyControl BIST modes

The BIST modes within the PhyControl layer are designed to cover two main functions.

- Verification of the interface between PhyControl (controller layer) and SerDes
- Generation of limited patterns for compliance testing

The following programming steps are performed when the system software places the PhyControl into one of its BIST modes:

1. Initialize the Phy1Cfg register to place the system into a state of Force Ready and Near End Retimed Loopback, to enable PhyControl BIST Pattern and PhyControl BIST Clear Error, and to select the required BIST pattern .
2. The software brings the device online. This operation causes the controller to de-assert the reset to the PHY and allow the setup start.
3. The software inserts a delay of 10 seconds before proceeding. This is to allow the setup to stabilise and the Rx part of the BIST checker lock to the incoming pattern.
4. Once the delay has elapsed, the software clears the PhyControl BIST Clear Error bit of the register.
5. When these steps are completed, the software monitors the BIST error bit of the Port Vendor Specific registers 0. If this bit remains low the PhyControl passes BIST. If this bit is set then the PhyControl fails BIST. The bit remains set until the software clears it using the PhyControl BIST Clear Error bit of the Phy1Cfg register.

The PhyControl layer supports a number of test patterns. Each pattern is applied in an un-encoded format to the PHY which encodes 8b/10b , serialises, loopbacks, de-serialises and decodes 10b/8b.

- LBP (Lone bit pattern): The lone-bit pattern is designed to be used to generate a pattern only.
- LFTP (Low frequency test pattern): This pattern provides a low frequency, which is allowed within the Serial ATA encoding rules. Pattern 32'hD30.3_D30.3_D30.3_D30.3. The pattern is repetitive.
- MFTP (Mid frequency test pattern): This pattern provides a middle frequency which is allowed within the Serial ATA encoding rules. Pattern encoded 32'hD24.3_D24.3_D24.3_D24.3. The pattern is repetitive.
- HFTP (High frequency test pattern): This pattern provides the maximum frequency allowed within the Serial ATA encoding rules. Pattern encoded 32'hD10.2_D10.2_D10.2_D10.2. The pattern is repetitive.
- PRBS Pattern: This pattern is a long running 32 bit PRBS pattern initially seeded to 32'h1234_5678; The PRBS algorithm is defined as: {PRBSTest[30:0], (PRBSTest[31] ^ PRBSTest[30] ^ PRBSTest[29] ^ PRBSTest[28] ^ PRBSTest[27] ^ PRBSTest[21])};
- BIST pattern: This pattern provides the counting pattern that allows maximum combinations of characters be tested. It consists of Pattern:
  - 32'hD0.0_ D1.0_D2.0_D3.0
  - 32'hD4.0_ D5.0_D6.0_D7.0
  - 32'hD8.0_ D9.0_D10.0_D11.0
  - 32'hD12.0 D13.0_D14.0_D15.0.
  - 32'hD28.7_ D29.7_D30.7_D31.7

BIST pattern transmission.

1. Seven ALIGN primitives are transmitted to force the Rx path to lock to the transmitted data.
2. A special lock primitive 32'h4A_4A_4A_7C is transmitted to start the pattern match on the Rx side.
3. The selected pattern is transmitted for 256 byte intervals and then the circuit starts from state 1.

22.7 PHY control configuration register OOB timing setup

In SATA systems three out of band (OOB) signals COMRESET, COMINIT, and COMWAKE are used during link initialization. These are achieved by transmission of either a burst of four Gen1 ALIGNp primitives or a burst composed of sixteen D24.3 characters, each burst having a duration of 160 UIOOB (106.67 ns). The individual bursts are followed by idle periods (at common-mode levels), having durations as depicted in the figure and the table below. The duration of the idle period is used to decode the type of OOB signal.

Figure 22-3. OOB signaling timing

Parameter	Units	Limit	Specification
UIOOB, UI During OOB Signaling	Ps	Nominal	666.67
COMINIT/COMRESET and COMWAKE Transmit Burst Length	Ns	Nominal (T1)	106.7
COMINIT/COMRESET Transmit Gap Length	Ns	Nominal (T2)	320.0
COMWAKE Transmit Gap Length	Ns	Nominal (T1)	106.7

Table continues on the next page...

COMWAKE Gap Detection Windows	Ns	May Detect Shall detect Shall not detect	35 ≤ T < 175 101.3 ≤ T ≤ 112 T < 35 or T ≥175
COMINIT/COMRESET Gap Detection	Ns	May Detect Shall detect Shall not detect	175 ≤ T < 525 304 ≤ T ≤ 336 T < 175 or T ≥ 525

22.8 Modifications to the AHCI standard PRDT entry

In order to support PRDT control of both the AW/RCACHE and AW/RPROT, the DW2 field of the standard PDRT entry has a new bit designation as follows:

Table 22-15. DW 2 – PRDT Re-designation for AXI operation

Bit	Description
31:00	Reserved
Re-designation	
31:18	Reserved
17	PROT bits controlled from PRD (Sub Control)
16	CACHE bits controlled from PRD (Sub Control)
15:11	Reserved
10:8	This is the value driven onto A[R/W]PROT when the AXI master is posting a data burst read / write address associated with this PRD entry to the memory controller
7:4	This is the value driven onto A[R/W]CACHE when the AXI master is posting a data burst read / write address associated with this PRD entry to the memory controller
3:2	Reserved
1	Inject a CRC error into the associated Received Data FIS
0	Inject a CRC error into the associated Transmitted Data FIS

When a PRDT entry is processed to move data through the AXI master, first the global control field of either the AXI Cache Control register or AXI PROT Control register bits 12 and 28 are examined to determine the values for both the PROT and CACHE control signals. If the global control indicates that the values should be controlled from the PRDT then the sub control determines which values control the PROT and CACHE signals.

Table 22-16. Global control field

Global	Sub	AR/WPROT / AR/WCACHE
0	0	AXI Cache/PROT Control Register

Table continues on the next page...

Table 22-16. Global control field (continued)

Global	Sub	AR/WPROT / AR/WCACHE
0	1	AXI Cache/PROT Control Register
1	0	AXI Cache/PROT Control Register
1	1	DW 2 – PRDT Re-designation for AXI operation

22.9 Transport layer architectural overview

The function of the SATA transport layer is to interface between the command and link layers in the transmission and reception of the Frame Information Structures (FIS).

During transmission, the transport layer frames the FIS placed into the TX FIFO. The FISs are framed based on a programmed length for non-data FIS and/or a configurable length for data FIS. When the transport layer is instructed to send a non-data FIS, it employs a retry policy until the far end signals accepts the transmitted FIS.

On reception, the transport layer de-frames the FISes and places these into the RX FIFO. When an FIS is received, the transport layer informs the command layer.

For a non-data FIS, the FIS is considered received when an EOF is signaled by the link layer and FIS is received with a good CRC.

For a short vendor-specific FIS, the FIS is considered as a non-data FIS.

For longer vendor-specific FIS, the FIS reception is signaled when the RX FIFO reaches its water mark.

For a data FIS, the FIS is considered received when the First DWord (header) is written into the FIFO.

The transport layer is responsible for crossing the clock domain between the transport layer txDWord and the rxDWord clocks and command layer clock domain. The receive FIFO is written to the transport layer receive DWord clock with data contained in the FIS sent by the link layer. Once the data is stable at the output of the receive FIFO on the command layer clock domain, the command layer can take the data. If the command layer is not ready to accept the data, the data builds up in the receive FIFO. When the receive FIFO exceeds its threshold, the transport layer stalls the link layer, which in turn sends HOLD primitives to the far end to stall it. This threshold takes into consideration the latency involved in getting the far end to stop transmitting the data. This threshold is programmable to allow for the use of high latency repeaters or re-timers in between the host and the device.

The transmit FIFO is written to the command layer clock, with data to be sent in the FIS transferred by the DMA controller. Once the data is stable at the output of the transmit FIFO on the transmit DWord clock domain, the link layer can take the data. If the transmit FIFO cannot supply data to the link layer, the transport layer stalls the link layer, which in turn sends HOLD primitives to the far end to stall it.

22.10 Link layer overview

22.10.1 General operation

The function of the SATA link layer is to interface between the transport and PhyCtrl layers in the transmission and reception of frames and primitives. The link layer utilizes the two unidirectional links provided by the SATA interface to maintain coordinated communication between the host and the device. Payload data can only be transmitted in one direction at a time.

On transmit, the link layer first communicates with the peer far end link layer to determine if it is ready to receive. Assuming the far end link layer can receive data, the local link layer begins to take data in the form of DWords from its transport layer. It inserts Start-of-Frame (SOF) before the start of the data portion of a frame, calculates and inserts the CRC after the data portion of a frame, and inserts the End-of-Frame (EOF) primitive at the end. The link layer scrambles the contents of the frame, including the calculated CRC, but excluding the SOF and EOF delimiters, and any other embedded primitives. The 8B/10B encoding of the data is done either in the PHY control layer or in the PHY layer, depending on the nature of the PHY employed. At the end of the transmission, the link layer reports transmission status to the transport layer.

On receive, the link layer first acknowledges its readiness to receive its peer link layer. Then it awaits reception of the SOF primitive that marks the start of the received data. Following detection of the SOF primitive, the link layer proceeds to accept the incoming data. The 8B/10B decoding of the data is done either in the PHY control layer or in the PHY layer. After this, the link layer removes all primitives including the SOF and EOF delimiters. It then descrambles the contents of the frame. The link layer also calculates the CRC on the incoming frame between the SOF and EOF delimiters, and compares this calculated value to the received value. Any mismatch is reported to the transport layer. During frame reception, disparity or code errors are reported to the command layer, and appropriate action is taken in the link layer. The descrambled and decoded receive data stream is passed to the transport layer as the frame is being received. Finally at the end of the frame, the link layer reports reception status to the transport layer.

The link layer also partakes in flow control between the local and remote ends. The layer supports flow control actions based on the local FIFO status (which is located in the transport layer), or in response to receiving flow control messages from the remote end.

The transmit side of the link layer is also responsible for inserting a pair of ALIGN primitives every 254 DWords, or more frequently, if programmed by the user.

22.10.2 List of functions

The link layer is composed of a number of functions, as listed below:

- Link layer state machines
- Frame content scrambler and descrambler
- CRC generation and checking
- Bus interfaces to PhyCtrl and transport layer
- CONT primitive processing
- ALIGN insertion on transmit
- Debug functionality
- BIST support

22.10.3 Link layer state machines

Functionally, there are four link layer state machines. These are listed below and described in the subsequent sub-sections:

- Link idle state machine
- Transmit state machine
- Receive state machine
- Power mode change state machine

22.10.3.1 Link idle state machine

The link idle state machine performs the following functions:

- It is responsible for detecting a transmit request from the transport layer, or a frame reception request from the far end, and arbitrating if these two events coincide. The SATA specification defines that the host end always backs down in this case.

- It interprets power mode change requests from both transport and PhyCtrl layers, and initiates actions to enable the power mode change.
- It also detects negation and assertion of PHY_READY from the PhyCtrl layer and notifies the transport layer of the change.

22.10.3.2 Transmit state machine

The transmit state machine manages frame transmission to the PhyCtrl layer. The state machine places the Start-of-Frame (SOF) and End-of-Frame (EOF) headers on each frame, calculates the CRC and inserts it before the EOF delimiter. Between the SOF and CRC markers, the link layer accepts the current DWord from the transport layer and uses this as the next DWord of the frame. The link layer also inserts a pair of ALIGN primitives every 254 DWords of the frame data. Finally, at the end of the frame transmission, the state machine waits for status from the far end link layer through the received R_OK or R_ERR primitives. If the far end receives the frame correctly, the local link layer signals TX_OK to the transport layer; else, it signals TX_NOT_OK to the transport layer.

This state machine also partakes in the flow control actions if necessary, during packet transmission. If the transport layer cannot supply a new DWord and the frame is not finished, the transmit state machine responds by sending HOLD primitives until such time as the transport layer is ready with valid frame data. Also, during frame transmission, if the state machine detects a received HOLD primitive from the PHY Layer, it interrupts the current frame transmission and sends HOLDA primitives to the PHY to be transmitted to the far end.

The current frame transmission can only be aborted by two events.

- On reception of a DMAT primitive from the far end. In this case, the link layer state machine stops the current transfer and calculates and inserts the current CRC. This is a controlled termination.
- When the transport layer wishes to send a control register frame, signaled through TRANSMIT_CRF.

If at any point in the frame transmission process, the link layer detects error conditions, it signals these to the command layer. The errors can occur if:

- PHY_READY negates
- The link layer receives SYNC primitives during frame transmission

22.10.3.3 Receive state machine

The receive state machine is responsible for frame reception from the PHY layer. It removes the Start-of-Frame (SOF) and End-of-Frame (EOF) headers and other primitives from each frame, calculates the CRC and compares it to the received CRC. Between the SOF and CRC markers, the link layer accepts the current DWord from the PhyCtrl layer and uses this as the next DWord of the frame, transferring it to the transport layer. At the end of the frame reception, if the calculated CRC is not the same as the received CRC, the link layer signals an error to the transport layer. This is done through RX_CRC_OK and RX_CRC_NOT_OK statuses. During frame reception, if no errors are detected, the link layer transmits R_IP primitives to the far end peer link layer. Finally, at the end of the frame reception, the link layer sends the R_OK primitive if no error is detected during reception. If an error is detected, it sends an R_ERR primitive instead.

The receive state machine also partakes in flow control actions if necessary, during FIS reception. If the transport layer cannot accept a new DWord, (because its receive FIFO has reached its watermark level), and the FIS is not finished, the receive state machine responds by sending HOLD primitives on the back channel until such time as the transport layer is ready to accept FIS data again. Also, during FIS reception, if the state machine detects a received HOLD primitive from the far end, it responds by sending HOLDA primitives to the far end.

The current frame reception can be interrupted if the transport layer wishes to send a control register frame, signaled through TRANSMIT_CRF.

If at any point in the frame reception process, the link layer detects error conditions, it signals these to the command layer. The errors can occur if

- PHY_READY negates
- The link layer receives SYNC primitives
- The link layer receives WTRM primitives before EOF

22.10.3.4 Power mode change state machine

This state machine handles change of power mode requests. These requests can come from the command layer Super Set registers or the far end. The command layer signals a change request by asserting CFG_LINK_GO_PARTIAL or CFG_LINK_GO_SLUMBER. This state machine responds by transmitting PMREQ_P/ PMREQ_S primitives to the far end and waiting for PMACK primitives from it in response. Once PMACK is received, the state machine instructs the PHY layer to enter either partial or slumber state, by driving PHY_GO_PARTIAL or PHY_GO_SLUMBER signals. These signals must remain active for the duration of the new power mode.

A write of '1' to the CFG_LINK_WAKEUP bit or reception of a COMWAKE from the far end will initiate a resume to active power mode.

Note: CFG_LINK_WAKEUP should be set back to '0' once PhyReady is asserted again.

If the link layer receives a PMREQ_P/PMREQ_S primitive from a peer link layer, and is enabled to perform power management modes (CFG_EN_PARTIAL/SLUMBER = '1'), it responds by sending at least four PMACK primitives, and then asserts PHY_GO_PARTIAL or PHY_GO_SLUMBER to the PhyCtrl block. A write of '1' to the CFG_LINK_WAKEUP bit or reception of a COMWAKE from the far end initiates a resume to the active power mode.

Note: CFG_LINK_WAKEUP should be set back to '0' once PhyReady is asserted again.

If the link layer receives an XRDY primitive from the far-end while it is in the partial or slumber state, it returns to idle and signals a link sequence error to the command layer, that is LINK_SEQ_ERROR = 1. Automatic partial to slumber transitions can be set by programming the CAP2 register (offset 24h)

22.10.4 Frame content scrambler and descrambler

Two separate scramblers are used in serial ATA - one for the data payload and another for repeated primitive suppression.

The contents of each DWord of data (excluding all primitives) between SOF and EOF must be scrambled before 8B/10B encoding. Scrambling is performed on DWord quantities according to the following polynomial:

G(X) = X16 + X15 + X13 + X4 + 1

The scrambler is initialized with a seed value of 16'hFFFF at each SOF transmission and rolls over every 2048 DWords. Payload data is scrambled prior to transmission, by XORing the data to be transmitted with the output of this scrambler.

If a CONT primitive is transmitted, the intervening data between the last CONT primitive and a subsequent primitive must be scrambled as well. This scrambler uses the same polynomial as defined above for data payload scrambling and is reset to the initial value upon detection of a COMINIT or COMRESET event (. PHY_GOT_CIOR = '1'). If a CONT primitive is transmitted or received during a frame transfer, the current data payload scrambler value at the last DWord is held.

When payload data is received by the link layer it is de-scrambled by XORing it with the output of its descrambler. The descrambler is re-seeded at the beginning of the received data payload, that is at each SOF reception. The descrambler uses the same polynomial as the scrambler.

22.10.5 CRC generator and checker

A 32 bit CRC is calculated on the data contents of each frame, and is inserted in the DWord before the EOF. The CRC covers all data bytes in the frame excluding any primitives, such as SOF, EOF, HOLD, HOLDA, DMAT, SYNC, X_RDY, R_RDY or ALIGNs.

The CRC generator works on DWord quantities. Any padding to the boundary is done in the transport layer. The polynomial used for the CRC is as follows:

G(X) = X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1

The CRC is initialized with a seed value of 32'h52325032 at each SOF.

The CRC generation or checking does not apply to primitives (as stated above) or to CONT'ed primitives. If a CONT primitive is transmitted or received, the intervening data between the last CONT primitive and a subsequent primitive is not included in the CRC calculation for a frame. If this happens during a frame transfer, the current CRC scrambler value at the last DWord is held.

22.10.6 8B/10B encode and decode

All data and primitives must be encoded prior to transmission on the line. The standard implementation has the 8B/10B encoder and decoder located in the PHY control layer.

22.10.7 CONT primitive processing

The link layer is capable of replacing repetitive primitive streams with scrambled data, by use of the CONT primitive. This reduces EMI emissions, as primitives are not scrambled.

The link layer can transmit a CONT primitive at a point where it knows it must transmit a number of repeating primitives. After a CONT primitive has been transmitted, the link layer transmits scrambled junk data to the PHY layer. The content of this junk data is disregarded. At the far end link layer, the reception of a CONT primitive implies that the last received valid primitive will be repeated until it receives the next valid non-align primitive. Transmission of a new valid primitive halts the current CONT processing; reception of a new valid non-align primitive halts the current CONT processing.

This action can occur on transmit and receive. The link layer supports both transmission and reception of CONT primitives. The scrambler and the CRC generator/checker actions are changed during the transmission and reception of CONT sequences as described in Frame content scrambler and descrambler and CRC generator and checker.

Transmission of CONT primitives are not allowed until the link layer state machine has initialized correctly, and has transmitted 16 non-align primitives.

For diagnostic purposes CONT insertion can be suppressed in the transmit path; the receive path can always receive those.

22.10.8 ALIGN insertion

The link layer is responsible for ALIGN insertion and removal at a fixed frequency. A pair of ALIGN primitives are inserted into the transmit data stream every 254 DWords. These primitives are used in the PHY layer to maintain character alignment and to help in elastic buffering, if required.

At the receive end, the ALIGN primitives are stripped from the incoming data stream in the link layer.

For diagnostic purposes, the rate of ALIGNs can be increased as much as two ALIGNS per one DWord, that is: ALIGN, ALIGN, data, ALIGN, ALIGN.

Also, the CFG_SEND_4_ALIGNS bit can be set to instruct the link layer to send four aligns at a time, instead of two.

22.10.9 Debug functionality

There are a number of useful features designed into the ink layer to aid debug. These include:

- A configuration bit in the command layer (CFG_TX_BADCRC) can be set to force the link layer to transmit a bad CRC at the end of a frame. One bad CRC is transmitted every time the bit is configured. To force transmission of a new bad CRC, the configuration bit needs to be cleared and written to again.
- A configuration bit in the command layer (CFG_RX_BADCRC) can be set to force the link layer to detect a bad CRC value at the end of a frame. Only one bad CRC is be detected for one setting. The configuration bit must be cleared and written to again to force detection of another bad CRC.

- A configuration bit in the command layer (CFG_TXPRIM_JUNK) can be set to force the junk data that is transmitted after a CONT primitive to be equal to 32'hDEADBEEF.
- The align insertion rate can be increased using the CFG_ALIGN_RATE register field in the command layer.
- Four error counters can be monitored by issuing register reads to the command layer. These error counters include: Disparity error counter, Code error counter, PHY Internal error counter and Control Character error counter (see signal descriptions for detail).
- A number of configuration bits in the command layer can be used to override normal primitive insertion. For example,
  - Set CFG_PRIM_OVR_STATE = the L_SendHold state (16)
  - Set CFG_PRIM = 32'hb5b5957c , that is a SYNC primitive
  - During the transfer, set CFG_PRIM_OVR_EN = 1. When the link layer detects a rising edge on CFG_PRIM_OVR_EN, it inserts one SYNC primitive into the datastream in place of the HOLD, when the LINK_STATE reaches the L_SendHold state. Only one HOLD primitive is overridden - the CFG_PRIM_OVR_EN must be cleared and written to again to force another override to occur.

22.10.10 BIST support

The transmit and receive sub-blocks of the link layer contain logic to support BIST activate FIS functionality.

When a BIST activate FIS is either received or transmitted successfully by the transport layer, it issues a request to the link layer to enter BIST mode (LINK_BIST_REQ = 1). This forces the link layer to enter a BIST state in its state machine as soon as it receives a SYNC primitive from the far-end. When the BIST state is entered, an acknowledge signal is sent to the transport layer, that is LINK_BIST_ACK = 1. In the BIST state, the link layer transmits a data sequence as specified by the two BIST data patterns in the BIST activate FIS. The link layer also monitors the incoming data from the PhyCtrl block to detect if the BIST data pattern is the same as specified in the BIST activate FIS. When it detects the correct data sequence, the BIST_ERR output is de-asserted. This signal stays de-asserted unless an error occurs in the datastream from the far-end. The BIST_ERR output is sent to the command layer from where it can be read by the host software. The link layer also outputs a signal called BIST_ACTIVE to the command layer. This singnal is asserted when the link layer enters its BIST state and de-asserted when BIST mode is exited due to a COMRESET being received.

22.11 PHY control layer overview

The PHY control layer operates between the SerDes and link layers.

The main functions of the PHY control layer are

- Data path operation
  - Rx data path
  - Tx data path
- PHY Initialization state machine
  - Out of band processing
  - Speed negotiation

On receive, the PHY Control layer converts the encoded 20-bit or decoded 16-bit parallel data from the SerDes to a 32-bit DWord, which it presents to the link layer. The PHY control layer aligns the control word of the SATA primitive to the lowest word position of the DWord. The two main configurations of the PHY control layer operate on either encoded data or decoded data, as sourced from the SerDes. In the case of encoded data, the PHY control contains an 8b/10b decoder function and decodes the incoming data into data, control/data and code or disparity error. In the case of decoded data, the PHY control sources data, control/data and code or disparity error from the SerDes.

The receive DWord clock is generated from the receive core clock outputted by the SerDes. This is generated as part of the DWord alignment process.

On transmit, the PHY control layer takes in the 32-bit transmit data from the link layer and converts the data into encoded 20-bit or un-encoded 16-bit parallel data from the SerDes. The control/data bit from the link layer (which is always assumed to be associated with the lowest byte position of the transmit DWord) is also passed onto the SerDes with the appropriate word. The PHY control layer takes the transmit word clock outputted by the SerDes and converts it to a DWord transmit clock which it sends to the link layer.

The PHY control layer implements the PHY initialization state machine. The state machine implements the device PHY initialization state machine as defined in the Serial ATA Revision 3.1 June 2, 2009.

22.12 Reset

The SATA AHCI controller supports three levels of reset:

- Software reset - a single device on one of the ports is reset, but the HBA and physical communication remain intact. This is the least intrusive.
- Port reset – the physical communication between the HBA and device on a port are disabled. This is more intrusive.
- HBA reset – the entire HBA is reset, and all ports are disabled. This is the most intrusive.

When an HBA or port reset occurs, PHY communication is re-established with the device through a COMRESET followed by the normal out-of-band communication sequence defined in SATA. At the end of the reset, the device, if working properly, will send a D2H Register FIS, which contains the device signature. When the HBA receives this FIS, it updates the PxTFD[STS] and PxTFD[ERR] bits, and updates the PxSIG register with the signature.

Software is recommended to follow a staggered reset approach, starting from a less intrusive reset mechanism and then using a more intrusive reset mechanism only if the less intrusive mechanism does not succeed in clearing the condition.

22.12.1 Software reset

Legacy software contained a standard mechanism for generating a reset to a SATA device – setting the SRST (software reset) bit in the Device Control register. SATA has a more robust mechanism called COMRESET, also referred to as port reset. A port reset is the preferred mechanism for error recovery and should be used in place of software reset.

To issue a software reset in AHCI, software builds two H2D Register FISes in the command list. The first Register FIS has the SRST bit set to '1' in the Control field of the Register FIS, the 'C' bit is set to '0' in the Register FIS, and the command table has the CH[R] (reset) and CH[C] (clear BSY on R_OK) bits set to '1'. The CH[R] (reset) bit causes the HBA to perform a SYNC escape if necessary to put the device into an idle condition before sending the software reset. The CH[C] (clear BSY on R_OK) bit needs to be set for the first Register FIS to clear the BSY bit and proceed to issue the next Register FIS since the device does not send a response to the first Register FIS in a software reset sequence. The second Register FIS has the SRST bit set to '0' in the Control field of the Register FIS, the 'C' bit is set to '0' in the Register FIS, and the command table has the CH[R] (reset) and CH[C] (clear BSY on R_OK) bits cleared to '0'.

Refer to the Serial ATA Revision 2.6 specification for more information on the software reset FIS sequence.

Table 22-17. Register FIS host to device SRST On

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	Features								Command								C	R	R	R	PM Port				FIS Type (27h)									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	32'h	00000027
1	Device								LBA High								LBA Mid								LBA Low									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000
2	Features (exp)								LBA High (exp)								LBA Mid (exp)								LBA Low (exp) (0)									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000
3	Control								Reserved (0)								Sector Count (exp)								Sector Count									
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	04000000
4	Reserved (0)								Reserved (0)								Reserved (0)								Reserved (0)									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000

Table 22-18. Register FIS host to device SRST Off

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	Features								Command								C	R	R	R	PM Port				FIS Type (27h)									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	32'h	00000027
1	Device								LBA High								LBA Mid								LBA Low									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000
2	Features (exp)								LBA High (exp)								LBA Mid (exp)								LBA Low (exp) (0)									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000
3	Control								Reserved (0)								Sector Count (exp)								Sector Count									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000
4	Reserved (0)								Reserved (0)								Reserved (0)								Reserved (0)									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32'h	00000000

When issuing a software reset sequence, there should not be other commands in the command list. Before issuing the software reset, software must clear PxCMD[ST], wait for the port to be idle (PxCMD[CR] = '0'), and then re-set PxCMD[ST].

PxTFD[STS[BSY]] and PxTFD[STS[DRQ]] must be cleared prior to issuing the reset. If PxTFD[STS[BSY]] or PxTFD[STS[DRQ]] is still set based on the failed command, then a port reset should be attempted or command list override (PxCMD[CLO]) should be used if supported. Note that a device-to-host FIS from a previously failed command may be received after the PxCMD[ST] bit has been cleared and/or re-set. It is recommended that the HBA ignore such a FIS or SYNC Escape that FIS (as directed by the CH[R] bit being set in the first software reset H2D Register FIS).

22.12.2 Port reset

If a port is not functioning properly after a software reset, software may attempt to re-initialize communication with the port through a COMRESET. It must first clear PxCMD[ST] and wait for PxCMD[CR] to clear to '0' before re-initializing communication. However, if PxCMD[CR] does not clear within a reasonable time (500 milliseconds), it may assume the interface is in a hung condition and may continue with issuing the port reset.

Software causes a port reset (COMRESET) by writing 4'h1 to the PxSCTL[DET] bit to invoke a COMRESET on the interface and start a re-establishment of PHY layer communications. Software shall wait at least 1 millisecond before clearing PxSCTL[DET] to 4'h0; this ensures that at least one COMRESET signal is sent over the interface. After clearing PxSCTL[DET] to 4'h0, software should wait for Serial ATA AHCI 1.3 Specification 101 communication to be re-established as indicated by bit 0 of PxSSTS[DET] being set to 4'h1. Then software should write all 1s to the PxSERR register to clear any bits that were set as part of the port reset.

When PxSCTL[DET] is set to 4'h1, the HBA shall reset PxTFD[STS] to 8'h7F and shall reset PxSSTS[DET] to 4'h0. When PxSCTL[DET] is set to 4'h0, upon receiving a COMINIT from the attached device, PxTFD[STS[BSY]] shall be set to 1'h1 by the HBA.

22.12.3 HBA reset

If the HBA becomes unusable for multiple ports, and a software reset or port reset does not correct the problem, software may reset the entire HBA by setting GHC[HR] to '1'. When software sets the GHC[HR] bit to '1', the HBA shall perform an internal reset action. The bit shall be cleared to '0' by the HBA when the reset is complete. A software write of '0' to GHC[HR] shall have no effect. To perform the HBA reset, software sets GHC[HR] to '1' and may poll until this bit is read to be '0', at which point software knows that the HBA reset has completed.

If the HBA has not cleared GHC[HR] to '0' within 1 second of software setting GHC[HR] to '1', the HBA is in a hung or locked state.

When GHC[HR] is set to '1', GHC[AE], GHC[IE], the IS register, and all port register fields (except PxFB/PxFBU/PxCLB/PxCLBU) that are not HwInit in the HBA's register memory space are reset. The HBA's configuration space and all other global registers/ bits are not affected by setting GHC[HR] to '1'. Any HwInit bits in the port specific registers are not affected by setting GHC[HR] to '1'. The port specific registers PxFB, PxFBU, PxCLB, and PxCLBU are not affected by setting GHC[HR] to '1'. If the HBA supports staggered spin-up, the PxCMD[SUD] bit will be reset to '0'; software is responsible for setting the PxCMD[SUD] and PxSCTL[DET] fields appropriately such that communication can be established on the SATA link. If the HBA does not support staggered spin-up, the HBA reset shall cause a COMRESET to be sent on the port.

Chapter 23
SerDes Module

23.1 SerDes Module as Implemented on LS1088A

23.1.1 SerDes 1 and SerDes 2 Protocol Notation

The following table shows the supported networking protocol options for SerDes modules 1 and 2. The following notation conventions are used in the table:

- XFI notation (also supporting 10GBASE-KR): XFIm means XFI (1 lane @ 10.3125 Gbaud)
  - "m" indicates which MAC on the WRIOP
  - For example, "XFI1," indicates XFI using MAC 1.
- SGMII notation:
  - SGm means SGMII (1 lane @ 1.25 Gbaud)
    - "m" indicates which MAC
    - For example, "sg1" indicates SGMII for MAC 1.
  - *SGm* means SGMII (1 lane @ 3.125 Gbaud)
    - "m" indicates which MAC
    - For example, "*SG1*" indicates SGMII for MAC 1
- QSGMII notation: QSG.a means QSGMII (1 lane @ 5 Gbps)
  - "a" indicates MACs 3-6, 'b' indicates MACs 7-10.
- PCIe notation:
  - PCIem x is PCIe @ 5/2.5 GT/s or 8/5/2.5 GT/s, m indicates the PCIe controller number and x the associated width configuration.
  - PCIe lanes may operate at up to a Gen 3 rate (8 Gbps); otherwise, only Gen 2 (5 Gbps) or Gen 1 (2.5 Gbps) is available.
  - Operating rate depends on maximum rate selection and link training.
- Per Lane PLL mapping
  - 1 = PLL1

- 2 = PLL2
- x = not applicable

- SATA notation: SATA indicates SATA operating at 6, 3 or 1.5 Gbps depending on the software rate selection.

23.1.2 SerDes options

NOTE

The “Protocol Number” is a protocol combination identifier provided to ease communication about the table but is not used for SerDes protocol configuration. RCW fields are used to select the protocol (SRDS_PRTCL_S1_LN3,2,1,0).

NOTE

The WRIOP minimum frequency specifications provided below are based on a 28 Gbps at 600 MHz WRIOP performance target assuming linear performance scaling with frequency. The platform minimum frequency specification of 500 MHz is the limit in most cases.

Table 23-1. Networking SerDes 1

Protocol Number (decimal)	RCW[SRDS_PRTCL_S1_LN0]	RCW[SRDS_PRTCL_S1_LN1]	RCW[SRDS_PRTCL_S1_LN2]	RCW[SRDS_PRTCL_S1_LN3]	0 Lane D	1 Lane C	2 Lane B	3 Lane A	PLL mapping	EC1 RGMII (MAC4) available	EC2 RGMII (MAC5) available	SerDes 1 throughput (Gbps)	WRIOP min frequency (MHz)	SerDes 1 + EC1 + EC2 (Gbps)	WRIOP min frequency (MHz)
0	0	0	0	0						Y	Y	0	500	2	500
18	3	3	3	3	SG2	SG1	SG7	SG3	1111	Y	Y	4	500	6	500
21	1	1	3	3	XFI2	XFI1	SG7	SG3	2211	Y	Y	22	500	24	515
22	1	3	3	3	XFI2	SG1	SG7	SG3	2111	Y	Y	13	500	15	500
23	2	3	3	3	***SG2***	SG1	SG7	SG3	2111	Y	Y	5.5	500	7.5	500
24	2	2	3	3	***SG2***	***SG1***	SG7	SG3	2211	Y	Y	7	500	9	500
25	1	1	4	3	XFI2	XFI1	QSGb	SG3	2211	Y	Y	25	536	27	579
26	1	3	4	3	XFI2	SG1	QSGb	SG3	2111	Y	Y	16	500	18	500

Table continues on the next page...

Table 23-1. Networking SerDes 1 (continued)

27	2	3	4	3	***SG2***	SG1	QSG b	SG3	2111	Y	Y	8.5	500	10.5	500
28	2	2	4	3	***SG2***	***SG1***	QSG b	SG3	2211	Y	Y	10	500	12	500
29	1	1	4	4	XFI2	XFI1	QSG b	QSG a	2211	N	N	28	600	28	600
30	1	3	4	4	XFI2	SG1	QSG b	QSG a	2111	N	N	19	500	19	500
31	2	3	4	4	***SG2***	SG1	QSG b	QSG a	2111	N	N	11.5	500	11.5	500
32	2	2	4	4	***SG2***	***SG1***	QSG b	QSG a	2211	N	N	13	500	13	500
53	3	3	4	3	SG2	SG1	QSG b	SG3	1111	Y	Y	7	500	9	500
54	3	3	4	4	SG2	SG1	QSG b	QSG a	1111	N	N	10	500	10	500
55	1	1	5	3	XFI2	XFI1	PCIe 1	SG3	2211	Y	Y	21	500	23	500
56	2	2	5	3	***SG2***	***SG1***	PCIe 1	SG3	2211	Y	Y	6	500	8	500
57	4	3	5	3	QSG b	SG1	PCIe 1	SG3	1111	Y	Y	6	500	8	500
58	3	3	5	3	SG2	SG1	PCIe 1	SG3	1111	Y	Y	3	500	5	500

Table 23-2. Non Networking SerDes 2

Protocol Number (Decimal)	RCW[SRDS_PRTCL_S2_LN0]	RCW[SRDS_PRTCL_S2_LN1]	RCW[SRDS_PRTCL_S2_LN2]	RCW[SRDS_PRTCL_S2_LN3]	Lane A	Lane B	Lane C	Lane D	PCIe Gen 3	PLL mapping PCIe Gen 3	PLL mapping PCIe Gen 2
12	8	8	8	8	PCIe1 x4				Y	1111	2222
13	5	5	5	9	PCIe1 x1	PCIe2 x1	PCIe3 x1	SATA	N		2221
14	7	7	6	9	PCIe1 x2		PCIe2 x1	SATA	N		2221
19	7	7	7	7	PCIe1 x2		PCIe3 x2		Y	1111	2222
20	5	5	7	7	PCIe1 x1	PCIe2 x1	PCIe3 x2		Y	1111	2222
59	5	5	0	6	PCIe1 x1	PCIe2 x1		PCIe3 x1	Y	1111	2222

Table continues on the next page...

Table 23-2. Non Networking SerDes 2 (continued)

60	0	5	0	6		PCIe2 x1		PCIe3 x1	Y	1111	2222
61	0	5	5	9		PCIe2 x1	PCIe3 x1	SATA	N		2221

23.1.3 Networking SerDes Protocol Lane Configuration

Table 23-3. Networking SerDes Protocol Lane Configuration

RCW[SRDS_PRTCL_S1_LN#]	Lane # Protocol Selection	0 D	1 C	2 B	3 A
0	x	x	x	x	x
1	XFI 10.3125 Gbaud	XFI2	XFI1		
2	SGMII 3.25 Gbaud	*SG2*	*SG1*		
3	SGMII 1.25 Gbaud	SG2	SG1	SG7	SG3
4	QSGMII 5 Gbps	QSGb		QSGb	QSGa
5				PCIe1	
6					
7					
8					
9-F	Reserved				

23.1.3.1 Configuration of Networking SerDes

SerDes 1 Configuration:

- Protocol(s): Selected using RCW[SRDS_PRTCL_S1_LN0,1,2,3]
- PLLs: Enabled using RCW[SRDS_PLL_PD_PLL1,2] PLL Reference Clock: RCW[SRDS_PLL_REF_CLK_SEL_S1]
- Maximum PCIe rate: Selected using RCW[SRDS_DIV_PEX_S1]
- Networking (non-PCIe) rates are determined by the protocol selection.

23.1.3.2 Rules for disabling a networking SerDes module

If SerDes 1 and/or 2 is not used, then it must be disabled as described in Disabling Unused SerDes Modules.

23.1.3.3 Rules for Selecting Valid Networking SerDes Options

A valid configuration for SerDes 1 is subject to the following:

- Simultaneous selection of QSGMII.7, 8, 9, 10 on lanes 0 and 2 is not allowed.
- Rules restricting PLL powerdown selection
  - SerDes 1's PLL1 must be powered down using SRDS_PLL_PD_S1 when "IP Protocol Number" is any of the following values: 0
  - SerDes 1's PLL2 must be powered down using SRDS_PLL_PD_S1 when "IP Protocol Number" is any of the following values: 0, 18, 53, 54, 57, or 58.
- Simultaneous selection of PCIe.1 on SerDes1 and SerDes2 is not allowed.

NOTE

Note that SerDes 1's PLL powerdowns are directly controlled by RCW[SRDS_PLL_PD_S1] only.

23.1.3.4 Rules for using PCIe1 on SerDes1

When PCIe1 x1 is selected for use on SerDes1, PCIe1 must not be selected for use on SerDes2. Explicitly this means that when RCW[SRDS_PRTL_S1_LN2]=0x5 (PCIe1 x1 on SerDes1 lane 2), the following SerDes2 RCW values are not legal.

- RCW[SRDS_PRTL_S2_LN0] = 0x5, 0x7, 0x8
- RCW[SRDS_PRTL_S2_LN1] = 0x6, 0x7, 0x8
- RCW[SRDS_PRTL_S2_LN2] = 0x8
- RCW[SRDS_PRTL_S2_LN3] = 0x8

23.1.3.5 RGMII Interfaces

In addition to the SerDes-based networking interfaces, two RGMII interfaces are also available for two MACs. The EC1 RGMII interface is enabled using RCW's EC1 field. The EC2 RGMII interface is enabled using RCW's EC2 field.

When RGMII is enabled for a particular MAC, it always takes precedence over SerDes-based interfaces.

Figure 23-1. RGMII Interfaces

23.1.3.6 Rules for disabling networking SerDes PLLs when not all lanes are needed

When not all of the SerDes1 lanes are needed, it is permissible to power down the unused lanes which can be accomplished in software. The PLL associated with the powered-down lanes may be disabled based on the PLL lane mapping provided in SerDes options

23.1.4 Non Networking SerDes Protocol Lane Configuration

Table 23-4. Non Networking SerDes Protocol Lane Configuration

RCW[SRDS_PRTCL_S2_LN#]	Lane # Protocol Selection	0 A	1 B	2 C	3 D
0	x	x	x	x	x
1					
2					
3					
4					
5	PCIe x1	PCIe1 x1	PCIe2 x1	PCIe3 x1	
6	PCIe x1		PCIe1 x1	PCIe2 x1	PCIe3 x1
7	PCIe x2	PCIe1 x2	PCIe1 x2	PCIe3 x2	PCIe3 x2
8	PCIe x4	PCIe1 x4	PCIe1 x4	PCIe1 x4	PCIe1 x4
9	SATA				SATA
A-F	Reserved				

23.1.4.1 Configuration of Non-Networking SerDes

The available non-networking protocols available for use are shown in Table 14-5. Correct configuration involves selecting the correct values for the SerDes RCW fields.

SerDes 2 Configuration:

- Protocol(s): Selected using RCW[SRDS_PRTCL_S2_LN0,1,2,3]
- PLLs: Enabled using RCW[SRDS_PLL_PD_S2]
- PLL Reference Clock: RCW[SRDS_PLL_REF_CLK_SEL_S2]
- Maximum rate: Selected using the following:
  - PCIe: RCW[SRDS_DIV_PEX_S2]

  SATA rate selection is done in software

23.1.4.2 Rules for Selecting Valid Non Networking SerDes Options

A valid configuration for SerDes 2 is subject to the following:

- Simultaneous selection of PCIe.1 on SerDes1 and SerDes2 is not allowed.
- Rules restricting PLL powerdown selection
  - SerDes 2's PLL1 must be powered down using SRDS_PLL_PD_S2 when "IP protocol Number" is any of the following values: 0
  - SerDes 2's PLL2 must be powered down using SRDS_PLL_PD_S2 when PCIe Gen3 maximum link training rates are not selected and protocol value is 0C, 13, 14, 3B or 3C.
  - SerDes 2's PLL2 must be powered down using SRDS_PLL_PD_S2 when protocol value is 0.

NOTE

Note that SerDes 2's PLL powerdowns are directly controlled by RCW[SRDS_PLL_PD_S2] only.

23.1.5 Disabling Unused SerDes Modules

It is possible to independently disable each SerDes module, S1-S2, by disabling all PLLs associated with the SerDes module.

SerDes 1 is disabled as follows:

- SRDS_PLL_PD_S1 = 2'b11 (both PLLs configured as powered down)
- SRDS_PLL_REF_CLK_SEL_S1 = 2'b00
- SRDS_PRTCL_S1_LN3,2,1,0 = {4'h0,4'h0,4'h0,4'h0}(no other values are permitted when both PLLs are powered down)

SerDes 2 is disabled as follows:

- SRDS_PLL_PD_S2 = 2'b11 (both PLLs configured as powered down)
- SRDS_PLL_REF_CLK_SEL_S2 = 2'b00
- SRDS_PRTCL_S2_LN3,2,1,0 = {4'h0,4'h0,4'h0,4'h0}(no other values are permitted when both PLLs are powered down)

23.1.6 Powering Down Unused SerDes Lanes

Powering down of SerDes data lanes is only supported in software using registers in the SerDes Control Block. Also see the RCW bit definition for SRDS_PLL_PD_S1 for how lanes can be powered down based on the powering down of individual PLLs.

23.1.7 Valid Reference Clocks and PLL Configurations for SerDes Protocols

Each SerDes protocol allows for a finite set of valid SerDes-related RCW fields and reference clock frequencies, as shown in Table 23-5.

Table 23-5. Valid SerDes Reference Clocks and RCW Encodings

SerDes Protocol (given lane)	Valid reference clock frequency	Valid setting as determined by SRDS_PRTCL_S*n*	Valid setting as determined by SRDS_PLL_REF_CLK_SEL_S*n*	Valid setting as determined by SRDS_DIV_[prot]_S*n*
Networking Interfaces				
SGMII (1.25 Gbps)	100 MHz	SGMII @ 1.25 Gbps	0: 100 MHz	Don't Care
	125 MHz		1: 125 MHz	
2.5x SGMII (3.125 Gbps)	125 MHz	SGMII @ 3.125 Gbps	0: 125 MHz	Don't Care
	156.25 MHz		1: 156.25 MHz	
QSGMII (5.0 Gbps)	100 MHz	Any QSGMII	0: 100 MHz	0 : 5 Gbps
	125 MHz		1: 125 MHz	
XFI (10.3125 Gbps) or 10GBASE-KR	156.25 Mhz	XFI @ 10.3125 Gbps	0: 156.25 MHz	Don't Care
Non Networking Interfaces				
PCI Express 2.5 Gbps	100 MHz[1]	Any PCIe	0: 100 MHz	2'b10: 2.5 G
(doesn't negotiate upwards)	125 MHz[1]		1: 125 MHz	
PCI Express 5 Gbps	100 MHz[1]	Any PCIe	0: 100 MHz	2'b01: 5.0 G
(can negotiate up to 5 Gbps)	125 MHz[1]		1: 125 MHz	
PCI Express 8 Gbps	100 MHz[1]	Any PCIe	0: 100 MHz	2'b00: 8.0 G
(can negotiate up to 8 Gbps)	125 MHz[1]		1: 125 MHz	
SATA (1.5, 3 or 6 Gbps)	100 MHz	Any SATA	0: 100 MHz	Don't Care[2]
	125 MHz		1: 125 MHz	

1. A spread-spectrum reference clock is permitted for PCI Express, provided that common SerDes reference clock must be used for both link partners. Whenever the spread-spectrum clocking is used for the SerDes reference clock of a single PCI Express controller, the same SerDes PLL must not be used concurrently for any other protocols including another PCI Express controller.
2. SerDes lanes configured as SATA initially operate at 3.0 Gbps. 1.5 Gbps operation may be enabled later through the SATA IP itself.

23.1.8 Ethernet MAC Capabilities

The eWRIOP supports 10 Ethernet MACs which support interface protocols and MACSEC as summarized in the table below.

Table 23-6. MAC Capabilities

MAC	RGMII 1 Gbps	SGMII 1 Gbps	SGMII 2.5 Gbps	QSGMII 5 Gbps	XFI 10 Gbps	MACSEC Supported
1	-	Y	Y	-	Y	-
2	-	Y	Y	-	Y	-
3	-	Y	-	MACs 3-6	-	Y
4	Y	-	-		-	Y
5	Y	-	-		-	Y
6	-	-	-		-	Y
7	-	Y	-	MACs 7-10	-	-
8	-	-	-		-	-
9	-	-	-		-	-
10	-	-	-		-	-

23.1.9 Protocol Converter to MAC and Interface Mapping

Table 23-7. Protocol Converter to MAC and Interface Mapping

SerDes1 PCVTR	WRIOP MAC	SerDes1 Interface	SerDes Protocol Table
XFIa	MAC1	XGMIIa	XFI.1
XFIb	MAC2	XGMIIb	XFI.2
SGMIIa	MAC3	GMIIa	SGMII.3
SGMIIb	MAC7	GMIIb	SGMII.7
SGMIIc	MAC1	GMIIc	SGMII.1
SGMIId	MAC2	GMIId	SGMII.2
QSGMIIa	MAC3,4,5,6	GMIIefgh	QSGMII.3,4,5,6
QSGMIIb	MAC7,8,9,10	GMIIijkl	QSGMII.7,8,9,10

23.1.10 Non Networking Protocol to Interface Mapping

Table 23-8. Non Networking Protocol to Interface Mapping

SerDes 2 PCVTR	SerDes 2 Protocol Table
PCIea	PCIe1
PCIeb	PCIe2
PCIec	PCIe3
SATAa	SATA

23.2 Overview

The SerDes module implements link serialization/deserialization and PCS functions for high speed serial interfaces from 1.25 Gbaud to 10.3125 Gbaud.

23.2.1 Features

The SerDes module includes 4 data lanes and 2 PLLs and supports the following protocols:

- PCI Express 3.0 (November, 2010) @ 2.5, 5, 8 Gbaud
- SGMII ENG-46158, Revision 1.9 (July, 2009) @ 1.25, 3.125 Gbaud
- QSGMII EDCS-540123, Revision 1.2 (September, 2007) @ 5 Gbaud
- IEEE 802.3-2015
    - 1000Base-KX @1.25 Gbaud
    - XFI @ 10.3125 Gbaud
    - 10GBase-KR @ 10.3125 Gbaud
- Serial ATA Revision 3.0 (June, 2009) @ 1.5, 3.0, 6.0 Gbaud

23.3 Modes of Operation

The SerDes supports several combinations of protocols and frequencies. See SerDes Module as Implemented on LS1088A, for details on the combinations.

23.4 External Signals Description

The table below lists the external signals for the SerDes.

Table 23-9. SerDes Interface Signals

Pin Name	Description	No. of Signals	I/O
SDn_TX[0:3]_P	Transmitter serial output, positive data	4	O
SDn_TX_B[0:3]_N	Transmitter serial output, negative data	4	O
SDn_IMP_CAL_TX	Tx Impedance Calibration	1	I
SDn_RX[0:3]_P	Receiver serial output, positive data	4	I

Table continues on the next page...

Table 23-9. SerDes Interface Signals (continued)

Pin Name	Description	No. of Signals	I/O
SDn_RX_B[0:3]_N	Receiver serial output, negative data	4	I
SDn_IMP_CAL_RX	Rx Impedance Calibration	1	I
x=1, 2			
SDn_REFx_CLK_P	Reference clock input to PLLx	2	I
SDn_REFx_CLK_N	Reference clock-bar input to PLLx	2	I

The table below shows the mapping of external interface pins to register lane numbering.

Table 23-10. SD1 Lane Mapping

Lane	External Interface Pins
3	SD1_TX3_P/SD1_TX3_N + SD1_RX3_P/SD1_RX3_N
2	SD1_TX2_P/SD1_TX2_N + SD1_RX2_P/SD1_RX2_N
1	SD1_TX1_P/SD1_TX1_N + SD1_RX1_P/SD1_RX1_N
0	SD1_TX0_P/SD1_TX0_N + SD1_RX0_P/SD1_RX0_N

Table 23-11. SD2 Lane Mapping

Lane	External Interface Pins
0	SD2_TX0_P/SD_TX0_N + SD2_RX0_P/SD_RX0_N
1	SD2_TX1_P/SD_TX1_N + SD2_RX1_P/SD_RX1_N
2	SD2_TX2_P/SD_TX2_N + SD2_RX2_P/SD_RX2_N
3	SD2_TX3_P/SD_TX3_N + SD2_RX3_P/SD_RX3_N

23.5 SerDes register descriptions

The SerDes module is programmed by control/status registers (CSRs). The CSRs are used for mode control, and to extract status information. All accesses to and from the registers must be made as 32-bit accesses. There is no support for accesses of sizes other than 32 bits. Writes to reserved register bits must always preserved the previous value; that is, the register should be programmed by reading the value, modifying appropriate fields, and writing back the value. Unless otherwise specified, the read value of unmapped registers or of reserved bits in mapped registers is not defined, and must not be assumed to be 0.

The table below lists the address, name, and a cross-reference to the complete description of each register. The offsets to the memory map table are defined for each SerDes module from 0x0000 to 0x1FFF (8 KB).

Unlisted register addresses are reserved.

- Reserved fields are always ignored for the purposes of determining access type.
- Reserved registers and fields must be preserved on writes.
- R (read only) indicates that all the non-reserved fields in a register are read-only.
- R/W (read/write) indicates all of the non-reserved fields in a register are either read/write or read-only, with at least one read/write field. The register description indicates which fields are read-write and which read-only.
- RC (read clear) indicates all of the fields in a register are cleared on read, are are not otherwise writeable.
- W1C indicates that all of the non-reserved fields in a register are cleared by writing ones to them.
- Mixed indicates a combination of access types.
- Special is used when no other category applies. In this case the register figure and field description provide the details for access.

This section describes the control, status, and test registers for SerDes and Protocol PCS layers.

All reserved register fields must be preserved on register writes (read-modified-write).

23.5.1 SerDes memory map

SerDes1 base address: 1EA_0000h

SerDes2 base address: 1EB_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	SerDes PLL1 Reset Control Register (PLL1RSTCTL)	32	RW	0467_452Fh
4h	SerDes PLL1 Control Register 0 (PLL1CR0)	32	RW	8080_0008h
8h	SerDes PLL1 Control Register 1 (PLL1CR1)	32	RW	0800_4100h
18h	SerDes PLL1 Control Register 5 (PLL1CR5)	32	RW	0000_0000h
20h	SerDes PLL2 Reset Control Register (PLL2RSTCTL)	32	RW	0467_452Fh
24h	SerDes PLL2 Control Register 0 (PLL2CR0)	32	RW	8080_0008h
28h	SerDes PLL2 Control Register 1 (PLL2CR1)	32	RW	0800_4100h
38h	SerDes PLL2 Control Register 5 (PLL2CR5)	32	RW	0000_0000h
90h	SerDes Transmit Calibration Control Register (TCALCR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
94h	SerDes Transmit Calibration Control Register 1 (TCALCR1)	32	RW	0000_0000h
A0h	Receive Calibration Control Register (RCALCR)	32	RW	0000_0000h
A4h	SerDes Receive Calibration Control Register 1 (RCALCR1)	32	RW	0000_0000h
B0h	General Control Register 0 (GR0)	32	RW	0000_0000h
100h	Lane A Protocol Select Status Register 0 (LNAPSSR0)	32	RO	0000_0000h
120h	Lane B Protocol Select Status Register 0 (LNBPSSR0)	32	RO	0000_0000h
140h	Lane C Protocol Select Status Register 0 (LNCPSSR0)	32	RO	0000_0000h
160h	Lane D Protocol Select Status Register 0 (LNDPSSR0)	32	RO	0000_0000h
200h	Protocol Configuration Register 0 (PCCR0)	32	RW	0000_0000h
208h	Protocol Configuration Register 2 (PCCR2)	32	RW	0000_0000h
220h	Protocol Configuration Register 8 (PCCR8)	32	RW	0000_0000h
224h	Protocol Configuration Register 9 (PCCR9)	32	RW	0000_0000h
22Ch	Protocol Configuration Register B (PCCRB)	32	RW	0000_0000h
300h	PCIe Equalization Configuration Register (PEXEQCR)	32	RW	0000_0C10h
304h	PCIe Equalization Preset 0 Register (PEXEQP0CR)	32	RW	0000_C000h
308h	PCIe Equalization Preset 1 Register (PEXEQP1CR)	32	RW	0000_8000h
30Ch	PCIe Equalization Preset 2 Register (PEXEQP2CR)	32	RW	0000_A000h
310h	PCIe Equalization Preset 3 Register (PEXEQP3CR)	32	RW	0000_6000h
314h	PCIe Equalization Preset 4 Register (PEXEQP4CR)	32	RW	0000_0000h
318h	PCIe Equalization Preset 5 Register (PEXEQP5CR)	32	RW	0000_0005h
31Ch	PCIe Equalization Preset 6 Register (PEXEQP6CR)	32	RW	0000_0006h
320h	PCIe Equalization Preset 7 Register (PEXEQP7CR)	32	RW	0000_9005h
324h	PCIe Equalization Preset 8 Register (PEXEQP8CR)	32	RW	0000_6006h
328h	PCIe Equalization Preset 9 Register (PEXEQP9CR)	32	RW	0000_0008h
32Ch	PCIe Equalization Preset 10 Register (PEXEQP10CR)	32	RW	0000_A006h
800h	General Control Register 0 - Lane A (LNAGCR0)	32	RW	1104_0000h
804h	General Control Register 1 - Lane A (LNAGCR1)	32	RW	004C_4011h
80Ch	Speed Switch Control Register 0 - Lane A (LNASSCR0)	32	RW	9828_2B00h
810h	Receive Equalization Control Register 0 - Lane A (LNARECR0)	32	RW	0000_001Fh
814h	Receive Equalization Control Register 1- Lane A (LNARECR1)	32	RO	0000_0008h
818h	Transmit Equalization Control Register 0 - Lane A (LNATECR0)	32	RW	1028_3000h
81Ch	Speed Switch Control Register 1- Lane 0 (LNASSCR1)	32	RW	0000_0000h
820h	TTL Control Register 0 - Lane A (LNATTLCR0)	32	RW	0000_0400h
83Ch	Test Control/Status Register 3 - Lane A (LNATCSR3)	32	RW	0400_0000h
840h	General Control Register 0 - Lane B (LNBGCR0)	32	RW	1104_0000h
844h	General Control Register 1 - Lane B (LNBGCR1)	32	RW	004C_4011h
84Ch	Speed Switch Control Register 0 - Lane B (LNBSSCR0)	32	RW	9828_2B00h
850h	Receive Equalization Control Register 0 - Lane B (LNBRECR0)	32	RW	0000_001Fh
854h	Receive Equalization Control Register 1- Lane B (LNBRECR1)	32	RO	0000_0008h
858h	Transmit Equalization Control Register 0 - Lane B (LNBTECR0)	32	RW	1028_3000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
85Ch	Speed Switch Control Register 1- Lane 0 (LNBSSCR1)	32	RW	0000_0000h
860h	TTL Control Register 0 - Lane B (LNBTTLCR0)	32	RW	0000_0400h
87Ch	Test Control/Status Register 3 - Lane B (LNBTCSR3)	32	RW	0400_0000h
880h	General Control Register 0 - Lane C (LNCGCR0)	32	RW	1104_0000h
884h	General Control Register 1 - Lane C (LNCGCR1)	32	RW	004C_4011h
88Ch	Speed Switch Control Register 0 - Lane C (LNCSSCR0)	32	RW	9828_2B00h
890h	Receive Equalization Control Register 0 - Lane C (LNCRECR0)	32	RW	0000_001Fh
894h	Receive Equalization Control Register 1- Lane C (LNCRECR1)	32	RO	0000_0008h
898h	Transmit Equalization Control Register 0 - Lane C (LNCTECR0)	32	RW	1028_3000h
89Ch	Speed Switch Control Register 1- Lane 0 (LNCSSCR1)	32	RW	0000_0000h
8A0h	TTL Control Register 0 - Lane C (LNCTTLCR0)	32	RW	0000_0400h
8BCh	Test Control/Status Register 3 - Lane C (LNCTCSR3)	32	RW	0400_0000h
8C0h	General Control Register 0 - Lane D (LNDGCR0)	32	RW	1104_0000h
8C4h	General Control Register 1 - Lane D (LNDGCR1)	32	RW	004C_4011h
8CCh	Speed Switch Control Register 0 - Lane D (LNDSSCR0)	32	RW	9828_2B00h
8D0h	Receive Equalization Control Register 0 - Lane D (LNDRECR0)	32	RW	0000_001Fh
8D4h	Receive Equalization Control Register 1- Lane D (LNDRECR1)	32	RO	0000_0008h
8D8h	Transmit Equalization Control Register 0 - Lane D (LNDTECR0)	32	RW	1028_3000h
8DCh	Speed Switch Control Register 1- Lane 0 (LNDSSCR1)	32	RW	0000_0000h
8E0h	TTL Control Register 0 - Lane D (LNDTTLCR0)	32	RW	0000_0400h
8FCh	Test Control/Status Register 3 - Lane D (LNDTCSR3)	32	RW	0400_0000h
1000h	PEXA Protocol Control Register 0 (PEXACR0)	32	RW	2000_0000h
1040h	PEXB Protocol Control Register 0 (PEXBCR0)	32	RW	2000_0000h
1080h	PEXC Protocol Control Register 0 (PEXCCR0)	32	RW	2000_0000h
1804h	SGMIIA Protocol Control Register 1 (SGMIIACR1)	32	RW	0000_0000h
180Ch	SGMIIA Protocol Control Register 3 (SGMIIACR3)	32	RO	0000_0000h
1814h	SGMIIB Protocol Control Register 1 (SGMIIBCR1)	32	RW	0000_0000h
181Ch	SGMIIB Protocol Control Register 3 (SGMIIBCR3)	32	RO	0000_0000h
1824h	SGMIIC Protocol Control Register 1 (SGMIICCR1)	32	RW	0000_0000h
182Ch	SGMIIC Protocol Control Register 3 (SGMIICCR3)	32	RO	0000_0000h
1834h	SGMIID Protocol Control Register 1 (SGMIIDCR1)	32	RW	0000_0000h
183Ch	SGMIID Protocol Control Register 3 (SGMIIDCR3)	32	RO	0000_0000h
1884h	QSGMIIA Protocol Control Register 1 (QSGMIIACR1)	32	RW	0000_0000h
188Ch	QSGMIIA Protocol Control Register 3 (QSGMIIACR3)	32	RO	0000_0000h
1894h	QSGMIIB Protocol Control Register 1 (QSGMIIBCR1)	32	RW	0000_0000h
189Ch	QSGMIIB Protocol Control Register 3 (QSGMIIBCR3)	32	RO	0000_0000h
1984h	XFIA Protocol Control Register 1 (XFIACR1)	32	RW	0000_0000h
198Ch	XFIA Protocol Control Register 3 (XFIACR3)	32	RO	0000_0000h
1994h	XFIB Protocol Control Register 1 (XFIBCR1)	32	RW	0000_0000h
199Ch	XFIB Protocol Control Register 3 (XFIBCR3)	32	RO	0000_0000h

23.5.2 SerDes PLLa Reset Control Register (PLL1RSTCTL - PLL2RSTCTL)

23.5.2.1 Offset

Register	Offset
PLL1RSTCTL	0h
PLL2RSTCTL	20h

23.5.2.2 Function

PLL*n*RSTCTL contains control and status bits for the SerDes PLL *n* reset state machine.

This register should not be modified unless RST_ERR=1 or RST_DONE=1

23.5.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSTREQ	RST_DONE	RST_ERR	Reserved												
W																
Reset	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								PLLRST_B	SDRST_B	SDEN	Reserved				
W																
Reset	0	1	0	0	0	1	0	1	0	0	1	0	1	1	1	1

23.5.2.4 Fields

Field	Function
31 RSTREQ	Reset request PLL Reset Request - write 1 self-clearing. Software setting of 1 initiates a soft reset of SerDes PLL *n*, along with all lanes using PLL *n*. Hardware automatically clears this bit before reset is done. Note: This field must be 0 if PLL2 does not have a reference clock. Clearing this bit during the reset sequence has no effect. 0b - Not requesting PLL soft reset. 1b - Requesting PLL soft reset.
30 RST_DONE	Reset done PLL Reset Done from Control Block State Machine 0b - PLL reset sequence in progress. 1b - PLL reset sequence complete.
29 RST_ERR	Reset Error No PLL Lock before counter time_out 0b - Normal function 1b - PLL lock didn't happen in the expected time period
28-8 —	Reserved
7 PLLRST_B	PLL reset. Resets PLLn PLL reset. Resets PLLn . Default value: 0 The SerDes reset state machine overrides this field to 0 in some states. Setting PLLRST_B=0 when RST_DONE=0 forces a SerDes PLL reset regardless of the reset state machine state, and may cause the reset state machine to time out and set RST_ERR=1. 0b - Force PLL reset 1b - Application Mode, unless RSTREQ=1
6 SDRST_B	SRDS group reset. Resets all lanes operating from PLLn The SerDes reset state machine overrides this field to 0 in some states. 0b - Force SerDes group reset 1b - Application Mode, unless RSTREQ=1
5 SDEN	SerDes enable. Enable the section of the SerDes module clocked by PLLn SerDes enable. Enable the section of the SerDes module clocked by PLLn. Default value: 1 The SerDes reset state machine overrides this field to 0 in some states. Setting SDEN=0 when RST_DONE=0 forces a SerDes powerdown regardless of the reset state machine state, and may cause the reset state machine to time out and set RST_ERR=1. Note: This field must be 0 if PLL2 does not have a reference clock. 0b - Force SerDes power down 1b - Application Mode, unless RSTREQ=1
4-0	Reserved

Field	Function
—	

23.5.3 SerDes PLLa Control Register 0 (PLL1CR0 - PLL2CR0)

23.5.3.1 Offset

Register	Offset
PLL1CR0	4h
PLL2CR0	24h

23.5.3.2 Function

PLLnCR0 contains control and status bits for SerDes PLL *n*

NOTE

This register may be automatically updated when PCI Express is active on the SerDes. Only write to this register when all associated PCI Express controllers are inactive.

23.5.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	POFF	RFCLK_SEL			RFCLK_EN	Reserved			PLL_LCK	Reserved	Reserved		FRATE_SEL			
Reset	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved														DLYDIV_SEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

23.5.3.4 Fields

Field	Function
31 POFF	PLL off Power down an unused PLL Default value: 0 0b - PLL on 1b - PLL off. All lanes referencing this PLL must also be disabled.
30-28 RFCLK_SEL	Reference clock frequency select All others reserved Default value set by reset configuration. Recommended setting per protocol: • SATA: 000, 001 or 011 • PCIe: 000 or 001 • SGMII @ 3.125: 001 or 010 • SGMII/1000Base-KX @ 1.25: 000 or 001 • QSGMII: 000, 001 or 010 • XFI/10GBase-KR: 010 or 100 000b - 100 MHz 001b - 125 MHz 010b - 156.25 MHz 011b - 150 MHz
27 RFCLK_EN	Reference clock observe enable Default value: 0 Recommended setting: 0 0b - Disable reference clock output 1b - Enable buffered version of SD_REF_CLKn
26-24 —	Reserved
23 PLL_LCK	PLL lock Indicates PLL(n) has calibrated and locked 0b - PLL is not locked 1b - PLL is locked
22 —	Reserved
21-20 —	Reserved
19-16 FRATE_SEL	Select frequency of PLL VCO. All lanes within a group must operate at a multiple of this frequency and within specified limits of the protocol(s). Default value set by reset configuration. Required settings per protocol: All others are reserved 0000b - 5.00 GHz ---------- [SGMII QSGMII PCIe] 0110b - 5.15625 GHz ----- [XFI] 0111b - 4.00 GHz ---------- [PCIe after gen3 initialization]

Table continues on the next page...

Field	Function
	1001b - 3.125 GHz -------- [SGMII] 1010b - 3.00 GHz ---------- [SATA]
15-2 —	Reserved
1-0 DLYDIV_SEL	Select PLLn_ex_dly_clk divider value This feature is used for: • 1000Base-KX When that mode is not active, DLYDIV_SEL should be 00. 00b - PLLn_ex_dly_clk off 01b - PLLn_ex_dly_clk = FRATE_SEL/16 (e.g. 312.5 MHz for 5.0 GHz)

23.5.4 SerDes PLLa Control Register 1 (PLL1CR1 - PLL2CR1)

23.5.4.1 Offset

Register	Offset
PLL1CR1	8h
PLL2CR1	28h

23.5.4.2 Function

PLLnCR1 contains control bits for SerDes PLL *n*.

23.5.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				PLLBW_SEL	Reserved						Reserved				
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

23.5.4.4 Fields

Field	Function
31-28 —	Reserved
27 PLLBW_SEL	Select Higher PLL(n) Bandwidth Recommended setting per PLL: 1 0b - Nominal PLL Bandwidth 1b - PLL Bandwidth Setting
26-21 —	Reserved
20-0 —	Reserved

23.5.5 SerDes PLLa Control Register 5 (PLL1CR5 - PLL2CR5)

23.5.5.1 Offset

Register	Offset
PLL1CR5	18h
PLL2CR5	38h

23.5.5.2 Function

PLLnCR5 contains control bits for the SerDes x PLL *n*.

23.5.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SEL_REFCLK_AMP_DIS	Reserved			LEFT_REF_BUF_EN	Reserved		RIGHT_REF_BUF_EN	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.5.4 Fields

Field	Function
31 SEL_REFCLK_AMP_DIS	Disables the reference clock amplifier and selects usage of the reference clock coming from the left, either from SoC or from another 10G Serdes PLL Default value set by reset
30-28 —	Reserved
27 LEFT_REF_BUF_EN	Enable for left directed reference clock buffer
26-25 —	Reserved
24 RIGHT_REF_BUF_EN	Enable for right directed reference clock buffer
23-0 —	Reserved

23.5.6 SerDes Transmit Calibration Control Register (TCALCR)

23.5.6.1 Offset

Register	Offset
TCALCR	90h

23.5.6.2 Function

TCALCR contains the control bits used for Transmit Calibration.

23.5.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CALRST_B	Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.6.4 Fields

Field	Function
31-28 —	Reserved
27 CALRST_B	Reset the transmit calibration During POR (warm reset sequence) it is active and is released when the first PLL comes out of reset. 0b - Reset 1b - Application Mode
26-0 —	Reserved

23.5.7 SerDes Transmit Calibration Control Register 1 (TCALCR1)

23.5.7.1 Offset

Register	Offset
TCALCR1	94h

23.5.7.2 Function

TCALCR1 contains the bits used for reference clock controls in the right half-lane.

23.5.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				DIG_OUT_REFCLK_EN	Reserved		ANA_OUT_REFCLK_EN	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.7.4 Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

Field	Function
27 DIG_OUT_REF CLK_EN	Enable CMOS digital buffered version of selected ref_clk to the right. Used in conjunction with PLLn_CR5[RIGHT_REF_BUF_EN]
26-25 —	Reserved
24 ANA_OUT_REF CLK_EN	Enable CML analog buffered version of selected ref_clk to the left. Used in conjunction with PLLn_CR5[RIGHT_REF_BUF_EN].
23-0 —	Reserved

23.5.8 Receive Calibration Control Register (RCALCR)

23.5.8.1 Offset

Register	Offset
RCALCR	A0h

23.5.8.2 Function

RCALCR contains the control bits for Receive Calibration.

23.5.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CALRST_B	Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.8.4 Fields

Field	Function
31-28 —	Reserved
27 CALRST_B	Reset the receiver calibration During POR (warm reset sequence) it is active and is released when the first PLL comes out of reset 0b - Reset 1b - Application Mode
26-0 —	Reserved

23.5.9 SerDes Receive Calibration Control Register 1 (RCALCR1)

23.5.9.1 Offset

Register	Offset
RCALCR1	A4h

23.5.9.2 Function

RCALCR1 contains the bits for reference clock controls on the left half-lane.

23.5.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ANA_IN_REFCLK_BUF_EN	Reserved			DIG_OUT_REFCLK_EN	Reserved		ANA_OUT_REFCLK_EN	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														BI_REFCTL_OUT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.9.4 Fields

Field	Function
31 ANA_IN_REFCLK_BUF_EN	Enable CML analog buffer used to transition SoC reference clock to 10G Serdes
30-28 —	Reserved
27 DIG_OUT_REFCLK_EN	Enable CMOS digital buffered version of selected ref_clk to the left. Used in conjunction with PLLn_CR5[LEFT_REF_BUF_EN].
26-25 —	Reserved
24 ANA_OUT_REFCLK_EN	Enable CML analog buffered version of selected ref_clk to the left. Used in conjunction with PLLn_CR5[LEFT_REF_BUF_EN].
23-2 —	Reserved

Table continues on the next page...

Field	Function
1-0 BI_REFCTL_OUT	Output COP DFT/BurnIn/JTAG reference clock controls to wrapper

23.5.10 General Control Register 0 (GR0)

23.5.10.1 Offset

Register	Offset
GR0	B0h

23.5.10.2 Function

GR0 contains general control/status bits for the SerDes.

23.5.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	XPAD_SEL	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.10.4 Fields

Field	Function
31-15	Reserved

Table continues on the next page...

Field	Function
—	
14 XPAD_SEL	Describes to SerDes module the value of the power supply being used by the Serdes I/Os: Full SerDes reset required after changing this value. **Warning**: setting XPAD_SEL to low xpadvdd supply when the supply is > 1.35V+5% may cause irreparable damage to the device. 0b - High xpadvdd supply (nominal 1.5V - see H/W spec for details) 1b - Low xpadvdd supply (nominal 1.35V - see H/W spec for details)
13-0 —	Reserved

23.5.11 Lane a Protocol Select Status Register 0 (LNAPSSR0 - LNDPSSR0)

23.5.11.1 Offset

Register	Offset
LNAPSSR0	100h
LNBPSSR0	120h
LNCPSSR0	140h
LNDPSSR0	160h

23.5.11.2 Function

LNmPSSR0 contains decoded protocol select information per lane.

The primary usage of this register is to map an Ethernet link to its corresponding MAC and MDIO address space.

23.5.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	TYPE							Reserved				MAC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					PCS			Reserved				LANE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.11.4 Fields

Field	Function
31 —	Reserved
30-24 TYPE	Procotol Type Bits 1-5 are same as LNmGCR0[PROTS]. Bits 6-7: • QSGMII: 01 • All others: 00
23-20 —	Reserved
19-16 MAC	MAC instance Note: if TYPE[6:7]=QSGMII, then the MAC instance listed is the MAC connected to port 0, which is also the MAC used for MDIO accesses. 0000b - MAC1 0001b - MAC2 0010b - MAC3 0011b - MAC4 0100b - MAC5 0101b - MAC6 0110b - MAC7 0111b - MAC8 1000b - MAC9 1001b - MAC10
15-11 —	Reserved
10-8 PCS	PCS instance of TYPE within PHY 000b - PCSa/1

Table continues on the next page...

Field	Function
	001b - PCSb/2 010b - PCSc/3 011b - PCSd/4
7-4 —	Reserved
3-0 LANE	Lane number within PCS Example 1: for x4 PCIe [3:0] on lanes [3:0], LANE=0 for n=0, LANE=1 for n=1, LANE=2 for n=2 and LANE=3 for n=3 Note that the lane number within PCS does not take into account lane reversal, either via auto-negotiate, or by S/W control in SerDes registers. All others reserved 0000b - lane 0 0001b - lane 1 0010b - lane 2 0011b - lane 3

23.5.12 Protocol Configuration Register 0 (PCCR0)

23.5.12.1 Offset

Register	Offset
PCCR0	200h

23.5.12.2 Function

PCCR0 contains the protocol configuration for PCIe.

23.5.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	PEXA_CFG			Reserved	PEXB_CFG			Reserved	PEXC_CFG			Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.12.4 Fields

Field	Function
31 —	Reserved
30-28 PEXA_CFG	PEXa Configuration All others reserved Default value set by RCW configuration. 000b - disabled 001b - x1 on Lane 1 010b - x4 [3:0] on Lanes [0:3] (SerDes #2 only) 011b - x1 on Lane 0 (SerDes #2 only) 100b - x2 [1:0] on Lanes [0:1] (SerDes #2 only)
27 —	Reserved
26-24 PEXB_CFG	PEXb Configuration SerDes #2 only All others reserved Default value set by RCW configuration. 000b - disabled 001b - x1 on Lane 2 010b - x1 on Lane 1
23 —	Reserved
22-20 PEXC_CFG	PEXc Configuration SerDes #2 only All others reserved

Table continues on the next page...

Field	Function
	Default value set by RCW configuration. 000b - disabled 001b - x1 on Lane 3 010b - x1 on Lane 2 011b - x2[1:0] on Lanes [2:3]
19 —	Reserved
18-16 —	Reserved
15-0 —	Reserved

23.5.13 Protocol Configuration Register 2 (PCCR2)

23.5.13.1 Offset

Register	Offset
PCCR2	208h

23.5.13.2 Function

PCCR2 contains the protocol configuration for SATA and Aurora.

23.5.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	SATAA_CFG			Reserved	Reserved			Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved			Reserved								Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.13.4 Fields

Field	Function
31 —	Reserved
30-28 SATAA_CFG	SATAa Configuration SerDes #2 only All others reserved 000b - Disabled 001b - x1 on lane 3
27 —	Reserved
26-24 —	Reserved
23-15 —	Reserved
14-12 —	Reserved
11-4 —	Reserved
3 —	Reserved
2-0 —	Reserved

23.5.14 Protocol Configuration Register 8 (PCCR8)

23.5.14.1 Offset

Register	Offset
PCCR8	220h

23.5.14.2 Function

PCCR8 contains the protocol configuration for SGMII /1000Base-KX.

23.5.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	SGMIIA_KX	SGMIIA_CFG			SGMIIB_KX	SGMIIB_CFG			SGMIIC_KX	SGMIIC_CFG			SGMIID_KX	SGMIID_CFG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	Reserved			Reserved	Reserved			Reserved	Reserved			Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.14.4 Fields

Field	Function
31 SGMIIA_KX	SGMIIb 1000Base-KX Configuration: Note: this configuration bit must be set before performing any MDIO initialization of the PCS. 0b - SGMII mode 1b - 1000Base-KX mode

Table continues on the next page...

Field	Function
30-28 SGMIIA_CFG	SGMIIa Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - disabled 001b - Enabled on Lane 0 to WRIOP Mac 3
27 SGMIIB_KX	SGMIIc 1000Base-KX Configuration: Note: this configuration bit must be set before performing any MDIO initialization of the PCS. 0b - SGMII mode 1b - 1000Base-KX mode
26-24 SGMIIB_CFG	SGMIIb Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - disabled 001b - Enabled on Lane 1 to WRIOP Mac 7
23 SGMIIC_KX	SGMIId 1000Base-KX Configuration: Note: this configuration bit must be set before performing any MDIO initialization of the PCS. 0b - SGMII mode 1b - 1000Base-KX mode
22-20 SGMIIC_CFG	SGMIIc Configuration SerDes #1 only All others reserved Default value set by RCW configuration 000b - disabled 001b - Enabled on Lane 2 to WRIOP Mac 1
19 SGMIID_KX	SGMIIe 1000Base-KX Configuration: Note: this configuration bit must be set before performing any MDIO initialization of the PCS. 0b - SGMII mode 1b - 1000Base-KX mode
18-16 SGMIID_CFG	SGMIId Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - disabled 001b - Enabled on Lane 3 to WRIOP Mac 2
15 —	Reserved
14-12 —	Reserved
11	Reserved

Table continues on the next page...

Field	Function
—	
10-8 —	Reserved
7 —	Reserved
6-4 —	Reserved
3 —	Reserved
2-0 —	Reserved

23.5.15 Protocol Configuration Register 9 (PCCR9)

23.5.15.1 Offset

Register	Offset
PCCR9	224h

23.5.15.2 Function

PCCR9 contains the Protocol Configuration for QSGMII.

23.5.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	QSGMIIA_CFG			Reserved	QSGMIIB_CFG			Reserved	Reserved			Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.15.4 Fields

Field	Function
31 —	Reserved
30-28 QSGMIIA_CFG	QSGMIIa Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - disabled 001b - Enabled on Lane 0 connected to WRIOP MACs 3,4,5,6
27 —	Reserved
26-24 QSGMIIB_CFG	QSGMIIb Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - disabled 001b - Enabled on Lane 1 connected to WRIOP MACs 7,8,9,10 010b - Enabled on Lane 3 connected to WRIOP MACs 7,8,9,10
23 —	Reserved
22-20 —	Reserved
19	Reserved

Table continues on the next page...

Field	Function
—	
18-16 —	Reserved
15-0 —	Reserved

23.5.16 Protocol Configuration Register B (PCCRB)

23.5.16.1 Offset

Register	Offset
PCCRB	22Ch

23.5.16.2 Function

PCCRB contains the protocol configuration for XFI /10GBase-KR.

23.5.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	XFIA_CFG			Reserved	XFIB_CFG			Reserved	Reserved			Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved			Reserved	Reserved			Reserved	Reserved			Reserved	Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.16.4 Fields

Field	Function
31 —	Reserved
30-28 XFIA_CFG	XFIa Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - Disabled 001b - Enabled on Lane 0 to WRIOP MAC 1
27 —	Reserved
26-24 XFIB_CFG	XFIb Configuration SerDes #1 only All others reserved Default value set by RCW configuration. 000b - Disabled 001b - Enabled on Lane 3 connected to WRIOP MAC 2
23 —	Reserved
22-20 —	Reserved
19 —	Reserved
18-16 —	Reserved
15 —	Reserved
14-12 —	Reserved
11 —	Reserved
10-8 —	Reserved
7 —	Reserved
6-4 —	Reserved
3	Reserved

Table continues on the next page...

Field	Function
—	
2-0 —	Reserved

23.5.17 PCIe Equalization Configuration Register (PEXEQCR)

23.5.17.1 Offset

Register	Offset
PEXEQCR	300h

23.5.17.2 Function

PEXEQCR contains the PCIe gen3 Tx equalization configuration values.

23.5.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				FS						LF					
Reset	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0

23.5.17.4 Fields

Field	Function
31-12 —	Reserved

Table continues on the next page...

Field	Function
11-6 FS	PCI Express FS value. This field determines the FS value advertised during 8.0 GT/s link equalization phase 1, and is also used to calculate the preset C(0). Default value = 0x30 This value applies to all PCIe links on
5-0 LF	PCI Express LF value. This field determines the LF value advertised during 8.0 GT/s link equalization phase 1. Default value =0x10 This value applies to all PCIe links on

23.5.18 PCIe Equalization Preset 0 Register (PEXEQP0CR)

23.5.18.1 Offset

Register	Offset
PEXEQP0CR	304h

23.5.18.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.18.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000 000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.19 PCIe Equalization Preset 1 Register (PEXEQP1CR)

23.5.19.1 Offset

Register	Offset
PEXEQP1CR	308h

23.5.19.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.19.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000

Field	Function
	000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.20 PCIe Equalization Preset 2 Register (PEXEQP2CR)

23.5.20.1 Offset

Register	Offset
PEXEQP2CR	30Ch

23.5.20.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.20.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000 000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.21 PCIe Equalization Preset 3 Register (PEXEQP3CR)

23.5.21.1 Offset

Register	Offset
PEXEQP3CR	310h

23.5.21.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.21.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000

Field	Function
	000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.22 PCIe Equalization Preset 4 Register (PEXEQP4CR)

23.5.22.1 Offset

Register	Offset
PEXEQP4CR	314h

23.5.22.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.22.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000 000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.23 PCIe Equalization Preset 5 Register (PEXEQP5CR)

23.5.23.1 Offset

Register	Offset
PEXEQP5CR	318h

23.5.23.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

23.5.23.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000

Field	Function
	000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.24 PCIe Equalization Preset 6 Register (PEXEQP6CR)

23.5.24.1 Offset

Register	Offset
PEXEQP6CR	31Ch

23.5.24.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

23.5.24.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000 000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.25 PCIe Equalization Preset 7 Register (PEXEQP7CR)

23.5.25.1 Offset

Register	Offset
PEXEQP7CR	320h

23.5.25.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1

23.5.25.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000

Field	Function
	000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.26 PCIe Equalization Preset 8 Register (PEXEQP8CR)

23.5.26.1 Offset

Register	Offset
PEXEQP8CR	324h

23.5.26.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0

23.5.26.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000 000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.27 PCIe Equalization Preset 9 Register (PEXEQP9CR)

23.5.27.1 Offset

Register	Offset
PEXEQP9CR	328h

23.5.27.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

23.5.27.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000

Field	Function
	000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.28 PCIe Equalization Preset 10 Register (PEXEQP10CR)

23.5.28.1 Offset

Register	Offset
PEXEQP10CR	32Ch

23.5.28.2 Function

PEXEQPyCR (y=0-10) contains the PCIe gen3 Tx equalization preset values. Note that C(0) is automatically calculated from C(-1), C(+1), and PEXEQCR[FS].

23.5.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														CP1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CP1				Reserved						CM1					
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0

23.5.28.4 Fields

Field	Function
31-18 —	Reserved
17-12 CP1	C(+1) preset value. Default values for preset a= 000000b - 6'b00_1100 000001b - 6'b00_1000 000010b - 6'b00_1010 000011b - 6'b00_0110 000100b - 6'b00_0000 000101b - 6'b00_0000 000110b - 6'b00_0000 000111b - 6'b00_1001 001000b - 6'b00_0110 001001b - 6'b00_0000 001010b - 6'b00_1010
11-6 —	Reserved
5-0 CM1	C(-1) preset value. Default values for preset a= 000000b - 6'b00_0000 000001b - 6'b00_0000 000010b - 6'b00_0000 000011b - 6'b00_0000 000100b - 6'b00_0000 000101b - 6'b00_0101 000110b - 6'b00_0110 000111b - 6'b00_0101 001000b - 6'b00_0110 001001b - 6'b00_1000 001010b - 6'b00_0110

23.5.29 General Control Register 0 - Lane a (LNAGCR0 - LNDGCR0)

23.5.29.1 Offset

Register	Offset
LNAGCR0	800h
LNBGCR0	840h

Table continues on the next page...

Register	Offset
LNCGCR0	880h
LNDGCR0	8C0h

23.5.29.2 Function

LNmGCR0 contains functional control bits for SerDes lane *m*.

Special consideration must be taken when writing this register while PLLnRSTCTL[RST_DONE]=0, or if PCIe is running on this lane. See RPLL_LES, TPLL_LES, RRST_B and TRST_B fields for details.

See , for details on the sequence required to change settings.

23.5.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RPLL_LES	Reserved	RRAT_SEL		TPLL_LES	Reserved	TRAT_SEL		Reserved	RRST_B	TRST_B	RX_PD	TX_PD	IF20BIT_EN	Reserved	FIRST_LANE
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				PROTS					Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.29.4 Fields

Field	Function
31 RPLL_LES	Directs the RX portion of lane a to use the corresponding PLL. Default value set by reset configuration. Note: must be set the same as TPLL_LES for normal function. Note: this field may be updated by hardware if this lane is enabled for PCIe gen3. Use care to ensure that the PCIe controller is not attempting a speed switch to/from gen3 when writing this register, e.g. by disabling the controller. 0b - use PLL2

Table continues on the next page...

Field	Function
	1b - use PLL1
30 —	Reserved
29-28 RRAT_SEL	Receiver speed selection for lane a, relative to the corresponding PLLnCR0[FRATE_SEL], n as selected by LNaGCR0[RPLL_LES]: Default value set by reset configuration. Required setting per protocol: • SGMII 10 (1.25 Gbaud) • SGMII 00 (3.125 Gbaud) • QSGMII 00 • XFI/10GBase-KR 11 • PCIe 01 • SATA 10 This register field is read-only when running in PCIeor SATA mode. The above required values are defaults for gen1 initialization. Note: must be set the same as TRAT_SEL for normal function. 00b - Same as FRATE_SEL 01b - FRATE_SEL/2 10b - FRATE_SEL/4 11b - FRATE_SEL*2
27 TPLL_LES	Used to direct the TX portion of lane to use the corresponding PLL. Default value set by reset configuration. Note: must be set the same as RPLL_LES for normal function. See for details on the sequence required to change this setting. Note: this field may be updated by hardware if this lane is enabled for PCIe gen3. Use care to ensure that the PCIe controller is not attempting a speed switch to/from gen3 when writing this register, e.g. by disabling the controller. 0b - use PLL2 1b - use PLL1
26 —	Reserved
25-24 TRAT_SEL	Transmitter speed selection for lane a, relative to the corresponding PLLnCR0[FRATE_SEL], n as selected by LNaGCR0[TPLL_LES]: Default value set by reset configuration. Required setting per protocol: see RRAT_SEL This register field is read-only when running in PCIeor SATA mode. Note: must be set the same as RRAT_SEL for normal function. 00b - Same as FRATE_SEL 01b - FRATE_SEL/2 10b - FRATE_SEL/4 11b - FRATE_SEL*2
23 —	Reserved
22 RRST_B	Resets receiver for lane a Recommended setting per protocol: 1, unless PLLnRSTCTL[RST_DONE]=0, then 0.

Table continues on the next page...

Field	Function
	This register value is read-only during SerDes reset (POR or PLLnRSTCTL[RSTREQ]). This register field is read-only when running in PCIeor SATA mode. In SATA mode, this field may read as 1 while a controller-driven lane reset is in progress. Used in lane reset and reconfiguring procedures. See Lane Reset and Reconfiguration, for details. 0b - Reset 1b - Application Mode
21 TRST_B	Resets transmitter for lane a Recommended setting per protocol: 1 This register value is read-only during SerDes reset (POR or PLLnRSTCTL[RSTREQ]). This register field is read-only when running in PCIeor SATA mode. Used in lane reset and reconfiguring procedures. See Lane Reset and Reconfiguration, for details. 0b - Reset 1b - Application Mode
20 RX_PD	Lane powerdown for receiver on lane a Default value set by reset configuration. This register field read-only during SerDes reset (POR or PLLnRSTCTL[RSTREQ]). See Lane Enable After Powerdown for details on re-enabling powered down lanes. 0b - Lane receiver active 1b - Lane receiver powered down
19 TX_PD	Lane powerdown for transmitter on lane a Default value set by reset configuration. This register value is read-only during SerDes reset (POR or PLLnRSTCTL[RSTREQ]). Note: the user must not assert TX_PD for the master clock lane (TX_CLK_1STLANE=1) of a multi-lane link during normal function. See Lane Enable After Powerdown for details on e-enabling powered down lanes. 0b - Lane transmitter active 1b - Lane transmitter powered down
18 IF20BIT_EN	20-bit interface enable Default value set by reset configuration. Required value per protocol: PCIe 1 XFI 1 SGMII 0 QSGMII 1 SATA 1 0b - 10-bit interface 1b - 20-bit interface
17 —	Reserved
16 FIRST_LANE	Indicates this lane is the first (lane 0) of a group of lanes Default/recommended value set by reset configuration

Table continues on the next page...

Field	Function
	0b - Lane a is not lane 0 of the link 1b - Lane a is lane 0 of the link or unused lane
15-12 —	Reserved
11-7 PROTS	Lane Protocol Select. Default value is set by reset configuration All others reserved 00000b - PCIe 00001b - SGMII 1000Base-KX QSGMII OC-SGMII 00010b - SATA 01010b - XFI 10GBase-KR
6-0 —	Reserved

23.5.30 General Control Register 1 - Lane a (LNAGCR1 - LNDGCR1)

23.5.30.1 Offset

Register	Offset
LNAGCR1	804h
LNBGCR1	844h
LNCGCR1	884h
LNDGCR1	8C4h

23.5.30.2 Function

LNmGCR1 contains functional control bits for SerDes lane a.

23.5.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RDAT_INV	TDAT_INV	Reserved			OPAD_CTL	Reserved			REIDL_TH			REIDL_EX_SEL		REIDL_ET_SEL	
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REIDL_EX_MSB	REIDL_ET_MSB	REQ_CTL_SNP	REQ_CDR_SNP	Reserved				TRSTDIR	REQ_BIN_SNP	ISLEW_RCTL		Reserved		OSLEW_RCTL	
Reset	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1

23.5.30.4 Fields

Field	Function
31 RDAT_INV	Invert Rx data. Has the same effect as swapping SD_RX[m]_P and SD_RX[m]_N. Note: this field is read-only when this lane is operating as PCIe. 0b - Rx data is not inverted 1b - Rx data is inverted before it is decoded
30 TDAT_INV	Invert Tx data. Has the same effect as swapping SD_TX[m]_P and SD_TX[m]_N. 0b - Tx data is not inverted 1b - Tx data is inverted before it is transmitted
29-27 —	Reserved
26 OPAD_CTL	TX Output pad control signal for common mode Note: this field is read-only when this lane is operating as PCIe, SATA, XFI, SGMII or QSGMII. 0b - Transmitter Enabled 1b - Force Transmitter Output to Common Mode
25-23 —	Reserved
22-20 REIDL_TH	Receiver electrical idle detection threshold control Default value set by reset configuration Recommended value per protocol: PCIe: 100 (2.5 Gbaud) SGMII: 001 (1.25 Gbaud)

Table continues on the next page...

Field	Function
	SGMII: 000 (3.125 Gbaud) SATA: 100 (1.5 Gbaud) Others: 000 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[REIDL_TH_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[REIDL_TH_1).
19-18 REIDL_EX_SEL	Exit electrical idle filter select = {REIDL_EX_MSB,REIDL_EX_SEL} Default value set by reset configuration Recommended value per protocol: PCIe: 011 (2.5 Gbaud) SGMII: 011 (1.25 Gbaud) SGMII: 000 (3.125 Gbaud) 1GKX: 000 SATA: 001 (1.5 Gbaud) Others: 000 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[REIDL_EX_SEL_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[REIDL_EX_SEL_1).
17-16 REIDL_ET_SEL	Enter idle filter select = {REIDL_ET_MSB,REIDL_ET_SEL}: Recommended setting per protocol: PCIe: 111 (2.5 Gbaud) SGMII: 100 (1.25 Gbaud) SGMII: 000 (3.125 Gbaud) 1GKX: 000 SATA: 001 (1.5 Gbaud) Others: 000 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[REIDL_ET_SEL_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[REIDL_ET_SEL_1).
15 REIDL_EX_MSB	Exit idle filter select MSB. See REIDL_EX_SEL for settings. Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[REIDL_EX_MSB_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[REIDL_EX_MSB_1).
14 REIDL_ET_MSB	Enter idle filter select MSB. See REIDL_ET_SEL for settings. Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[REIDL_ET_MSB_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[REIDL_ET_MSB_1).
13 REQ_CTL_SNP	Initiate snapshot of RX Equalization Control Gaink2, Gaink3, and Offset Registers Recommended Setting per protocol: 0
12 REQ_CDR_SNP	nitiate snapshot of RX Clock/Data Recovery (CDR) Registers Recommended Setting per protocol: 0
11-8 —	Reserved

Table continues on the next page...

Field	Function
7 TRSTDIR	Multi-lane protocol Tx clock synchronization control Default value set by reset configuration. Recommended setting per protocol: PCIe: Others: 0
6 REQ_BIN_SNP	Initiate snapshot of RX Equalization Control Binning Registers Note: this field is read-only when this lane is operating as PCIe
5-4 ISLEW_RCTL	Slew control for Quadrature Generator Default value set by RCS configuration. Recommended setting per protocol: PCIe: 01 (2.5 Gbaud) SGMII: 01 (1.25 Gbaud) SGMII: 10 (3.125 Gbaud) QSGMII: 01 XFI: 01 SATA: 10 (1.5 Gbaud) Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[ISLEW_RCTL_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[ISLEW_RCTL_1).
3-2 —	Reserved
1-0 OSLEW_RCTL	Phase Interpolator Output clock edge rate control Default value set by reset configuration. Recommended setting per protocol: • PCIe: 01 (2.5 Gbaud) • SGMII: 01 (1.25 Gbaud) • SGMII: 10 (3.125 Gbaud) • QSGMII: 01 • XFI: 01 • SATA: 10 (1.5 Gbaud) Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[OSLEW_RCTL_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[OSLEW_RCTL_1).

23.5.31 Speed Switch Control Register 0 - Lane a (LNASSCR0 - LNDSSCR0)

23.5.31.1 Offset

Register	Offset
LNASSCR0	80Ch
LNBSSCR0	84Ch
LNCSSCR0	88Ch
LNDSSCR0	8CCh

23.5.31.2 Function

LNmSSCR0 contains control bits for modifying the PCI Express 5 Gbaud and SATA 3 Gbaud behavior of SerDes lane a. It is reserved for other protocols.

23.5.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REIDL_TH_0			REIDL_EX_SEL_0		REIDL_ET_SEL_0		REIDL_EX_MSB_0	REIDL_ET_MSB_0	ISLEW_RCTL_0		OSLEW_RCTL_0		RXEQ_BST_0	BASE_WAND_0	
Reset	1	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSETOVD6_0	TEQ_TYPE_0		SGN_PREQ_0	SGN_POST1Q_0	RATIO_PST1Q_0					AMP_RED_0					
Reset	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0

23.5.31.4 Fields

Field	Function
31-29 REIDL_TH_0	Receiver electrical idle detection threshold control Default settings set per reset configuration.

Table continues on the next page...

Field	Function
	Recommended settings per protocol: PCIe: 100 (5 Gbaud) SATA: 101 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_TH]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[REIDL_TH_1].
28-27 REIDL_EX_SEL_0	Exit electrical idle filter select = {REIDL_EX_MSB_0,REIDL_EX_SEL_0}: Default settings set per reset configuration. Recommended settings per protocol: PCIe: 011 (5 Gbaud) SATA: 001 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_EX_SEL]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[REIDL_EX_SEL_1].
26-25 REIDL_ET_SEL_0	Enter idle filter select = {REIDL_ET_MSB_0,REIDL_ET_SEL_0}: Default settings set per reset configuration. Recommended settings per protocol: PCIe: 000 (5 Gbaud) SATA: 001 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_ET_SEL]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[REIDL_ET_SEL_1].
24 REIDL_EX_MSB_0	Exit idle filter select MSB. See REIDL_EX_SEL_0 for settings. For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_EX_MSB]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[REIDL_EX_MSB_1].
23 REIDL_ET_MSB_0	Enter idle filter select MSB. See REIDL_ET_SEL_0 for settings. For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_ET_MSB]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[REIDL_ET_MSB_1].
22-21 ISLEW_RCTL_0	Slew control for Quadrature Generator Default settings set per reset configuration. Recommended settings per protocol: PCIe: 01 (5 Gbaud) SATA: 10 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[ISLEW_RCTRL]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[ISLEW_RCTRL_1].
20-19 OSLEW_RCTL_0	Phase Interpolator Output clock edge rate control Default settings set per reset configuration. Recommended settings per protocol:

Table continues on the next page...

Field	Function
	PCIe: 01 SATA: 10 For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[OSLEW_RCTL]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[OSLEW_RCTL_1].
18 RXEQ_BST_0	Rx Equalization Boost Default value set by reset configuration Recommended value per protocol: 0 For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaRECR0[RXEQ_BST]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[RXEQ_BST_1].
17-16 BASE_WAND_0	Baseline Wander Control Select Default settings set per reset configuration. Recommended settings per protocol: 00 For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaRECR0[BASE_WAND]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[BASE_WAND_1]. 00b - off (8b10b data) 01b - default BinBLW threshold 10b - alternate BinBLW sign 11b - Use RX Eq offset as GainBLW override
15 OSETOVD6_0	Binary Decode of Lane Adaptive Equalization offset intitialization or override value. [6] = 1: Double Imposed Offset Default value set by reset configuration Recommended setting per protocol: PCIe: 0 (5 Gbaud) SATA: 0 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaRECR0[OSETOVD]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[OSETOVD6_1].
14-13 TEQ_TYPE_0	Lane transmit equalization. Default value set by reset configuration Recommended setting per protocol: PCIe: 01 (5 Gbaud) SATA: 01 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[TEQ_TYPE]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[TEQ_TYPE_1]. 00b - No TX Equalization 01b - 2 Levels of TX Equalization (+1 post cursor) 10b - 3 Levels of TX Equalization (+1 pre-cursor and +1 post-cursor) 11b - Reserved

Table continues on the next page...

Field	Function
12 SGN_PREQ_0	Precursor sign Default value set by reset configuration Recommended setting per protocol: PCIe: 0 (5 Gbaud) SATA: 0 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[SGN_PREQ]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[SGN_PREQ_1]. 0b - Negative Sign (close eye) 1b - Positive Sign (open eye)
11 SGN_POST1Q_0	Post1q sign Default value set by reset configuration Recommended setting per protocol: PCIe: 1 (5 Gbaud) SATA: 1 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[SGN_POST1Q]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[SGN_POST1Q_1] 0b - Negative Sign (close eye) 1b - Positive Sign (open eye)
10-6 RATIO_PST1Q_0	Ratio of full swing transition bit to first post-cursor. Default value set by reset configuration Recommended setting per protocol: PCIe: 0_1100 (5 Gbaud) SATA: 0_0010 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[RATIO_PST1Q]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[RATIO_PST1Q_1]. Note: this field is read-only when this lane is operating as PCIe
5-0 AMP_RED_0	Overall TX Amplitude Reduction Default value set by reset configuration Recommended setting per protocol: PCIe: 00_0000 (5 Gbaud) SATA: 00_0111 (3 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[AMP_RED]. For PCIe 8 Gbaud and SATA 6 Gbaud, see LNaSSCR1[AMP_RED_1]. Note: this field is read-only when this lane is operating as PCIe

23.5.32 Receive Equalization Control Register 0 - Lane a (LNARECR0 - LNDRECR0)

23.5.32.1 Offset

Register	Offset
LNARECR0	810h
LNBRECR0	850h
LNCRECR0	890h
LNDRECR0	8D0h

23.5.32.2 Function

LNmRECR0 contains functional control bits for SerDes lane a

23.5.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			RXEQ_BST	GK2OVD				Reserved				GK3OVD			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GK2OVD_EN	GK3OVD_EN	OSETOVD_EN	Reserved	BASE_WAND		Reserved		Reserved	OSETOVD						
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

23.5.32.4 Fields

Field	Function
31-29 —	Reserved
28 RXEQ_BST	Rx Equalization Boost Default value set by reset configuration Recommended value per protocol: 0 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[RXEQ_BST_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[RXEQ_BST_1).
27-24 GK2OVD	Binary decode of lane Adaptive Equalization gaink2 initialization or override value. Default value set by reset configuration. Recommended settings per protocol: PCIe: 0000 SGMII: 1111 (1.25 Gbaud) SGMII: 0000 (3.125 Gbaud) QSGMII: 0000 XFI: 0000 SATA: 0000
23-20 —	Reserved
19-16 GK3OVD	Binary decode of lane Adaptive Equalization gaink3 intitialization or override value. Default value set by reset config. Recommended settings per protocol: PCIe: 0000 SGMII: 1111 (1.25 Gbaud) SGMII: 0000 (3.125 Gbaud) QSGMII: 0000 XFI: 0000 SATA: 0000
15 GK2OVD_EN	Controls source of rx equalization "gaink2" setting. Recommended settings per protocol: SGMII: 1 (1.25 Gbaud) SGMII: 0 (3.125 Gbaud) Others: 0 0b - Use rxeq adaption derived gaink2 1b - Fix gaink2 according to GK2OVD
14 GK3OVD_EN	Controls source of rx equalization "gaink3" setting. Recommended settings per protocol: SGMII: 1 (1.25 Gbaud)

Table continues on the next page...

Field	Function
	SGMII: 0 (3.125 Gbaud) Others: 0 0 Use rxeq adaption derived gaink3 1 Fix gaink3 according to GK3OVD
13 OSETOVD_EN	Controls source of rx equalization "offset" setting. Default value set by reset config Recommended setting per protocol: 0 0b - On release of srds_reset_b, initial rx eq offset is LNaRECR0[OSETOVD] and rx eq loop will begin adjustment at this value 1b - rx eq offset is fixed at LNaRECR0[OSETOVD]
12 —	Reserved
11-10 BASE_WAND	Baseline Wander Control Select Default value set by reset config Recommended setting per protocol: XFI: 01 Others: 00 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[BASE_WAND_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[BASE_WAND_1). 00b - off (8b10b data) 01b - default BinBLW threshold 10b - alternate BinBLW sign 11b - Use OSETOVD[4:0] as GainBLW override
9-8 —	Reserved
7 —	Reserved
6-0 OSETOVD	Binary Decode of Lane Adaptive Equalization offset intitialization or override value. [6] 1: Double Imposed Offset [5:0] b00 0000: + Maximum Imposed Offset [5:0] b01 1111: No imposed offset [5:0] b11 1111: - Maximum Imposed Offset Note: If BASE_WAND= 11, then use OSETOVD[4:0] as GainBLW Override Default value set by reset config Recommended setting per protocol: XFI: 101_1111 Others: 001_1111 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[OSETOVD_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[OSETOVD_1)

23.5.33 Receive Equalization Control Register 1- Lane a (LNARECR1 - LNDRECR1)

23.5.33.1 Offset

Register	Offset
LNARECR1	814h
LNBRECR1	854h
LNCRECR1	894h
LNDRECR1	8D4h

23.5.33.2 Function

LNmRECR1 contains control and status bits for receiver equalization on lane a.

23.5.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													EQ_BSNP_DN	EQ_CSNP_DN	CDR_SNP_DN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

23.5.33.4 Fields

Field	Function
31-3 —	Reserved
2 EQ_BSNP_DN	Snapshot of RX EQ Bin Complete
1 EQ_CSNP_DN	Snapshot of RX EQ Ctrl Complete
0 CDR_SNP_DN	Snapshot of CDR Loop Complete

23.5.34 Transmit Equalization Control Register 0 - Lane a (LNATECR0 - LNDTECR0)

23.5.34.1 Offset

Register	Offset
LNATECR0	818h
LNBTECR0	858h
LNCTECR0	898h
LNDTECR0	8D8h

23.5.34.2 Function

LNmTECR0 contains Tx equalization control bits for SerDes lane a.

23.5.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		TEQ_TYPE		Reserved	SGN_PREQ	RATIO_PREQ				SGN_POST1Q	RATIO_PST1Q				
Reset	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ADPT_EQ						Reserved		AMP_RED					
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

23.5.34.4 Fields

Field	Function
31-30 —	Reserved
29-28 TEQ_TYPE	Selects amount/type of Transmit Equalization Default value set by reset configuration. Recommended settings per protocol: SGMII: 00 (1.25 Gbaud) SGMII: 01 (3.125 Gbaud) XFI: 01 10G-KR: 10 Others: 01 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[TEQ_TYPE_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[TEQ_TYPE_1) 00b - No TX Equalization 01b - 2 Levels of TX Equalization (+1 post cursor) 10b - 3 Levels of TX Equalization (+1 pre-cursor and +1 post-cursor) 11b - Reserved
27 —	Reserved
26 SGN_PREQ	Precursor sign Default value set by reset configuration. Recommended settings per protocol:

Table continues on the next page...

Field	Function
	XFI: 0 10G-KR: 1 Others: 0 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[SGN_PREQ_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[SGN_PREQ_1) 0b - Negative Sign(close eye) 1b - Positive Sign (open eye)
25-22 RATIO_PREQ	Ratio of full swing transition bit to pre-cursor Default value set by reset configuration. Recommended settings per protocol: XFI: 0000 10G-KR: 0011 Others: 0000 Note: this field is read-only when this lane is operating as PCIe
21 SGN_POST1Q	Post q Sign Default value set by reset configuration. Recommended settings per protocol: SGMII: 0 (1.25 Gbaud) SGMII: 1 (3.125 Gbaud) Others: 1 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[SGN_POST1Q_0) Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[SGN_POST1Q_1). 0b - Negative Sign (close eye) 1b - Positive Sign (open eye)
20-16 RATIO_PST1Q	Ratio of full swing transition bit to first post-cursor. Default value set by reset configuration. Recommended settings per protocol: PCIe: 0_1000 (2.5 Gbaud) SGMII: 0_0000 (1.25 Gbaud) SGMII: 0_0110 (3.125 Gbaud) XFI: 0_0011 10G-KR: 0_1100 SATA: 0_0010 (1.5 Gbaud) Others: 0_0110 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[RATIO_PST1Q_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[RATIO_PST1Q_1). Note: this field is read-only when this lane is operating as PCIe
15-14 —	Reserved

Table continues on the next page...

Field	Function
13-8 ADPT_EQ	Transmitter Adjustments for 8G/10G Default value set by reset configuration. Recommended settings per protocol: 10G-KR: 10_0000 Others: 11_0000 Note: this field is read-only when this lane is operating as PCIe
7-6 —	Reserved
5-0 AMP_RED	Overall TX Amplitude Reduction Default value set by reset configuration. Recommended settings per protocol: SGMII: 00_0110 (1.25 Gbaud) SGMII: 00_0000 (3.125 Gbaud) 1000BASE-KX: 00_0000 QSGMII: 00_0010 XFI: 00_0111 10GKR: 00_0000 SATA: 01_0000 (1.5 Gbaud) Others: 00_0000 Note: SATA 3 Gbaud and PCIe 5 Gbaud setting taken from LNaSSCR0[AMP_RED_0). Note: SATA 6 Gbaud and PCIe 8 Gbaud setting taken from LNaSSCR1[AMP_RED_1) Note: this field is read-only when this lane is operating as PCIe

23.5.35 Speed Switch Control Register 1- Lane 0 (LNASSCR1 - LNDSSCR1)

23.5.35.1 Offset

Register	Offset
LNASSCR1	81Ch
LNBSSCR1	85Ch
LNCSSCR1	89Ch
LNDSSCR1	8DCh

23.5.35.2 Function

LNaSSCR1 contains control bits for modifying PCI Express 8 Gbaud and SATA 6 Gbaud behavior of SerDes lane a. It is reserved for other protocols.

23.5.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REIDL_TH_1			REIDL_EX_SEL_1		REIDL_ET_SEL_1		REIDL_EX_MSB_1	REIDL_ET_MSB_1	ISLEW_RCTL_1		OSLEW_RCTL_1		RXEQ_BST_1	BASE_WAND_1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSETOVD6_1	TEQ_TYPE_1		SGN_PREQ_1	SGN_POST1Q_1	RATIO_PST1Q_1					AMP_RED_1					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.35.4 Fields

Field	Function
31-29 REIDL_TH_1	Receiver electrical idle detection threshold control Default set by reset configuration. Recommended settings per protocol: PCIe: 100 (8 Gbaud) SATA: 000 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_TH]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[REIDL_TH_0].
28-27 REIDL_EX_SEL_1	Exit electrical idle filter select = REIDL_EX_MSB_1||REIDL_EX_SEL_1 Default set by reset configuration. Recommended settings per protocol:

Table continues on the next page...

Field	Function
	PCIe: 011 (8 Gbaud) SATA: 000 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_EX_SEL]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[REIDL_EX_SEL_0].
26-25 REIDL_ET_SEL_1	Enter idle filter select = REIDL_ET_MSB_1||REIDL_ET_SEL_1: Default set by reset configuration. Recommended setting per protocol: PCIe: 000 (8 Gbaud) SATA: 000 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_ET_SEL]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[REIDL_ET_SEL_0].
24 REIDL_EX_MSB_1	Exit idle filter select MSB. See REIDL_EX_SEL_1 for settings. Default set by reset configuration. For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_EX_MSB]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[REIDL_EX_MSB_0].
23 REIDL_ET_MSB_1	Enter idle filter select MSB. See REIDL_ET_SEL_1 for settings. For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[REIDL_ET_MSB]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[REIDL_ET_MSB_0].
22-21 ISLEW_RCTL_1	Slew control for Quadrature Generator Default value set by reset configuration Recommended Setting per protocol: PCIe: 10 (8 Gbaud) SATA: 10 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[ISLEW_RCTL]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[ISLEW_RCTL_0].
20-19 OSLEW_RCTL_1	Phase Interpolator Output clock edge rate control Default value set by reset configuration Recommended Setting per protocol: PCIe: 01 (8 Gbaud) SATA: 10 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaGCR1[OSLEW_RCTL]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[OSLEW_RCTL_0].
18 RXEQ_BST_1	Rx Equalization Boost Default value set by reset configuration Recommended value per protocol: 0

Table continues on the next page...

Field	Function
	For PCIe 2.5 Gbaud and SATA 1.5 Gbaud settings, see LNaRECR0[RXEQ_BST]. For PCIe 5 Gbaud and SATA 3 Gbaud settings, see LNaSSCR0[RXEQ_BST_0].
17-16 BASE_WAND_1	Baseline Wander Control Select Default value set by reset config Recommended setting per protocol: PCIe: 01 (8 Gbaud) SATA: 00 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud settings, see LNaRECR0[BASE_WAND]. For PCIe 5 Gbaud and SATA 3 Gbaud settings, see LNaSSCR0[BASE_WAND_0]. 00b - off (8b10b data) 01b - default BinBLW threshold 10b - alternate BinBLW sign 11b - Use OSETOVD[4:0] as GainBLW override
15 OSETOVD6_1	Binary Decode of Lane Adaptive Equalization offset intitialization or override value. [6] = 1: Double Imposed Offset Default value set by reset configuration Recommended setting per protocol: PCIe: 1 (8 Gbaud) SATA: 0 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaRECR0[OSETOVD]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[OSETOVD6_0].
14-13 TEQ_TYPE_1	Lane transmit equalization. Default value set by reset configuration Recommended setting per protocol: PCIe: 10 (8 Gbaud) SATA: 01 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[TEQ_TYPE]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[TEQ_TYPE_0]. 00b - No TX Equalization 01b - 2 Levels of TX Equalization (+1 post cursor) 10b - 3 Levels of TX Equalization (+1 pre-cursor and +1 post-cursor) 11b - Reserved
12 SGN_PREQ_1	Precursor sign Default value set by reset configuration Recommended setting per protocol: PCIe: 1 (8 Gbaud) SATA: 0 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[SGN_PREQ]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[SGN_PREQ_0].

Table continues on the next page...

Field	Function
	0b - Negative Sign (close eye) 1b - Positive Sign (open eye)
11 SGN_POST1Q_1	Post1q sign Default value set by reset configuration Recommended setting per protocol: PCIe: 1 (8 Gbaud) SATA: 1 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[SGN_POST1Q]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[SGN_POST1Q_0]. 0b - Negative Sign (close eye) 1b - Positive Sign (open eye)
10-6 RATIO_PST1Q_1	Ratio of full swing transition bit to first post-cursor. Default value set by reset configuration Recommended setting per protocol: PCIe: 0_1100 (8 Gbaud) SATA: 0_0010 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[AMP_RED]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[AMP_RED_0]. Note: this field is read-only when this lane is operating as PCIe
5-0 AMP_RED_1	Overall TX Amplitude Reduction Default value set by reset configuration Recommended setting per protocol: PCIe: 00_0000 (8 Gbaud) SATA: 00_0000 (6 Gbaud) For PCIe 2.5 Gbaud and SATA 1.5 Gbaud and protocols other than PCIe or SATA, see LNaTECR0[AMP_RED]. For PCIe 5 Gbaud and SATA 3 Gbaud, see LNaSSCR0[AMP_RED_0]. Note: this field is read-only when this lane is operating as PCIe

23.5.36 TTL Control Register 0 - Lane a (LNATTLCR0 - LNDTTLCR0)

23.5.36.1 Offset

Register	Offset
LNATTLCR0	820h
LNBTTLCR0	860h
LNCTTLCR0	8A0h
LNDTTLCR0	8E0h

23.5.36.2 Function

LNmTTLCR0 contains control bits for the Transition Tracking Loop (TTL) on SerDes lane a.

23.5.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		FLT_SEL						Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

23.5.36.4 Fields

Field	Function
31-30 —	Reserved
29-24 FLT_SEL	Selects the gain 'Kfr', 'Kph' and TTL Edge Counting Window Widths in the CDR Loop for the Lane. Default value set by reset configuration. Recommended values per protocol: SGMII: 11_1001 (1.25 Gbaud) SGMII: 00_0000 (3.125 Gbaud)

Table continues on the next page...

Field	Function
	Others: 00_0000
23-0 —	Reserved

23.5.37 Test Control/Status Register 3 - Lane a (LNATCSR3 - LNDTCSR3)

23.5.37.1 Offset

Register	Offset
LNATCSR3	83Ch
LNBTCSR3	87Ch
LNCTCSR3	8BCh
LNDTCSR3	8FCh

23.5.37.2 Function

LNmTCSR3 contains test control and status bits.

23.5.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		LPBK_EN		CDR_LCK	Reserved										
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.37.4 Fields

Field	Function
31-30 —	Reserved
29-28 LPBK_EN	Loopback data from TX to RX All others reserved Note: Loopback using PCI-Express requires Root Complex configuration. PCI Express End point loopback is not supported. 00b - Application Mode 01b - Loopback Mode
27 CDR_LCK	When asserted, CDR loop has acquired a valid Rx clock
26-0 —	Reserved

23.5.38 PEXa Protocol Control Register 0 (PEXACR0 - PEXCCR0)

23.5.38.1 Offset

Register	Offset
PEXACR0	1000h
PEXBCR0	1040h
PEXCCR0	1080h

23.5.38.2 Function

The PCIe Protocol Control and Status Registers support the control and status bits related to the PCIe PCS layers and PHY control.

PEXnCR0 contains control bits used for the PEXn protocol.

23.5.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				RD_SW	Reserved										
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.38.4 Fields

Field	Function
31-28 —	Reserved
27 RD_SW	Reduced Swing operation for 8 GT/s Note: full versus reduced swing for 2.5 and 5.0 GT/s is selected in the PCIe controller 0b - Full swing operation at 8 GT/s 1b - Reduced swing operation at 8 GT/s
26-0 —	Reserved

23.5.39 SGMIIa Protocol Control Register 1 (SGMIIACR1 - SGMIIDCR1)

23.5.39.1 Offset

Register	Offset
SGMIIACR1	1804h
SGMIIBCR1	1814h
SGMIICCR1	1824h
SGMIIDCR1	1834h

23.5.39.2 Function

SGMIInCR1 contains control bits used for the SGMIIn protocol.

23.5.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MDEV_PORT					Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SGPCS_EN	Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.39.4 Fields

Field	Function
31-27 MDEV_PORT	MDIO bus port address. • When accessing using Clause 22, this field is compared to MDIO_CTL[PORT_ADDR] to accept a command. • When accessing SGMII using clause 45, the MDIO Ethernet Management Interface register MDIO_CTL[DEV_ADDR], must be 03h and this field is compared to MDIO_CTL[PORT_ADDR] to accept a command. The register address (starting at 8000h) is specified in MDIO_ADDR. Default value: 0 Note: software must wait at least 3 platform clocks after changing this value before performing any MDIO accesses to the SGMIIa PCS.
26-12 —	Reserved
11 SGPCS_EN	SGMII PCS enable Recommended value: 1 if SGMIIa is enabled, else 0 Default value is set by reset Note: must be set to 1 as part of econfiguring a port from XFI /10GBase-KR to SGMII or 1000Base-KX operation
10-0 —	Reserved

23.5.40 SGMIIa Protocol Control Register 3 (SGMIIACR3 - SGMIIDCR3)

23.5.40.1 Offset

Register	Offset
SGMIIACR3	180Ch
SGMIIBCR3	181Ch
SGMIICCR3	182Ch
SGMIIDCR3	183Ch

23.5.40.2 Function

SGMIInCR3 contains control bits used for the SGMIIn protocol.

23.5.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX_QU	TX_ACT	Reserved		RX_QU	RX_ACT	Reserved		WK_ER R	Reserved						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.40.4 Fields

Field	Function
31	Tx quiet LPI state (read-only). Tx is in electrical idle.

Table continues on the next page...

Field	Function
TX_QU	
30 TX_ACT	Tx active LPI state (read-only)
29-28 —	reserved
27 RX_QU	Rx quiet LPI state (read-only). Rx is in electrical idle
26 RX_ACT	Rx active LPI state (read-only)
25-24 —	reserved
23 WK_ERR	Error on LPI wake (read-only)
22-0 —	Reserved

23.5.41 QSGMIIa Protocol Control Register 1 (QSGMIIACR1 - QSGMIIBCR1)

23.5.41.1 Offset

Register	Offset
QSGMIIACR1	1884h
QSGMIIBCR1	1894h

23.5.41.2 Function

QSGMIInCR1 contains control bits used for the QSGMIIn protocol.

23.5.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MDEV_PORT			Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.41.4 Fields

Field	Function
31-29 MDEV_PORT	Device port ID. Compared to upper 3 bits of MDIO_CTL[PORT_ADDR] to accept a command. The lower 2 bits of the PHY address map the SGMII port within the QSGMII PCS. Default value: 0 Note: software must wait at least 3 platform clocks after changing this value before performing any MDIO accesses to the QSGMIIa PCS.
28-0 —	Reserved

23.5.42 QSGMIIa Protocol Control Register 3 (QSGMIIACR3 - QSGMIIBCR3)

23.5.42.1 Offset

Register	Offset
QSGMIIACR3	188Ch
QSGMIIBCR3	189Ch

23.5.42.2 Function

QSGMIInCR3 contains control/status bits for the QSGMIIn protocol.

23.5.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												WK_ERR0	WK_ERR1	WK_ERR2	WK_ERR3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX_QU0	RX_QU1	RX_QU2	RX_QU3	RX_ACT0	RX_ACT1	RX_ACT2	RX_ACT3	TX_QU0	TX_QU1	TX_QU2	TX_QU3	TX_ACT0	TX_ACT1	TX_ACT2	TX_ACT3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.42.4 Fields

Field	Function
31-20 —	Reserved
19 WK_ERR0	Port 0 Error on LPI wake (read-only)
18 WK_ERR1	Port 1 Error on LPI wake (read-only)
17 WK_ERR2	Port 2 Error on LPI wake (read-only)
16 WK_ERR3	Port 3 Error on LPI wake (read-only)
15 RX_QU0	Port 0 Rx quiet LPI state (read-only)
14 RX_QU1	Port 1 Rx quiet LPI state (read-only)
13 RX_QU2	Port 2 Rx quiet LPI state (read-only)
12 RX_QU3	Port 3 Rx quiet LPI state (read-only)
11	Port 0 Rx active LPI state (read-only)

Table continues on the next page...

Field	Function
RX_ACT0	
10 RX_ACT1	Port 1 Rx active LPI state (read-only)
9 RX_ACT2	Port 2 Rx active LPI state (read-only)
8 RX_ACT3	Port 3 Rx active LPI state (read-only)
7 TX_QU0	Port 0 Tx quiet LPI state (read-only)
6 TX_QU1	Port 1 Tx quiet LPI state (read-only)
5 TX_QU2	Port 2 Tx quiet LPI state (read-only)
4 TX_QU3	Port 3 Tx quiet LPI state (read-only)
3 TX_ACT0	Port 0 Tx active LPI state (read-only)
2 TX_ACT1	Port 1 Tx active LPI state (read-only)
1 TX_ACT2	Port 2 Tx active LPI state (read-only)
0 TX_ACT3	Port 3 Tx active LPI state (read-only)

23.5.43 XFIa Protocol Control Register 1 (XFIACR1 - XFIBCR1)

23.5.43.1 Offset

Register	Offset
XFIACR1	1984h
XFIBCR1	1994h

23.5.43.2 Function

XFInCR1 contains control bits used for the XFIn protocol

23.5.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MDEV_PORT					Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.43.4 Fields

Field	Function
31-27 MDEV_PORT	MDIO bus port address. Compared to the MDIO_CTL[PORT_ADDR] to accept a command. Default value: 0 Note: software must wait at least 3 platform clocks after changing this value before performing any MDIO accesses to the SGMIIn PCS.
26-0 —	Reserved

23.5.44 XFIa Protocol Control Register 3 (XFIACR3 - XFIBCR3)

23.5.44.1 Offset

Register	Offset
XFIACR3	198Ch
XFIBCR3	199Ch

23.5.44.2 Function

XFInCR3 contains control bits used for the XFIn rotocol.

23.5.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TX_QU	TX_ACT	Reserved		RX_QU	RX_ACT	Reserved		WK_ERR	TX_ALRT	Reserved					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.5.44.4 Fields

Field	Function
31 TX_QU	Tx quiet LPI state. Read-only
30 TX_ACT	Tx active LPI state. Read-only
29-28 —	Reserved
27 RX_QU	Rx quiet LPI state. Read-only
26 RX_ACT	Rx active LPI state. Read-only
25-24 —	Reserved
23 WK_ERR	LPI wake error. Read-only
22 TX_ALRT	LPI Tx alert. Read-only
21-0 —	Reserved

23.6 MDIO register spaces

The SerDes module also contains MDIO slave ports containing registers for each PCS module.

The PCS MDIO registers are accessed through an indirect method from the MDIO Ethernet management interface registers in the Ethernet MAC controllers. There are two methods used to access the PCS MDIO registers—dubbed Clause 45 and Clause 22 (from the IEEE 802.3 sub clauses that define them), depending on the protocol.

The Clause 45 protocols are subdivided into separate spaces selected by the device address. The following table shows the Clause 45 register spaces:

Table 23-12. Clause 45 register spaces

XFI MDIO registers			
Register space	**Device address**	**Notes**	**See**
XFI PMA/PMD registers	01h	Also used for 10GBase-KR	MDIO_XFI_PMD register descriptions
XFI PCS registers	03h	Also used for 10GBase-KR	MDIO_XFI_PCS register descriptions
XFI Auto-Negotiation registers	07h	Also used for 10GBase-KR	MDIO_XFI_AN register descriptions
XFI Vendor-Specific 1 registers	1Eh	Also used for 10GBase-KR	MDIO_XFI_VENDOR_SPEC register descriptions

Table 23-12. Clause 45 register spaces

1000Base-KX MDIO registers			
Register space	**Device address**	**Notes**	**See**
1000Base-KX PCS registers	03h	Includes vendor-specific alias of Clause 22 SGMII registers starting at register address 8000h	MDIO_KX_PCS register descriptions
1000Base-KX Auto-Negotiation registers	07h		MDIO_KX_AN register descriptions
1000Base-KX Vendor-Specific 1 registers	1Dh		MDIO_KX_VENDOR_SPEC register descriptions

The following table shows the Clause 22 register spaces:

Table 23-13. Clause 22 registers

Registers	Notes	See
SGMII		MDIO_SGMII register descriptions

Table continues on the next page...

Table 23-13. Clause 22 registers (continued)

Registers	Notes	See
QSGMII	For QSGMII, the most-significant 3 bits of MDIO_CTL[PHY_ADDR] are compared to QSGMIInCR1[MDEV_PORT]. The least-significant 2 bits are used to select the specific QSGMII PCS port (0, 1, 2, or 3).	MDIO_QSGMII register descriptions

NOTE

Throughout the MDIO register sections:

- Each MDIO register is 16 bits.
- Register offsets are 16-bit offsets, not byte offsets.
- Addresses not listed are reserved.
- Read accesses to reserved addresses return 0x0000.

23.6.1 MDIO_XFI_PMD register descriptions

The XFI PMA/PMD register space is selected when the associated XFInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 01h.

23.6.1.1 MDIO_XFI_PMD memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	XFI PMD Control 1 (XFI_PMD_CR1)	16	RW	0000h
96h	XFI 10GBASE-KR PMD Control Register (XFI_10GKR_PMD_CR)	16	RW	0000h
97h	XFI 10GBASE-KR PMD Status Register (XFI_10GKR_PMD_SR)	16	RW	0000h
98h	XFI 10GBASE-KR LP Coefficient Update Register (XFI_10GKR_LP_COEF_UPDATE)	16	RW	0000h
99h	XFI 10GBASE-KR LP Status Report Register (XFI_10GKR_LP_SRR)	16	RO	0000h
9Ah	XFI 10GBASE-KR LD Coefficient Update Register (XFI_10GKR_LD_COEF_UPDATE)	16	RO	0000h
9Bh	XFI 10GBASE-KR LD Status Report Register (XFI_10GKR_LD_SRR)	16	RW	0000h
AAh	XFI 10GBASE-R FEC Ability (XFI_10GR_FEC_ABIL)	16	RO	0001h
ABh	XFI 10GBASE-R FEC Control (XFI_10GR_FEC_CTL)	16	RW	0000h
ACh	XFI 10GBASE-R FEC Corrected Blocks Counter Lower (XFI_10GR_FEC_COR_BLK_CNT_L)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
ADh	XFI 10GBASE-R FEC Corrected Blocks Counter Upper (XFI_10GR_FEC_COR_BLK_CNT_U)	16	ROZ	0000h
AEh	XFI 10GBASE-R FEC Uncorrected Blocks Counter Lower (XFI_10GR_FEC_UNCOR_BLK_CNT_L)	16	RO	0000h
AFh	XFI 10GBASE-R FEC Uncorrected Blocks Counter Upper (XFI_10GR_FEC_UNCOR_BLK_CNT_U)	16	ROZ	0000h
8000h	XFI 10GBASE-R Vendor-specific PMA Status (XFI_VEND_SPEC_PMA_STAT)	16	RO	0000h
8001h	XFI Number of PRBS Sequence Errors Lower (XFI_10GKR_NPRBS_ERR_CNT_LO)	16	RO	0000h
8002h	XFI Number of PRBS Sequence Errors Upper (XFI_10GKR_NPRBS_ERR_CNT_HI)	16	RO	0000h

23.6.1.2 XFI PMD Control 1 (XFI_PMD_CR1)

23.6.1.2.1 Offset

Register	Offset
XFI_PMD_CR1	0h

23.6.1.2.2 Function

XFI PMD Control 1 register contains global controls for the XFI PMD module.

23.6.1.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															PMD_RESET
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.2.4 Fields

Field	Function
15-1 —	Reserved
0 PMD_RESET	PMD Reset PMD Reset Self-Clearing Reset Bit. When set to 1, resets the 10GBase-KR Link Training modules. The reset command clears all bits in the 10GBase-KR PMD control register

23.6.1.3 XFI 10GBASE-KR PMD Control Register (XFI_10GKR_PMD_CR)

23.6.1.3.1 Offset

Register	Offset
XFI_10GKR_PMD_CR	96h

23.6.1.3.2 Function

XFI 10GBASE-KR PMD Control Register contains global controls for the 10GBASE-KR.

23.6.1.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														TRAIN_EN	RESTART_TRAIN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.3.4 Fields

Field	Function
15-2	Reserved

Table continues on the next page...

Field	Function
—	
1 TRAIN_EN	Training Enable Training Enable 0b - 10GBase-KR startup protocol disabled 1b - 10GBase-KR startup protocol enabled (Note: To safely disable link training, the RESTART_TRAIN bit should be written with 1 with TRAIN_ENABLE = 0 to force exit of the state machine from whatever state it is in. (see IEEE 802.3 Fig 72-5 Training State Diagram).)
0 RESTART_TRAIN	Restart Training Restart Training Self Clearing Bit. When set to 1, resets the 10GBase-KR startup protocol and begins or ends its operation (depending on TRAIN_EN).

23.6.1.4 XFI 10GBASE-KR PMD Status Register (XFI_10GKR_PMD_SR)

23.6.1.4.1 Offset

Register	Offset
XFI_10GKR_PMD_SR	97h

23.6.1.4.2 Function

XFI 10GBASE-KR PMD Control Register contains status for the 10GBASE-KR

23.6.1.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TRAIN_FAIL	SUP_STAT	FRAME_LOCK	RX_STAT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.4.4 Fields

Field	Function
15-4 —	Reserved
3 TRAIN_FAIL	Training Failure Training Failure: 0b - Training failure has not been detected 1b - Training failure has been detected (Note: when training failure occurs, the link is kept in an inoperable state (PCS is not allowed to lock). The application then must write the 10GBase-KR PMD Control register with the restart bit being set and the enable bit being cleared to exit this condition and allow the link to come up without link training.)
2 SUP_STAT	Startup Protocol Status Start-up Protocol Status 0b - Startup protocol is completed (or not started) 1b - Startup protocol is in progress
1 FRAME_LOCK	Frame Lock Frame Lock 0b - Training frame delineation not detected 1b - Training frame delineation detected
0 RX_STAT	Rx Status Receiver Status 0b - Receiver training is proceeding (or not started) 1b - Receiver is trained and is ready to receive data (Note: this bit is writeable, which differs from the corresponding notion in IEEE 802.3 Clause 45.2.1.77 for MMD register 1.151) as the application has to decide when the local device is trained. Writing this bit sets the mr_trained variable allowing the training statemachine to proceed (see IEEE 802.3 Clause 72.6.10.3.1))

23.6.1.5 XFI 10GBASE-KR LP Coefficient Update Register (XFI_10GKR_LP_COEF_UPDATE)

23.6.1.5.1 Offset

Register	Offset
XFI_10GKR_LP_COEF_UPDATE	98h

23.6.1.5.2 Function

XFI 10GBASE-KR LP Coefficient Update Register contains controls for the 10GBASE-KR Link Partner Coefficient Update. The fields in this register are read-only if XFI 10GBASE-KR PMD Control Register[TRAIN_EN]=1, but they are writeable if TRAIN_EN=0.

23.6.1.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		PRESET	INIT	Reserved						COP1_UPD		COZ_UPD		COM1_UPD	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.5.4 Fields

Field	Function
15-14 —	Reserved
13 PRESET	Preset the transmitter coefficients Preset the transmitter coefficients 0b - Normal operation 1b - Preset the transmitter coefficients
12 INIT	Initialize the transmitter coefficients Initialize the transmitter coefficients 0b - Normal operation 1b - Initialize the transmitter coefficients
11-6 —	Reserved
5-4 COP1_UPD	Coefficient (+1) Update Coefficient (+1) update: 00b - hold 01b - increment 10b - decrement 11b - reserved
3-2 COZ_UPD	Coefficient (0) Update Coefficient (0) update: 00b - hold 01b - increment 10b - decrement

Table continues on the next page...

Field	Function
	11b - reserved
1-0 COM1_UPD	Coefficient (-1) Update Coefficient (-1) update: 00b - hold 01b - increment 10b - decrement 11b - reserved

23.6.1.6 XFI 10GBASE-KR LP Status Report Register (XFI_10GKR_LP_SRR)

23.6.1.6.1 Offset

Register	Offset
XFI_10GKR_LP_SRR	99h

23.6.1.6.2 Function

XFI 10GBASE-KR LP Coefficient Status Report Register contains the 10GBASE-KR Link Partner Coefficient reported status.

23.6.1.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RX_RDY	Reserved									COP1_STAT		COZ_STAT		COM1_STAT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.6.4 Fields

Field	Function
15 RX_RDY	Receiver Ready Receiver Ready

Table continues on the next page...

Field	Function
	0b - The LP receiver is requesting that training continue 1b - The LP receiver has determined that training is complete and is prepared to receive data.
14-6 —	Reserved
5-4 COP1_STAT	Coefficient (+1) status Coefficient (+1) status: 00b - not updated 01b - updated 10b - minimum 11b - maximum
3-2 COZ_STAT	Coefficient (0) status Coefficient (0) status: 00b - not updated 01b - updated 10b - minimum 11b - maximum
1-0 COM1_STAT	Coefficient (-1) status Coefficient (-1) status: 00b - not updated 01b - updated 10b - minimum 11b - maximum

23.6.1.7 XFI 10GBASE-KR LD Coefficient Update Register (XFI_10GKR_LD_COEF_UPDATE)

23.6.1.7.1 Offset

Register	Offset
XFI_10GKR_LD_COEF_UPDATE	9Ah

23.6.1.7.2 Function

XFI 10GBASE-KR LD Coefficient Update Register contains controls for the 10GBASE-KR Link Device Coefficient Update. The value written to this register is reported to the Link Partner during training.

23.6.1.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		PRESET	INIT	Reserved						COP1_UPD		COZ_UPD		COM1_UPD	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.7.4 Fields

Field	Function
15-14 —	Reserved
13 PRESET	Preset the transmitter coefficients Preset the transmitter coefficients 0b - Normal operation 1b - Preset the transmitter coefficients
12 INIT	Initialize the transmitter coefficients Initialize the transmitter coefficients 0b - Normal operation 1b - Initialize the transmitter coefficients
11-6 —	Reserved
5-4 COP1_UPD	Coefficient (+1) update Coefficient (+1) update: 00b - hold 01b - increment 10b - decrement 11b - reserved
3-2 COZ_UPD	Coefficient (0) update Coefficient (0) update: 00b - hold 01b - increment 10b - decrement 11b - reserved
1-0 COM1_UPD	Coefficient (-1) update Coefficient (-1) update: 00b - hold 01b - increment 10b - decrement 11b - reserved

23.6.1.8 XFI 10GBASE-KR LD Status Report Register (XFI_10GKR_LD_SRR)

23.6.1.8.1 Offset

Register	Offset
XFI_10GKR_LD_SRR	9Bh

23.6.1.8.2 Function

XFI 10GBASE-KR LD Coefficient Status Report Register contains the 10GBASE-KR Link Device Coefficient reported status

23.6.1.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX_RDY	Reserved									COP1_STAT		COZ_STAT		COM1_STAT	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.8.4 Fields

Field	Function
15 RX_RDY	Receiver Ready Receiver Ready 0b - The LD receiver is requesting that training continue (1: The LD receiver has determined that training is complete and is prepared to receive data.)
14-6 —	Reserved
5-4 COP1_STAT	Coefficient (+1) status Coefficient (+1) status: 00b - not updated 01b - updated 10b - minimum 11b - maximum

Table continues on the next page...

Field	Function
3-2 COZ_STAT	Coefficient (0) status Coefficient (0) status: 00b - not updated 01b - updated 10b - minimum 11b - maximum
1-0 COM1_STAT	Coefficient (-1) status Coefficient (-1) status: 00b - not updated 01b - updated 10b - minimum 11b - maximum

23.6.1.9 XFI 10GBASE-R FEC Ability (XFI_10GR_FEC_ABIL)

23.6.1.9.1 Offset

Register	Offset
XFI_10GR_FEC_ABIL	AAh

23.6.1.9.2 Function

XFI 10GBASE-R FEC Ability Register contains the status bits showing the PCS FEC capability.

23.6.1.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														FEC_ERR_ABIL	FEC_ABIL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.1.9.4 Fields

Field	Function
15-2 —	Reserved
1 FEC_ERR_ABIL	10GBASE-R FEC Error Indication Ability 10GBASE-R FEC Error Indication Ability Set to 0 to indicate that the FEC is not able to report FEC decoding errors to the PCS layer
0 FEC_ABIL	10GBase-R FEC Ability 10GBase-R FEC Ability Set to 1 to indicate this PCS implements the FEC functions

23.6.1.10 XFI 10GBASE-R FEC Control (XFI_10GR_FEC_CTL)

23.6.1.10.1 Offset

Register	Offset
XFI_10GR_FEC_CTL	ABh

23.6.1.10.2 Function

XFI 10GBASE-R FEC Control Register contains the control bits for the PCS FEC capability.

23.6.1.10.3 Diagram

23.6.1.10.4 Fields

Field	Function
15-2 —	Reserved
1 FEC_ENAB_ERR	10GBASE-R FEC Enable Error Indication 10GBASE-R FEC Error Indication Ability Set to 0 to indicate that the FEC is not able to report FEC decoding errors to the PCS layer
0 FEC_ENAB	10GBase-R FEC Enable 10GBase-R FEC Enable 0: FEC is not enabled 1: FEC is enabled

23.6.1.11 XFI 10GBASE-R FEC Corrected Blocks Counter Lower (XFI_10GR_FEC_COR_BLK_CNT_L)

23.6.1.11.1 Offset

Register	Offset
XFI_10GR_FEC_COR_BLK_CNT_L	ACh

23.6.1.11.2 Function

XFI 10GBASE-R FEC Corrected Blocks Counter Lower contains bits 15:0 of the FEC corrected blocks counter.

23.6.1.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COR_BL_CNT_L															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.11.4 Fields

Field	Function
15-0 COR_BL_CNT_L	10GBase-R FEC Corrected Bk=locks Counter Lower Number of corrected FEC blocks, lower 16 bits Reset to 0 when a read is performed on the upper bits. The counter does not roll over to 0 when the value 0xFFFF_FFFF is reached

23.6.1.12 XFI 10GBASE-R FEC Corrected Blocks Counter Upper (XFI_10GR_FEC_COR_BLK_CNT_U)

23.6.1.12.1 Offset

Register	Offset
XFI_10GR_FEC_COR_BLK_CNT_U	ADh

23.6.1.12.2 Function

XFI 10GBASE-R FEC Corrected Blocks Counter Upper contains bits 31:16 of the FEC corrected blocks counter.

23.6.1.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.12.4 Fields

Field	Function
15-0 COR_BL_CNT_U	10GBase-R FEC Corrected Bk=locks Counter Upper Number of corrected FEC blocks, upper 16 bits Reset to 0 when a read is performed. The counter does not roll over to 0 when the value 0xFFFF_FFFF is reached

23.6.1.13 XFI 10GBASE-R FEC Uncorrected Blocks Counter Lower (XFI_10GR_FEC_UNCOR_BLK_CNT_L)

23.6.1.13.1 Offset

Register	Offset
XFI_10GR_FEC_UNCOR_BLK_CNT_L	AEh

23.6.1.13.2 Function

XFI 10GBASE-R FEC Uncorrected Blocks Counter Lower contains bits 15:0 of the FEC uncorrected blocks counter.

23.6.1.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNCOR_BL_CNT_L															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.13.4 Fields

Field	Function
15-0 UNCOR_BL_CNT_L	10GBase-R FEC Uncorrected Bk=locks Counter Lower Number of uncorrected FEC blocks, lower 16 bits Reset to 0 when a read is performed on the upper bits. The counter does not roll over to 0 when the value 0xFFFF_FFFF is reached

23.6.1.14 XFI 10GBASE-R FEC Uncorrected Blocks Counter Upper (XFI_10GR_FEC_UNCOR_BLK_CNT_U)

23.6.1.14.1 Offset

Register	Offset
XFI_10GR_FEC_UNCOR_BLK_CNT_U	AFh

23.6.1.14.2 Function

XFI 10GBASE-R FEC Uncorrected Blocks Counter Upper contains bits 31:16 of the FEC uncorrected blocks counter.

23.6.1.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.14.4 Fields

Field	Function
15-0 UNCOR_BL_CNT_U	10GBase-R FEC Uncorrected Bk=locks Counter Upper Number of uncorrected FEC blocks, upper 16 bits Reset to 0 when a read is performed. The counter does not roll over to 0 when the value 0xFFFF_FFFF is reached

23.6.1.15 XFI 10GBASE-R Vendor-specific PMA Status (XFI_VEND_SPEC_PMA_STAT)

23.6.1.15.1 Offset

Register	Offset
XFI_VEND_SPEC_PMA_STAT	8000h

23.6.1.15.2 Function

XFI 10GBASE-R Vendor-specific PMA Status.

23.6.1.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															FEC_LOCKED
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.15.4 Fields

Field	Function
15-1 —	Reserved
0 FEC_LOCKED	10GBase-R FEC Locked FEC Receiver Locked 0: FEC receiver is not locked 2: FEC receiver is locked

23.6.1.16 XFI Number of PRBS Sequence Errors Lower (XFI_10GKR_NPRBS_ERR_CNT_LO)

23.6.1.16.1 Offset

Register	Offset
XFI_10GKR_NPRBS_ERR_CNT_LO	8001h

23.6.1.16.2 Function

The XFI Number of PRBS Sequence Errors Lower Register contains the lower half of the 32-bit counter for tracking number of link training frame PRBS Sequence Errors

23.6.1.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NPRBS_ERR_CNT_L															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.16.4 Fields

Field	Function
15-0 NPRBS_ERR_CNT_L	Number of Link Training Frame PRBS Sequence Error, lower 16 Bits Number of Link Training Frame PRBS Sequence Error, lower 16 Bits. Reset to 0 when a read is performed on the counter upper register, the counter does not roll over to 0 when the value 0xFFFFFFFF is reached.

23.6.1.17 XFI Number of PRBS Sequence Errors Upper (XFI_10GKR_NPRBS_ERR_CNT_HI)

23.6.1.17.1 Offset

Register	Offset
XFI_10GKR_NPRBS_ERR_CNT_HI	8002h

23.6.1.17.2 Function

The XFI Number of PRBS Sequence Errors Upper Register contains the upper half of the 32-bit counter for tracking number of link training frame PRBS Sequence Errors

23.6.1.17.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NPRBS_ERR_CNT_H															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.1.17.4 Fields

Field	Function
15-0 NPRBS_ERR_CNT_H	Number of Link Training Frame PRBS Sequence Error, upper 16 Bits Number of Link Training Frame PRBS Sequence Error, upper 16 Bits. Reset to 0 when a read is performed on the counter upper register, the counter does not roll over to 0 when the value 0xFFFFFFFF is reached.

23.6.2 MDIO_XFI_PCS register descriptions

The XFI PCS register space is selected when the associated XFInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 03h.

23.6.2.1 MDIO_XFI_PCS memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	XFI PCS Control 1 (XFI_PCS_CR1)	16	RW	2000h
1h	XFI PCS Status 1 (XFI_PCS_SR1)	16	RO	0002h
2h	XFI PCS Device Identifier Upper (XFI_PCS_DEV_ID_H)	16	RO	0083h
3h	XFI PCS Device Identifier Lower (XFI_PCS_DEV_ID_L)	16	RO	E400h
4h	XFI PCS Speed Ability (XFI_PCS_SPEED_ABIL)	16	RO	0001h
5h	XFI PCS Devices In Package 0 (XFI_PCS_DEV_PRES0)	16	RO	008Ah
6h	XFI PCS Devices in Package 1 (XFI_PCS_DEV_PRES1)	16	RO	0000h
7h	XFI 10G PCS Control 2 (XFI_PCS_CR2)	16	RO	000Bh
8h	XFI 10G PCS Status 2 (XFI_PCS_SR2)	16	RO	8001h
Eh	XFI PCS Package Identifier Upper (XFI_PCS_PKG_ID_H)	16	RO	0083h
Fh	XFI PCS Package Identifier Lower (XFI_PCS_PKG_ID_L)	16	RO	E400h
20h	XFI 10GBASE-R PCS Status 1 (XFI_PCS_10GR_SR1)	16	RO	0000h
21h	XFI 10GBASE-R PCS Status 2 (XFI_PCS_10GR_SR2)	16	RO	0000h
22h	XFI 10GBASE-R PCS Test Pattern Seed A 0 (XFI_PCS_TP_SEED_A0)	16	RW	0000h
23h	XFI 10GBASE-R PCS Test Pattern Seed A 1 (XFI_PCS_TP_SEED_A1)	16	RW	0000h
24h	XFI 10GBASE-R PCS Test Pattern Seed A 2 (XFI_PCS_TP_SEED_A2)	16	RW	0000h
25h	XFI 10GBASE-R PCS Test Pattern Seed A 3 (XFI_PCS_TP_SEED_A3)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
26h	XFI 10GBASE-R PCS Test Pattern Seed B 0 (XFI_PCS_TP_SEED_B0)	16	RW	0000h
27h	XFI 10GBASE-R PCS Test Pattern Seed B 1 (XFI_PCS_TP_SEED_B1)	16	RW	0000h
28h	XFI 10GBASE-R PCS Test Pattern Seed B 2 (XFI_PCS_TP_SEED_B2)	16	RW	0000h
29h	XFI 10GBASE-R PCS Test Pattern Seed B 3 (XFI_PCS_TP_SEED_B3)	16	RW	0000h
2Ah	XFI 10GBASE-R PCS Test Pattern Control (XFI_PCS_TP_CR)	16	RW	0000h
2Bh	XFI 10GBASE-R PCS Test Pattern Error Counter (XFI_PCS_TP_ERR_CNT)	16	RO	0000h
8000h	Vendor Specific PCS Status (XFI_PCS_VENDOR_SR)	16	RO	0000h

23.6.2.2 XFI PCS Control 1 (XFI_PCS_CR1)

23.6.2.2.1 Offset

Register	Offset
XFI_PCS_CR1	0h

23.6.2.2.2 Function

The PCS Control 1 Register contains controls for the XFI PCS

23.6.2.2.3 Diagram

23.6.2.2.4 Fields

Field	Function
15 RESET	Reset Reset Self Clearing bit. 0b - PCS is not in reset 1b - PCS transmit and receive functions are being reset
14 LPBK	Loopback Loopback Enable 0b - Normal function 1b - XGMII Tx data is returned to XGMII Rx
13 SPEED_SEL0	Speed Selection Speed Selection Read only 13 6
12 —	Reserved
11 LPWR	Low Power Operation Low Power Operation 0b - Normal function 1b - Low power operation: PCS is in reset state and most functions are disabled
10-7 —	Reserved
6 SPEED_SEL1	Speed Selection 1 See SPEED_SEL1 Read only
5-2 SPEED_SEL	Speed Selection Speed Selection Read only 0000b - 10 Gbps
1-0 —	Reserved

23.6.2.3 XFI PCS Status 1 (XFI_PCS_SR1)

23.6.2.3.1 Offset

Register	Offset
XFI_PCS_SR1	1h

23.6.2.3.2 Function

The XFI PCS Status 1 Register contains status bits for the XFI PCS

23.6.2.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				TX_LPI	RX_LPI	TX_LPI_ACTIVE	RX_LPI_ACTIVE	FAULT	LPI_CLK_STP_CAP	Reserved			PCS_RX_LNK_STAT	LPWR_ABIL	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

23.6.2.3.4 Fields

Field	Function
15-12 —	Reserved
11 TX_LPI	Tx LPI Once set, stays set until register read Note: for this device Tx LPI state does not Transmit LPI 0b - normal operation. 1b - transmit is or was in LPI (EEE)
10 RX_LPI	Rx LPI Receive LPI Once set, stays set until register read 0b - normal operation. 1b - receive is or was in LPI state (EEE)
9	TX LPI Active Transmit LPI Active

Table continues on the next page...

Field	Function
TX_LPI_ACTIVE	0b - normal operation 1b - transmit is currently in LPI state (EEE)
8 RX_LPI_ACTIVE	RX LPI Active Receive LPI Active 0b - normal operation 1b - receive is currently in LPI state (EEE).
7 FAULT	Fault Fault 0b - no fault condition detected 1b - fault condition detected
6 LPI_CLK_STP_CAP	LPI Clock Stop Capability Clock Stop Capable: 0b - MAC is not capable of stopping the clock during LPI
5-3 —	Reserved
2 PCS_RX_LNK_STAT	PCS Rx Link Status PCS Receive Link Status Once cleared, stays cleared until register read 0b - indicates that the PCS receive link is or was down 1b - indicates that the PCS receive link is up.
1 LPWR_ABIL	Low Power Ability Low Power Ability 1b - The PCS implements a low power mode, meaning bit setting 11 of PCS Control register 1 register is supported and allows to place the PCS is a reset state.
0 —	Reserved

23.6.2.4 XFI PCS Device Identifier Upper (XFI_PCS_DEV_ID_H)

23.6.2.4.1 Offset

Register	Offset
XFI_PCS_DEV_ID_H	2h

23.6.2.4.2 Function

The XFI PCS Device Identifier Upper Register contains the upper half of the PCS Device Identifier

23.6.2.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCS_DEV_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.2.4.4 Fields

Field	Function
15-0 PCS_DEV_ID_U	PCS Device ID Upper PCS Device ID Upper: 15:0 - OIU[3:18]

23.6.2.5 XFI PCS Device Identifier Lower (XFI_PCS_DEV_ID_L)

23.6.2.5.1 Offset

Register	Offset
XFI_PCS_DEV_ID_L	3h

23.6.2.5.2 Function

The XFI PCS Device Identifier Lower Register contains the lower half half of the PCS Device Identifier

23.6.2.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCS_DEV_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.2.5.4 Fields

Field	Function
15-0 PCS_DEV_ID_L	PCS Device ID Lower PCS Device ID Lower: 15:10 - OIU[19:24] 9:4 - Manufacturer's Model Number 3:0 - Revision Number

23.6.2.6 XFI PCS Speed Ability (XFI_PCS_SPEED_ABIL)

23.6.2.6.1 Offset

Register	Offset
XFI_PCS_SPEED_ABIL	4h

23.6.2.6.2 Function

The XFI PCS Speed Ability Register contains ability bits for the 10GBASE-R PCS

23.6.2.6.3 Diagram

23.6.2.6.4 Fields

Field	Function
15-2 —	Reserved
1 CAP_10P_TS_2B_TL_CAP	10Pass-TS/2Base-TL Capable 10PASS-TS/2BASE-TL Capable 0b - The PCS is not capable of operating as the 10P/2B PCS
0 CAP_10G	1-G Capable 10G Capable 1b - PCS is capable of operating at 10Gbps.

23.6.2.7 XFI PCS Devices In Package 0 (XFI_PCS_DEV_PRES0)

23.6.2.7.1 Offset

Register	Offset
XFI_PCS_DEV_PRES0	5h

23.6.2.7.2 Function

The XFI PCS Devices in Package 0 Register contains half of the devices present status

23.6.2.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AN_PRES	TC_PRES	DTE_XS_PRES	PHY_XS_PRES	PCS_PRES	WIS_PRES	PMD_PMA_PRES	C22_REG_PRES
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0

23.6.2.7.4 Fields

Field	Function
15-8 —	Reserved
7 AN_PRES	AN Present Auto-negotiation Present 1b - the PCS implements the Auto-Negotiation function
6 TC_PRES	TC Present TC Present 0b - the PCS does not implement TC functions
5 DTE_XS_PRES	DTE XS Present DTE XS Present 0b - the PCS does not implement DTE XS functions
4 PHY_XS_PRES	PHY XS Present PHY XS Present 0b - the PCS does not implement PHY XS functions.
3 PCS_PRES	PSC Present PCS Present 1b - the PCS implements PCS functions
2 WIS_PRES	WIS Present WIS Present 0b - the PCS does not implement a WIS function.
1 PMD_PMA_PRES	PMD/PMA Present PMD/PMA Present 1b - the PCS implements PMD/PMA functions (Link Training and FEC functions)
0 C22_REG_PRES	Clause 22 Registers Present Clause 22 Registers Present 0b - the PCS does not implement the Clause 22 Registers.

23.6.2.8 XFI PCS Devices in Package 1 (XFI_PCS_DEV_PRES1)

23.6.2.8.1 Offset

Register	Offset
XFI_PCS_DEV_PRES1	6h

23.6.2.8.2 Function

The XFI PCS Devices in Package 1 Register contains half of the devices present status

23.6.2.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VEND_SPEC_DEV2_PRES	VEND_SPEC_DEV1_PRES	CLAUSE22_EXT_PRES	Reserved												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.8.4 Fields

Field	Function
15 VEND_SPEC_DEV2_PRES	Vendor Specific Device 2 Present Vendor Specific Device 2 Present 0b - the PCS does not implement any vendor specific device.
14 VEND_SPEC_DEV1_PRES	Vendor Specific Device 1 Present Vendor Specific Device 1 Present 0b - the PCS does not implement any vendor specific device.
13 CLAUSE22_EXT_PRES	Clause 22 Extension Present Clause 22 Extension Present 0b - the PCS does not implement Clause 22 extensions.
12-0 —	Reserved

23.6.2.9 XFI 10G PCS Control 2 (XFI_PCS_CR2)

23.6.2.9.1 Offset

Register	Offset
XFI_PCS_CR2	7h

23.6.2.9.2 Function

The XFI 10G PCS Control 2 Register contains control bits for the 10G PCS

23.6.2.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													PCS_TYP_SEL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

23.6.2.9.4 Fields

Field	Function
15-3 —	Reserved
2-0 PCS_TYP_SEL	PSC Type Selection PCS Type Selection Read-only field, as this PCS only supports 10GBase-R functions. Note that the value of this field does not match the 802.3 encoding for 10GBase-R.

23.6.2.10 XFI 10G PCS Status 2 (XFI_PCS_SR2)

23.6.2.10.1 Offset

Register	Offset
XFI_PCS_SR2	8h

23.6.2.10.2 Function

The XFI 10G PCS Status 2 Register contains status bits for the 10G PCS

23.6.2.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEV_PRES		Reserved		TX_FAULT	RX_FAULT	Reserved						CAP_10GBASET	CAP_10GBASEW	CAP_10GBASEX	CAP_10GBASER
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.2.10.4 Fields

Field	Function
15-14 DEV_PRES	Device Present Device Present 10b - device responding at this address
13-12 —	Reserved
11 TX_FAULT	Tx Fault Transmit Fault 0b - no fault condition on transmit path 1b - fault condition on transmit path
10 RX_FAULT	Rx Fault Receive Fault 0b - no fault condition on receive path 1b - fault condition on receive path
9-4 —	Reserved
3 CAP_10GBASET	10GBase-T Capability 10GBase-T Capable 0b - PCS is not able to support 10GBase-T PCS type
2 CAP_10GBASEW	10GBase-W Capability 10GBase-W Capable 0b - PCS is not able to support 10GBase-W PCS type
1 CAP_10GBASEX	10GBase-X capability 10GBase-X Capable 0b - PCS is not able to support 10GBase-X PCS type
0 CAP_10GBASER	10GBase-R Capability 10GBase-R Capable 1b - PCS is able to support 10GBase-R PCS type

23.6.2.11 XFI PCS Package Identifier Upper (XFI_PCS_PKG_ID_H)

23.6.2.11.1 Offset

Register	Offset
XFI_PCS_PKG_ID_H	Eh

23.6.2.11.2 Function

The XFI PCS Package Identifier Upper Register contains the upper half of a 32-bit unique identifier for a particular type of package that the PCS is instantiated within

23.6.2.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCS_PKG_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.2.11.4 Fields

Field	Function
15-0 PCS_PKG_ID_U	PCS Package ID Upper PCS Package ID Upper: 15:0 - OIU[3:18]

23.6.2.12 XFI PCS Package Identifier Lower (XFI_PCS_PKG_ID_L)

23.6.2.12.1 Offset

Register	Offset
XFI_PCS_PKG_ID_L	Fh

23.6.2.12.2 Function

The XFI PCS Package Identifier Lower Register contains the lower half of a 32-bit unique identifier for a particular type of package that the PCS is instantiated within.

23.6.2.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCS_PKG_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.2.12.4 Fields

Field	Function
15-0 PCS_PKG_ID_L	PSC Package ID Lower PCS Package ID Lower: 15:10 - OIU[19:24] 9:4 - Manufacturer's Model Number 3:0 - Revision Number

23.6.2.13 XFI 10GBASE-R PCS Status 1 (XFI_PCS_10GR_SR1)

23.6.2.13.1 Offset

Register	Offset
XFI_PCS_10GR_SR1	20h

23.6.2.13.2 Function

The XFI 10GBASE-R PCS Status 1 Register contains 10GBase-R PCS status.

23.6.2.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			RX_LNK_STAT	Reserved								PRBS9_ABIL	PRBS31_ABIL	PCS_HI_BER	PCS_BLK_LK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.13.4 Fields

Field	Function
15-13 —	Reserved
12 RX_LNK_STAT	Rx Link Status 10GBase-R Receive Link Status 0b - 10GBase-R PCS receive link is down 1b - 10GBase-R PCS receive link is up
11-4 —	Reserved
3 PRBS9_ABIL	PRBS9 Pattern Testing Ability PRBS9 Pattern Testing Ability 0b - the PCS does not support PRBS9 pattern testing
2 PRBS31_ABIL	PRBS31 Pattern testing Ability PRBS31 Pattern Testing Ability 0b - the PCS does not support PRBS31 pattern testing
1 PCS_HI_BER	PSC High BER 10GBase-R PCS high BER 0b - PCS not reporting a high BER 1b - PCS reporting a high BER
0 PCS_BLK_LK	PCS Block Lock 10GBase-R PCS block lock 0b - PCS not locked to receive blocks 1b - PCS locked to receive blocks

23.6.2.14 XFI 10GBASE-R PCS Status 2 (XFI_PCS_10GR_SR2)

23.6.2.14.1 Offset

Register	Offset
XFI_PCS_10GR_SR2	21h

23.6.2.14.2 Function

XFI 10GBASE-R PCS Status 2 Register contains PCS status bits.

23.6.2.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	L_BLK_LK	LH_BER	BER						ERR_BLKS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.14.4 Fields

Field	Function
15 L_BLK_LK	Latched Block Lock Latched Block Lock When cleared, stays cleared until register read0: PCS does not have block lock 1b - PCS has block lock
14 LH_BER	Latched High BER Latched high BER When set, stays set until register read 0b - PCS has not reported a high BER 1b - PCS has reported a high BER
13-8 BER	BER BER Counter Does not roll over from 0x3F to 0x00 Cleared on register read or PCS reset
7-0 ERR_BLKS	Errored Blocks Counter Errored Blocks Counter Does not roll over from 0xFF to 0x00 Cleared on register read or PCS reset

23.6.2.15 XFI 10GBASE-R PCS Test Pattern Seed A 0 (XFI_PCS_TP_SEED_A0)

23.6.2.15.1 Offset

Register	Offset
XFI_PCS_TP_SEED_A0	22h

23.6.2.15.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed A Register 0 contains bits 15:0 of Test Pattern Seed A.

23.6.2.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEST_PAT_SEED_A															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.15.4 Fields

Field	Function
15-0 TEST_PAT_SEED_A	Test Pattern Seed A Test Pattern Seed A [15:0]

23.6.2.16 XFI 10GBASE-R PCS Test Pattern Seed A 1 (XFI_PCS_TP_SEED_A1)

23.6.2.16.1 Offset

Register	Offset
XFI_PCS_TP_SEED_A1	23h

23.6.2.16.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed A Register 1 contains bits 31:16 of Test Pattern Seed A.

23.6.2.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEST_PAT_SEED_A															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.16.4 Fields

Field	Function
15-0 TEST_PAT_SEED_A	Test Pattern Seed A Test Pattern Seed A [31:16]

23.6.2.17 XFI 10GBASE-R PCS Test Pattern Seed A 2 (XFI_PCS_TP_SEED_A2)

23.6.2.17.1 Offset

Register	Offset
XFI_PCS_TP_SEED_A2	24h

23.6.2.17.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed A Register 2 contains bits 47:32 of Test Pattern Seed A.

23.6.2.17.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEST_PAT_SEED_A															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.17.4 Fields

Field	Function
15-0 TEST_PAT_SEED_A	Test Pattern Seed A Test Pattern Seed A [47:32]

23.6.2.18 XFI 10GBASE-R PCS Test Pattern Seed A 3 (XFI_PCS_TP_SEED_A3)

23.6.2.18.1 Offset

Register	Offset
XFI_PCS_TP_SEED_A3	25h

23.6.2.18.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed A Register 3 contains bits 57:48 of Test Pattern Seed A.

23.6.2.18.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TEST_PAT_SEED_A									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.18.4 Fields

Field	Function
15-10 —	Reserved
9-0 TEST_PAT_SEED_A	Test Pattern Seed A Test Pattern Seed A [57:48]

23.6.2.19 XFI 10GBASE-R PCS Test Pattern Seed B 0 (XFI_PCS_TP_SEED_B0)

23.6.2.19.1 Offset

Register	Offset
XFI_PCS_TP_SEED_B0	26h

23.6.2.19.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed B Register 0 contains bits 15:0 of Test Pattern Seed B.

23.6.2.19.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEST_PAT_SEED_B															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.19.4 Fields

Field	Function
15-0 TEST_PAT_SEED_B	Test Pattern Seed B Test Pattern Seed B [15:0]

23.6.2.20 XFI 10GBASE-R PCS Test Pattern Seed B 1 (XFI_PCS_TP_SEED_B1)

23.6.2.20.1 Offset

Register	Offset
XFI_PCS_TP_SEED_B1	27h

23.6.2.20.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed B Register 1 contains bits 31:16 of Test Pattern Seed B.

23.6.2.20.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEST_PAT_SEED_B															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.20.4 Fields

Field	Function
15-0 TEST_PAT_SEED_B	Test Pattern Seed B Test Pattern Seed B [31:16]

23.6.2.21 XFI 10GBASE-R PCS Test Pattern Seed B 2 (XFI_PCS_TP_SEED_B2)

23.6.2.21.1 Offset

Register	Offset
XFI_PCS_TP_SEED_B2	28h

23.6.2.21.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed B Register 2 contains bits 47:32 of Test Pattern Seed B.

23.6.2.21.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TEST_PAT_SEED_B															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.21.4 Fields

Field	Function
15-0 TEST_PAT_SEED_B	Test Pattern Seed B Test Pattern Seed B [47:32]

23.6.2.22 XFI 10GBASE-R PCS Test Pattern Seed B 3 (XFI_PCS_TP_SEED_B3)

23.6.2.22.1 Offset

Register	Offset
XFI_PCS_TP_SEED_B3	29h

23.6.2.22.2 Function

The XFI 10GBASE-R PCS Test Pattern Seed B Register 3 contains bits 57:48 of Test Pattern Seed B.

23.6.2.22.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TEST_PAT_SEED_B									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.22.4 Fields

Field	Function
15-10 —	Reserved
9-0 TEST_PAT_SEED_B	Test Pattern Seed B Test Pattern Seed B [57:48]

23.6.2.23 XFI 10GBASE-R PCS Test Pattern Control (XFI_PCS_TP_CR)

23.6.2.23.1 Offset

Register	Offset
XFI_PCS_TP_CR	2Ah

23.6.2.23.2 Function

The XFI 10GBASE-R PCS Test Pattern Control Register contains control bits for test pattern function.

23.6.2.23.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									PRBS9_TX_TEST_PAT_EN	PRBS31_RX_TEST_PAT_EN	PRBS31_TX_TEST_PAT_EN	TX_TEST_PAT_EN	RX_TEST_PAT_EN	TEST_PAT_SEL	DATA_PAT_SEL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.23.4 Fields

Field	Function
15-7 —	Reserved
6 PRBS9_TX_TEST_PAT_EN	PRBS9 Tx Test Pattern Enable PRBS9 transmit test pattern enable Read-only always set to 0 0b - Disable PRBS9 test-pattern mode on the transmit path
5 PRBS31_RX_TEST_PAT_EN	PRBS31 Rx Test Pattern Enable PRBS31 receive test pattern enable Read-only always set to 0 0b - Disable PRBS31 test-pattern mode on the receive path
4 PRBS31_TX_TEST_PAT_EN	PRBS32 Tx Test Pattern Enable PRBS31 transmit test pattern enable Read-only always set to 0 0b - Disable PRBS31 test-pattern mode on the transmit path
3 TX_TEST_PAT_EN	Tx Test Pattern Enable Transmit test pattern enable 0b - Disable transmit test pattern 1b - Enable transmit test pattern
2 RX_TEST_PAT_EN	Rx Test Pattern Enable Receive test pattern enable 0b - Disable receive test pattern 1b - Enable receive test pattern
1 TEST_PAT_SEL	Test Pattern Select Test pattern select

Table continues on the next page...

Field	Function
	0b - Pseudo-random test pattern 1b - Square wave test pattern
0 DATA_PAT_SEL	Data Pattern Select Transmit test pattern enable 0b - LF data pattern 1b - Zeroes pattern

23.6.2.24 XFI 10GBASE-R PCS Test Pattern Error Counter (XFI_PCS_TP_ERR_CNT)

23.6.2.24.1 Offset

Register	Offset
XFI_PCS_TP_ERR_CNT	2Bh

23.6.2.24.2 Function

The XFI 10GBASE-R PCS Test Pattern Error Counter Register contains the error counter for the test pattern.

23.6.2.24.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TEST_PAT_ERR_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.24.4 Fields

Field	Function
15-0 TEST_PAT_ERR_CNT	Test Pattern Error Count Test Pattern Error Counter

23.6.2.25 Vendor Specific PCS Status (XFI_PCS_VENDOR_SR)

23.6.2.25.1 Offset

Register	Offset
XFI_PCS_VENDOR_SR	8000h

23.6.2.25.2 Function

The XFI Vendor Specific PCS Status Register contains the status for the XFI PCS.

23.6.2.25.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														TX_FIFO_FAULT	RX_FIFO_FAULT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.2.25.4 Fields

Field	Function
15-2 —	Reserved
1 TX_FIFO_FAULT	Tx FIFO Fault Transmit FIFO Fault Once set, stays set until register read 0b - No error on the transmit decoupling FIFO 1b - Error (overflow or underflow) on the transmit decoupling FIFO
0 RX_FIFO_FAULT	Rx FIFO Fault Receive FIFO Fault Once set, stays set until register read 0b - No error on the receive decoupling FIFO 1b - Error (overflow or underflow) on the receive decoupling FIFO

23.6.3 MDIO_XFI_AN register descriptions

The XFI auto-negotiation register space is selected when the associated XFInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 07h.

23.6.3.1 MDIO_XFI_AN memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	XFI AN Control (XF_AN_CR)	16	RW	0000h
1h	XFI AN Status (XF_AN_SR)	16	RO	0008h
2h	XFI AN Device Identifier Upper (XFI_AN_DEV_ID_H)	16	RO	0083h
3h	XFI AN Device Identifier Lower (XFI_DEV_ID_L)	16	RO	E400h
5h	XFI AN Devices In Package 0 (XFI_AN_DEV_PRES0)	16	RO	008Ah
6h	XFI AN Devices In Package 1 (XFI_AN_DEV_PRES1)	16	RO	0000h
Eh	XFI AN Package Identifier Upper (XF_AN_PKG_ID_H)	16	RO	0083h
Fh	XFI AN Package Identifier Lower (XFI_AN_PKG_ID_L)	16	RO	E400h
10h	XFI AN Advertisement 0 (XFI_AN_ADVERT0)	16	RW	0001h
11h	XFI AN Advertisement 1 (XFI_AN_ADVERT1)	16	RW	001Fh
12h	XFI AN Advertisement 2 (XFI_AN_ADVERT2)	16	RW	C000h
13h	XFI AN LP Base Page Ability 0 (XFI_AN_LP_BASE_PG_ABIL0)	16	RO	0000h
14h	XFI AN LP Base Page Ability 1 (XFI_AN_LP_BASE_PG_ABIL1)	16	RO	0000h
15h	XFI AN LP Base Page Ability 2 (XFI_AN_LP_BASE_PG_ABIL2)	16	RO	0000h
16h	XFI AN Extended Next Page Transmit 0 (XFI_AN_XNP_TX0)	16	RW	2001h
17h	XFI AN Extended Next Page Transmit 1 (XFI_AN_XNP_TX1)	16	RW	0000h
18h	XFI AN Extended Next Page Transmit 2 (XFI_AN_XNP_TX2)	16	RW	0000h
19h	XFI AN LP Extended Next Page Ability 0 (XFI_AN_LP_XNP_ABIL0)	16	RO	0000h
1Ah	XFI AN LP Extended Next Page Ability 1 (XFI_AN_LP_XNP_ABIL1)	16	RO	0000h
1Bh	XFI AN LP Extended Next Page Ability 2 (XFI_AN_LP_XNP_ABIL2)	16	RO	0000h
30h	XFI Backplane Ethernet Status (XFI_BP_STAT)	16	RO	0000h

23.6.3.2 XFI AN Control (XF_AN_CR)

23.6.3.2.1 Offset

Register	Offset
XF_AN_CR	0h

23.6.3.2.2 Function

The AN Control Register contains general controls for auto-negotiation.

23.6.3.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AN_RST	Reserved	EXT_NP_CTRL	AN_EN	Reserved		RESTART_AN	Reserved								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.2.4 Fields

Field	Function
15 AN_RST	AN Reset AN reset Self-clearing bit 0b - AN normal operation 1b - AN reset
14 —	Reserved
13 EXT_NP_CTRL	Extended Next Page Control Extended Next Page Control 0b - Extended next pages are disabled 1b - Extended next pages are enabled
12 AN_EN	AN Enable Auto-negotiation enable 0b - Disable auto-negotiation process 1b - Enable auto-negotiation process
11-10 —	Reserved
9	Restart AN

Table continues on the next page...

Field	Function
RESTART_AN	Restart auto-negotiation 0b - Auto-negotiation in process, disabled, or not supported 1b - Restart auto-negotiation process
8-0 —	Reserved

23.6.3.3 XFI AN Status (XF_AN_SR)

23.6.3.3.1 Offset

Register	Offset
XF_AN_SR	1h

23.6.3.3.2 Function

The XFI AN Status Register contains status bits for the XFI auto-negotiate function.

23.6.3.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						PAR_DET_FAULT	Reserved	EXT_NP_STAT	PAGE_RCV	AN_COMP	REM_FAULT	AN_ABIL	LNK_STAT	Reserved	LP_AN_ABIL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

23.6.3.3.4 Fields

Field	Function
15-10 —	Reserved
9	Parallel Detection Fault Parallel detection fault

Table continues on the next page...

Field	Function
PAR_DET_FAULT	Once set, stays set until register read 0b - A fault has not been detected via the parallel detection function 1b - A fault has been detected via the parallel detection function
8 —	Reserved
7 EXT_NP_STAT	Extended Next Page Status Extended next page status 0b - Extended next page is not allowed 1b - Extended next page format is used
6 PAGE_RCV	Page Received Page received Once set, stays set until register read 0b - A page has not been received 1b - A page has been received
5 AN_COMP	AN Complete Auto-Negotiation complete 0b - Auto-negotiation process not completed 1b - Auto-negotiation process completed
4 REM_FAULT	remote fault Remote Fault Once set, stays set until register read. 0b - No remote fault condition detected 1b - Remote fault condition detected
3 AN_ABIL	AN ability Auto-Negotiation Ability 1b - PHY is able to perform Auto-Negotiation
2 LNK_STAT	Link Status Link Status Once set, stays set until register read. 0b - Link is down 1b - Link is up
1 —	Reserved
0 LP_AN_ABIL	Link Partner AN ability Link partner Auto-Negotiation ability 0b - LP is not able to perform Auto-Negotiation 1b - LP is able to perform Auto-Negotiation

23.6.3.4 XFI AN Device Identifier Upper (XFI_AN_DEV_ID_H)

23.6.3.4.1 Offset

Register	Offset
XFI_AN_DEV_ID_H	2h

23.6.3.4.2 Function

The AN Device Identifier Upper Register contains the upper half of the 32-bit device identifier.

23.6.3.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AN_DEV_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.3.4.4 Fields

Field	Function
15-0 AN_DEV_ID_U	AN Device ID Upper AN Device ID Upper: 15:0 - OIU[3:18]

23.6.3.5 XFI AN Device Identifier Lower (XFI_DEV_ID_L)

23.6.3.5.1 Offset

Register	Offset
XFI_DEV_ID_L	3h

23.6.3.5.2 Function

The AN Device Identifier Lower Register contains the lower half of the 32-bit device identifier.

23.6.3.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AN_DEV_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.3.5.4 Fields

Field	Function
15-0 AN_DEV_ID_L	AN Device ID Lower AN Device ID Lower: 15:10 - OIU[19:24] 9:4 - Manufacturer's Model Number 3:0 - Revision Number

23.6.3.6 XFI AN Devices In Package 0 (XFI_AN_DEV_PRES0)

23.6.3.6.1 Offset

Register	Offset
XFI_AN_DEV_PRES0	5h

23.6.3.6.2 Function

The XFI AN Devices in Package 0 Register contains half of the AN devices in package status.

23.6.3.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AN_PRES	TC_PRES	DTE_XS_PRES	PHY_XS_PRES	PCS_PRES	WIS_PRES	PMD_PMA_PRES	C22_REG_PRES
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0

23.6.3.6.4 Fields

Field	Function
15-8 —	Reserved
7 AN_PRES	AN Present Auto-negotiation Present 1b - the PCS implements the Auto-Negotiation function
6 TC_PRES	TC Present TC Present 0b - the PCS does not implement TC functions
5 DTE_XS_PRES	DTE XS Present DTE XS Present 0b - the PCS does not implement DTE XS functions
4 PHY_XS_PRES	PHY XS Present PHY XS Present 0b - the PCS does not implement PHY XS functions.
3 PCS_PRES	PCS Present PCS Present 1b - the PCS implements PCS functions
2 WIS_PRES	WIS Present WIS Present 0b - the PCS does not implement a WIS function.
1 PMD_PMA_PRES	PMD/PMA Present PMD/PMA Present 1b - the PCS implements PMD/PMA functions (Link Training and FEC functions)
0 C22_REG_PRES	Clause 22 Registers Present Clause 22 Registers Present 0b - the PCS does not implement the Clause 22 Registers.

23.6.3.7 XFI AN Devices In Package 1 (XFI_AN_DEV_PRES1)

23.6.3.7.1 Offset

Register	Offset
XFI_AN_DEV_PRES1	6h

23.6.3.7.2 Function

The XFI AN Devices in Package 1 Register contains half of the AN devices in package status.

23.6.3.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VEND_SPEC_DEV2_PRES	VEND_SPEC_DEV1_PRES	CLAUSE22_EXT_PRES	Reserved												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.7.4 Fields

Field	Function
15 VEND_SPEC_DEV2_PRES	Vendor Specific Device 2 Present Vendor Specific Device 2 Present 0b - the PCS does not implement any vendor specific device.
14 VEND_SPEC_DEV1_PRES	Vendor Specific Device 1 Present Vendor Specific Device 1 Present 0b - the PCS does not implement any vendor specific device.
13	Clause 22 Extension Present

Table continues on the next page...

Field	Function
CLAUSE22_EXT_PRES	Clause 22 Extension Present 0b - the PCS does not implement Clause 22 extensions.
12-0 —	Reserved

23.6.3.8 XFI AN Package Identifier Upper (XF_AN_PKG_ID_H)

23.6.3.8.1 Offset

Register	Offset
XF_AN_PKG_ID_H	Eh

23.6.3.8.2 Function

The AN Package Identifier Upper Register contains the upper half of the 32-bit package identifier.

23.6.3.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AN_PKG_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.3.8.4 Fields

Field	Function
15-0 AN_PKG_ID_U	AN Package ID Upper AN Package ID Upper: 15:0 - OIU[3:18]

23.6.3.9 XFI AN Package Identifier Lower (XFI_AN_PKG_ID_L)

23.6.3.9.1 Offset

Register	Offset
XFI_AN_PKG_ID_L	Fh

23.6.3.9.2 Function

The AN Package Identifier Lower Register contains the lower half of the 32-bit package identifier.

23.6.3.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AN_PKG_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.3.9.4 Fields

Field	Function
15-0 AN_PKG_ID_L	AN Package ID Lower AN Package ID Lower: 15:10 - OIU[19:24] 9:4 - Manufacturer's Model Number 3:0 - Revision Number

23.6.3.10 XFI AN Advertisement 0 (XFI_AN_ADVERT0)

23.6.3.10.1 Offset

Register	Offset
XFI_AN_ADVERT0	10h

23.6.3.10.2 Function

The XFI AN Advertisement Register 0 contains bits 15:0 of the 48-bit page bits used during base page exchange. They should be programmed before auto-negotiation is started. The 48-bit value is distributed over registers 0, 1, 2, where register 2 holds the most significant 16 bits of value.

23.6.3.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NP	ACK	REM_FAULT	XNP_CAP	PAUSE_CAP		ECHOED_NONCE					SELECT_FLD				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.3.10.4 Fields

Field	Function
15 NP	Next Page Next Page: 0b - Device does not have a Next Page to send 1b - Device has a Next Page to send
14 ACK	Ack Should always be set to 0
13 REM_FAULT	Remote Fault Remote Fault 0b - Normal operation 1b - Device is indicating a remote fault condition
12 XNP_CAP	Extended Next Page capability Extended Next Page Capability 0b - Device does not support extended next pages
11-10 PAUSE_CAP	Pause Capability Pause Capability (ASM_DIR:PAUSE) 00b - No PAUSE 01b - Symmetric PAUSE 10b - Asymmetric PAUSE toward link partner 11b - Both Symmetric PAUSE and Asymmetric PAUSE toward local device
9-5	Echoed Nonce Echoed Nonce

Table continues on the next page...

Field	Function
ECHOED_NONCE	Nonce value received from link partner
4-0 SELECT_FLD	Selector Field Selector Field 00001b - IEEE Standard 802.3

23.6.3.11 XFI AN Advertisement 1 (XFI_AN_ADVERT1)

23.6.3.11.1 Offset

Register	Offset
XFI_AN_ADVERT1	11h

23.6.3.11.2 Function

The XFI AN Advertisement Register 1 contains bits 31:16 of the 48-bit page bits used during base page exchange. They should be programmed before auto-negotiation is started. The 48-bit value is distributed over registers 0, 1, 2, where register 2 holds the most significant 16 bits of value.

23.6.3.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TECH_A10_A0											TRANS_NONCE				
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

23.6.3.11.4 Fields

Field	Function
15-5 TECH_A10_A0	Technology A10:A0 Technology Field (A10:A0) A10:A3: reserved must be set to 0 A2: 10GBase-KR A1:A0: reserved must be set to 0

Table continues on the next page...

Field	Function
4-0 TRANS_NONCE	Transmitted Nonce Transmitted Nonce Must be set to unique value per device prior to enabling auto-negotiation Defaults to 0h1F

23.6.3.12 XFI AN Advertisement 2 (XFI_AN_ADVERT2)

23.6.3.12.1 Offset

Register	Offset
XFI_AN_ADVERT2	12h

23.6.3.12.2 Function

The XFI AN Advertisement Register 2 contains bits 47:32 of the 48-bit page bits used during base page exchange. They should be programmed before auto-negotiation is started. The 48-bit value is distributed over registers 0, 1, 2, where register 2 holds the most significant 16 bits of value.

23.6.3.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FEC_CAP		TECH_A24_A11													
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.12.4 Fields

Field	Function
15-14 FEC_CAP	FEC Capability FEC Capability 00b - Device is not capable of or requesting FEC

Table continues on the next page...

Field	Function
	11b - Device is capable of and requesting FEC support
13-0 TECH_A24_A11	Technology A24:A11 Technology Field (A24:A11) A24:A11: reserved must be set to 0

23.6.3.13 XFI AN LP Base Page Ability 0 (XFI_AN_LP_BASE_PG_ABIL0)

23.6.3.13.1 Offset

Register	Offset
XFI_AN_LP_BASE_PG_ABIL0	13h

23.6.3.13.2 Function

The XFI AN LP Base Page Ability Register 0 contains bits 15:0 of the link partner base storage. The 48-bit value is stored over registers 0, 1 and 2, where register 2 holds the most significant 16 bits of the value. The contents of the registers are identical to the XFI AN avertisement registers (see above).

23.6.3.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	REM_FAULT	XNP_CAP	PAUSE_CAP		ECHOED_NONCE					SELECT_FLD				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.13.4 Fields

Field	Function
15 NP	Next Page Next Page: 0b - Link Partner does not have a Next Page to send 1b - Link Partner has a Next Page to send
14 ACK	Ack Acknowledge 0b - Base Ability data not valid 1b - Base Ability data valid
13 REM_FAULT	Remote Fault Remote Fault 0b - Normal operation 1b - Link Partner is indicating a remote fault condition
12 XNP_CAP	Extended Next Page Capability Extended Next Page Capability 0b - Link Partner does not support extended next pages 1b - Link Partner supports extended next pages
11-10 PAUSE_CAP	Pause Capability Pause Capability (ASM_DIR:PAUSE) 00b - No PAUSE 01b - Symmetric PAUSE 10b - Asymmetric PAUSE toward local device 11b - Both Symmetric PAUSE and Asymmetric PAUSE toward link partner
9-5 ECHOED_NONCE	Echoed Nonce Echoed Nonce Nonce value received by link partner
4-0 SELECT_FLD	Selector Field Selector Field 00001b - IEEE Standard 802.3

23.6.3.14 XFI AN LP Base Page Ability 1 (XFI_AN_LP_BASE_PG_ABIL1)

23.6.3.14.1 Offset

Register	Offset
XFI_AN_LP_BASE_PG_ABIL1	14h

23.6.3.14.2 Function

The XFI AN LP Base Page Ability Register 1 contains bits 31:16 of the link partner base storage. The 48-bit value is stored over registers 0, 1 and 2, where register 2 holds the most significant 16 bits of the value. The contents of the registers are identical to the XFI AN avertisement registers (see above).

23.6.3.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TECH_A10_A0											TRANS_NONCE				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.14.4 Fields

Field	Function
15-5 TECH_A10_A0	Technology A10:A0 Technology Field (A10:A0) A10:A3: reserved must be set to 0 A2: 10GBase-KR A1:A0: reserved must be set to 0
4-0 TRANS_NONCE	Transmitted Nonce Transmitted Nonce Must be set to unique value per device prior to enabling auto-negotiation

23.6.3.15 XFI AN LP Base Page Ability 2 (XFI_AN_LP_BASE_PG_ABIL2)

23.6.3.15.1 Offset

Register	Offset
XFI_AN_LP_BASE_PG_ABIL2	15h

23.6.3.15.2 Function

The XFI AN LP Base Page Ability Register 2 contains bits 47:31 of the link partner base storage. The 48-bit value is stored over registers 0, 1 and 2, where register 2 holds the most significant 16 bits of the value. The contents of the registers are identical to the XFI AN avertisement registers (see above).

23.6.3.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEC_CAP		TECH_A24_A11													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.15.4 Fields

Field	Function
15-14 FEC_CAP	FEC Capability FEC Capability (F1:F0) 00b - Link partner is not capable of or requesting FEC 01b - Link partner is capable of but not requesting FEC 11b - Link partner is capable of and requesting FEC
13-0 TECH_A24_A11	Technology A24:A11 Technology Field (A24:A11) A24:A11: reserved must be set to 0

23.6.3.16 XFI AN Extended Next Page Transmit 0 (XFI_AN_XNP_TX0)

23.6.3.16.1 Offset

Register	Offset
XFI_AN_XNP_TX0	16h

23.6.3.16.2 Function

The XFI AN XNP Transmit Register 0 contains bits 15:0 of the extended next page transmit data.

23.6.3.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NP	Reserved	MSG_PAGE	ACK2	TOGGLE	MSG_UNF_CODE_FLD										
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.3.16.4 Fields

Field	Function
15 NP	Next Page Next Page 0b - Device does not have any more Next Pages to send 1b - Device has more Next Pages to Send
14 —	Reserved
13 MSG_PAGE	Message Page Message Page 0b - Next page is an unformatted page 1b - Next page is a message page
12 ACK2	Ack 2 Acknowledge 2 0b - The receiver is not able to act on the information defined in the message 1b - The receiver is able to act on the information defined in the message
11 TOGGLE	Toggle Toggle
10-0 MSG_UNF_CODE_FLD	Message/Unformatted Code Field Message/Unformatted Code Field When MSG_PAGE=1: Message code field, else unformatted code field. For the null message code, the value is 0x1

23.6.3.17 XFI AN Extended Next Page Transmit 1 (XFI_AN_XNP_TX1)

23.6.3.17.1 Offset

Register	Offset
XFI_AN_XNP_TX1	17h

23.6.3.17.2 Function

The XFI AN XNP Transmit Register 1 contains bits 31:16 of the extended next page transmit data.

23.6.3.17.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UNF_CODE_FLD1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.17.4 Fields

Field	Function
15-0 UNF_CODE_FLD1	Unformatted Code Field 1 Unformatted Code Field 1

23.6.3.18 XFI AN Extended Next Page Transmit 2 (XFI_AN_XNP_TX2)

23.6.3.18.1 Offset

Register	Offset
XFI_AN_XNP_TX2	18h

23.6.3.18.2 Function

The XFI AN XNP Transmit Register 2 contains bits 47:32 of the extended next page transmit data.

23.6.3.18.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UNF_CODE_FLD2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.18.4 Fields

Field	Function
15-0 UNF_CODE_FLD2	Unformatted Code Field 2 Unformatted Code Field 2

23.6.3.19 XFI AN LP Extended Next Page Ability 0 (XFI_AN_LP_XNP_ABIL0)

23.6.3.19.1 Offset

Register	Offset
XFI_AN_LP_XNP_ABIL0	19h

23.6.3.19.2 Function

The XFI AN LP XNP Ability Register 0 contains bits 15:0 of the extended next page value from the remote device.

23.6.3.19.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	Reserved	MSG_PAGE	ACK2	TOGGLE	MSG_UNF_CODE_FLD										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.19.4 Fields

Field	Function
15 NP	Next Page Next Page 0b - Device does not have any more Next Pages to send 1b - Device has more Next Pages to Send
14 —	Reserved
13 MSG_PAGE	Message Page Message Page 0b - Next page is an unformatted page 1b - Next page is a message page
12 ACK2	Ack 2 Acknowledge 2 0b - The receiver is not able to act on the information defined in the message 1b - The receiver is able to act on the information defined in the message
11 TOGGLE	Toggle Toggle
10-0 MSG_UNF_CODE_FLD	Message/Unformatted Code Field Message/Unformatted Code Field When MSG_PAGE=1: Message code field, else unformatted code field. For the null message code, the value is 0x1

23.6.3.20 XFI AN LP Extended Next Page Ability 1 (XFI_AN_LP_XNP_ABIL1)

23.6.3.20.1 Offset

Register	Offset
XFI_AN_LP_XNP_ABIL1	1Ah

23.6.3.20.2 Function

The XFI AN LP XNP Ability Register 1 contains bits 31:16 of the extended next page value from the remote device.

23.6.3.20.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNF_CODE_FLD1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.20.4 Fields

Field	Function
15-0 UNF_CODE_FLD1	Unformatted Code Field 1 Unformatted Code Field 1

23.6.3.21 XFI AN LP Extended Next Page Ability 2 (XFI_AN_LP_XNP_ABIL2)

23.6.3.21.1 Offset

Register	Offset
XFI_AN_LP_XNP_ABIL2	1Bh

23.6.3.21.2 Function

The XFI AN LP XNP Ability Register 2 contains bits 47:32 of the extended next page value from the remote device.

23.6.3.21.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNF_CODE_FLD2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.21.4 Fields

Field	Function
15-0 UNF_CODE_FLD2	Unformatted Code Field 2 Unformatted Code Field 2

23.6.3.22 XFI Backplane Ethernet Status (XFI_BP_STAT)

23.6.3.22.1 Offset

Register	Offset
XFI_BP_STAT	30h

23.6.3.22.2 Function

The XFI Backplane Ethernet Status Register provides the result after auto-negotiation.

23.6.3.22.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							Reserved	Reserved	Reserved		ST_BASE_R_FEC	ST_10GBASE_KR	ST_10GBASE_KX4	ST_1000BASE_KX	BP_AN_ABIL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.3.22.4 Fields

Field	Function
15-9 —	Reserved
8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4 ST_BASE_R_FEC	Base-R FEC If 1, negotiated to perform Base-R FEC
3 ST_10GBASE_KR	10GBase-KR 10GBase-KR 0b - Not auto-negotiated to 10GBase-KR 1b - Auto-negotiated to 10GBase-KR
2 ST_10GBASE_KX4	10GBase-KX4 Always 0 after negotiation complete
1 ST_1000BASE_KX	1000Base-KX Always 0 after negotiation complete
0 BP_AN_ABIL	Backplane AN Ability Backplane Base-R capable PHY type is implemented Always 1

23.6.4 MDIO_XFI_VENDOR_SPEC register descriptions

The XFI vendor-specific 1 register space is selected when the associated XFInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 1Eh.

23.6.4.1 MDIO_XFI_VENDOR_SPEC memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	XFI Revision (XFI_VND_REV)	16	RO	1010h
1h	XFI Scratch (XFI_VND_SCRATCH)	16	RW	0000h
2h	XFI PCS Interrupt Event (XFI_VND_PCS_INT)	16	RO	0000h
3h	XFI PCS Interrupt Mask (XFI_VND_PCS_INT_MSK)	16	RW	0000h
4h	XFI Auto-Negotiation Interrupt Event (XFI_VND_AN_INT)	16	RO	0000h
5h	XFI Auto-Negotiation Interrupt Mask (XFI_VND_AN_INT_MSK)	16	RW	0000h
6h	XFI Link Training Interrupt Event (XFI_VND_LT_INT)	16	RO	0000h
7h	XFI Link Training Interrupt Mask (XFI_VND_LT_INT_MSK)	16	RW	0000h

23.6.4.2 XFI Revision (XFI_VND_REV)

23.6.4.2.1 Offset

Register	Offset
XFI_VND_REV	0h

23.6.4.2.2 Function

The XFI revision register contains revision information for the XFI PCS.

23.6.4.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CFG_VER								IP_VER				IP_REV			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

23.6.4.2.4 Fields

Field	Function
15-8 CFG_VER	Configuration Version Integration Version
7-4 IP_VER	IP Version IP Version
3-0 IP_REV	IP Revision IP Revision

23.6.4.3 XFI Scratch (XFI_VND_SCRATCH)

23.6.4.3.1 Offset

Register	Offset
XFI_VND_SCRATCH	1h

23.6.4.3.2 Function

The XFI scratch register contains a read/writeable scratch register that can be used to test MDIO register access.

23.6.4.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.3.4 Fields

Field	Function
15-0 SCRATCH	Scratch register Scratch register

23.6.4.4 XFI PCS Interrupt Event (XFI_VND_PCS_INT)

23.6.4.4.1 Offset

Register	Offset
XFI_VND_PCS_INT	2h

23.6.4.4.2 Function

The XFI PCS interrupt event register contains interrupt event bits for PCS function. Note that XFI PCS interrupt event register bits are cleared when read.

23.6.4.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						PCS_TX_LP_IDLE	PCS_RX_LP_IDLE	PCS_TX_LOC_FAULT	PCS_TX_REM_FAULT	PCS_RX_LOC_FAULT	PCS_RX_REM_FAULT	PCS_HI_BER	PCS_BLK_LK_LOSS	TX_BUF_FAULT	RX_BUF_FAULT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.4.4 Fields

Field	Function
15-10 —	Reserved
9 PCS_TX_LP_IDLE	PCS Tx Low Power Idle PCS Tx Low Power Idle Cleared on register read 0b - PCS transmit path has not detected a Low Power Idle condition 1b - PCS trnasmit path has detected a Low Power Idle condition
8	PCS Rx Low Power Idle PCS Rx Low Power Idle

Table continues on the next page...

Field	Function
PCS_RX_LP_IDLE	Cleared on register read 0b - PCS receive path has not detected a Low Power Idle condition 1b - PCS receive path has detected a Low Power Idle condition
7 PCS_TX_LOC_FAULT	PCS Tx Local Fault PCS Tx Local Fault Cleared on register read 0b - PCS transmit path has not detected a Local Fault condition 1b - PCS transmit path has detected a Local Fault condition
6 PCS_TX_REM_FAULT	PCS Tx Remote Fault PCS Tx Remote Fault Cleared on register read 0b - PCS transmit path has not detected a Remote Fault condition 1b - PCS transmit path has detected a Remote Fault condition
5 PCS_RX_LOC_FAULT	PCS Rx Local Fault PCS Rx Local Fault Cleared on register read 0b - PCS receive path has not detected a Local Fault condition 1b - PCS receive path has detected a Local Fault condition
4 PCS_RX_REM_FAULT	PCS Rx Remote Fault PCS Rx Remote Fault Cleared on register read 0b - PCS receive path has not detected a Remote Fault condition 1b - PCS receive path has detected a Remote Fault condition
3 PCS_HI_BER	PCS High BER PCS High BER Cleared on register read 0b - PCS receive path has not detected a High Bit Error Rate condition 1b - PCS receive path has detected a High Bit Error Rate condition
2 PCS_BLK_LK_LOSS	PCS Block Lock Loss PCS Block Lock Loss Cleared on register read 0b - PCS block lock has not lost synchronization 1b - PCS block lock has lost synchronization
1 TX_BUF_FAULT	Tx Buffer Fault Tx Buffer Fault Cleared on register read 0b - No buffer error detected on Transmit Rate Matching FIFO 1b - Buffer error detected on Transmit Rate Matching FIFO
0 RX_BUF_FAULT	Rx Buffer Fault Rx Buffer Fault Cleared on register read 0b - No buffer error detected on Receive Rate Matching FIFO

Field	Function
	1b - Buffer error detected on Receive Rate Matching FIFO

23.6.4.5 XFI PCS Interrupt Mask (XFI_VND_PCS_INT_MSK)

23.6.4.5.1 Offset

Register	Offset
XFI_VND_PCS_INT_MSK	3h

23.6.4.5.2 Function

The XFI PCS Interrupt Mask Register contains the interrupt mask bits for the above PCS interrupt event bits.

23.6.4.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						PCS_TX_LP_IDLE_EN	PCS_RX_LP_IDLE_EN	PCS_TX_LOC_FAULT_EN	PCS_TX_REM_FAULT_EN	PCS_RX_LOC_FAULT_EN	PCS_RX_REM_FAULT_EN	PCS_HI_BER_EN	PCS_BLK_LK_LOSS_EN	TX_BUF_FAULT_EN	RX_BUF_FAULT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.5.4 Fields

Field	Function
15-10 —	Reserved
9 PCS_TX_LP_IDLE_EN	PCS Tx Low Power Idle Interrupt Mask PCS Tx Low Power Idle Interrupt Mask

Table continues on the next page...

Field	Function
	0b - PCS Tx Low Power Idle events do not cause an interrupt 1b - PCS Tx Low Power Idle events cause an interrupt
8 PCS_RX_LP_IDLE_EN	PCS Rx Low Power Idle PCS Rx Low Power Idle 0b - PCS Rx Low Power Idle events do not cause an interrupt 1b - PCS Rx Low Power Idle events cause an interrupt
7 PCS_TX_LOC_FAULT_EN	PCS Tx Local Fault PCS Tx Local Fault 0b - PCS Tx Local Fault events do not cause an interrupt 1b - PCS Tx Local Fault events cause an interrupt
6 PCS_TX_REM_FAULT_EN	PCS Tx Remote Fault PCS Tx Remote Fault 0b - PCS Tx Remote Fault events do not cause an interrupt 1b - PCS Tx Remote Fault events cause an interrupt
5 PCS_RX_LOC_FAULT_EN	PCS Rx Local Fault PCS Rx Local Fault 0b - PCS Rx Local Fault events do not cause an interrupt 1b - PCS Rx Local Fault events cause an interrupt
4 PCS_RX_REM_FAULT_EN	PCS Rx Remote Fault PCS Rx Remote Fault 0b - PCS Rx Remote Fault events do not cause an interrupt 1b - PCS Rx Remote Fault events cause an interrupt
3 PCS_HI_BER_EN	PCS High BER PCS High BER 0b - PCS High BER events do not cause an interrupt 1b - PCS High BER events cause an interrupt
2 PCS_BLK_LK_LOSS_EN	PCS Block Lock Loss PCS Block Lock Loss 0b - PCS Block Lock Loss events do not cause an interrupt 1b - PCS Block Lock Loss events cause an interrupt
1 TX_BUF_FAULT_EN	Tx Buffer Fault Tx Buffer Fault 0b - PCS Tx Buffer Fault events do not cause an interrupt 1b - PCS Tx Buffer Fault events cause an interrupt
0 RX_BUF_FAULT_EN	Rx Buffer Fault Rx Buffer Fault 0b - PCS Rx Buffer Fault events do not cause an interrupt 1b - PCS Rx Buffer Fault events cause an interrupt

23.6.4.6 XFI Auto-Negotiation Interrupt Event (XFI_VND_AN_INT)

23.6.4.6.1 Offset

Register	Offset
XFI_VND_AN_INT	4h

23.6.4.6.2 Function

The XFI auto-negotiation interrupt event register contains interrupt events bits related to auto-negotiation. Note that XFI auto-negotiation interrrupt event register bits are cleared when read.

23.6.4.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											NEXT_PAGE	BASE_PAGE	AN_PARDET_FLT	AN_PAGE_RX	AN_DONE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.6.4 Fields

Field	Function
15-5 —	Reserved
4 NEXT_PAGE	Next Page Received Next Page Received Asserted together with AN_PAGE_RX Cleared on register read 0b - No next page received 1b - Next page received and Partner Next Page Ability register is set
3 BASE_PAGE	Base Page Received Base Page Received Asserted together with AN_PAGE_RX

Table continues on the next page...

Field	Function
	Cleared on register read 0b - No base page received 1b - Base page received and Partner Ability register is set
2 AN_PARDET_FLT	Parallel detection fault Parallel detection fault Cleared on register read 0b - No fault detected 1b - Ambiguous link status detected while waiting on DME page receive
1 AN_PAGE_RX	Auto-negotiate page receive Auto-negotiate page receive Indicates the validity of the remote ability word (base page or next page) Cleared on register read 0b - An ability word was not received from the remote device 1b - An ability word was received from the remote device during backplane auto-negotiation
0 AN_DONE	Backplane auto-negotiation complete Backplane auto-negotiation complete Cleared on register read 0b - Backplane auto-negotiation not complete 1b - Backplane auto-negotiation complete

23.6.4.7 XFI Auto-Negotiation Interrupt Mask (XFI_VND_AN_INT_MSK)

23.6.4.7.1 Offset

Register	Offset
XFI_VND_AN_INT_MSK	5h

23.6.4.7.2 Function

The XFI auto-negotiation interrupt mask register contains interrupt mask bits for the above auto-negotiate interrupt events.

23.6.4.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											NEXT_PAGE	BASE_PAGE	AN_PARDET_FLT_EN	AN_PAGE_RX_EN	AN_DONE_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.7.4 Fields

Field	Function
15-5 —	Reserved
4 NEXT_PAGE	Next Page Received Event Enable Next Page Received Event Enable 0b - AN Next Page Received will not cause an interrupt event 1b - AN Next Page Received will cause an interrupt event
3 BASE_PAGE	Base Page Received Event Enable Base Page Received Event Enable 0b - AN Base Page Received will not cause an interrupt event 1b - AN Base Page Received will cause an interrupt event
2 AN_PARDET_FLT_EN	AN Parallel Detection Fault Event Enable AN Parallel Detection Fault Event Enable 0b - AN Parallel Dection Fault will not cause an interrupt event 1b - AN Parallel Dection Fault will cause an interrupt event
1 AN_PAGE_RX_EN	AN Page Recieve Event Enable AN Page Recieve Event Enable 0b - AN Page Receive will not cause an interrupt event 1b - AN Page Receive will cause an interrupt event
0 AN_DONE_EN	AN Done Event Enable AN Done Event Enable 0b - AN done will not cause an interrupt event 1b - AN done will cause an interrupt event

23.6.4.8 XFI Link Training Interrupt Event (XFI_VND_LT_INT)

23.6.4.8.1 Offset

Register	Offset
XFI_VND_LT_INT	6h

23.6.4.8.2 Function

The XFI link training interrupt event register contains interrupt event bits related to link training. Note that XFI link training interrupt event register bits are cleared when read

23.6.4.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												TRAIN_DONE	TRAIN_FAIL	TRAIN_FRM_RCV	TRAIN_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.8.4 Fields

Field	Function
15-4 —	Reserved
3 TRAIN_DONE	Link Training Completed Link Training Completed Cleared on register read 0b - Link training not complete 1b - Link training complete
2 TRAIN_FAIL	Link Taining Fault Indication Link Taining Fault Indication Cleared on register read 0b - Link training successful or not complete 1b - Link training failed
1 TRAIN_FRM_RCV	Link Training Frame Receive Indication Link Training Frame Receive Indication Cleared on register read

Table continues on the next page...

Field	Function
	Note: during link training, training frames are received continuously, leading to assertion of this event every ~420 ns. 0b - New coefficient update and status fields not available 1b - New coefficient update and status fields available
0 TRAIN_LOCK	Link Training Lock Indication Link Training Lock Indication Cleared on register read 0b - Link training state machine is not locked 1b - Link training state machine is locked

23.6.4.9 XFI Link Training Interrupt Mask (XFI_VND_LT_INT_MSK)

23.6.4.9.1 Offset

Register	Offset
XFI_VND_LT_INT_MSK	7h

23.6.4.9.2 Function

The XFI Link Training Interrupt Mask Register contains the mask bits for the above link training interrupt events.

23.6.4.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TRAIN_DONE_EN	TRAIN_FAIL_EN	TRAIN_FRM_RCV_EN	TRAIN_LOCK_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.4.9.4 Fields

Field	Function
15-4 —	Reserved
3 TRAIN_DONE_EN	Link Training Completed Interrupt Enable Link Training Completed Interrupt Enable 0b - Link Training Complete will not cause an interrupt event 1b - Link Training Complete will cause an interrupt event
2 TRAIN_FAIL_EN	Link Taining Fault Indication Interrupt Enable Link Taining Fault Indication Interrupt Enable 0b - Link Taining Fault Indication will not cause an interrupt event 1b - Link Taining Fault Indication will cause an interrupt event
1 TRAIN_FRM_RCV_EN	Link Training Frame Receive Indication Interrupt Enable Link Training Frame Receive Indication Interrupt Enable 0b - Link Training Frame Receive Indication will not cause an interrupt event 1b - Link Training Frame Receive Indication will cause an interrupt event
0 TRAIN_LOCK_EN	Link Training Lock Indication Interrupt Enable Link Training Lock Indication Interrupt Enable 0b - Link Training Lock Indication will not cause an interrupt event 1b - Link Training Lock Indication will cause an interrupt event

23.6.5 MDIO_KX_PCS register descriptions

The 1000Base-KX PCS register space is selected when the associated SGMIInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 03h.

This register space is also used for (vendor-specific) Clause 45 access to SGMII register aliases (starting at register address 8000h).

23.6.5.1 MDIO_KX_PCS memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	KX PCS Control (KX_PCS_CR)	16	RW	1140h
1h	KX PCS Status (KX_PCS_SR)	16	RO	0009h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2h	KX PCS Device Identifier Upper (KX_PCS_DEV_ID)	16	RO	0083h
3h	KX PCS Device Identifier Lower (KX_PCS_DEV_ID_L)	16	RO	E400h
5h	KX PCS Devices In Package 0 (KX_PCS_DEV_PRES0)	16	RO	0000h
6h	KX PCS Devices In Package 1 (KX_PCS_DEV_PRES1)	16	RO	0088h
Eh	KX PCS Package Identifier Upper (KX_PCS_PKG_ID_U)	16	RO	0083h
Fh	KX PCS Package Identifier Lower (KX_PCS_PKG_ID_L)	16	RO	E400h
8000h	SGMII Control (C45_SGMII_CR)	16	RW	1140h
8001h	SGMII Status (C45_SGMII_SR)	16	RO	0009h
8002h	SGMII PHY Identifier Upper (C45_SGMII_PHY_ID_H)	16	RO	0083h
8003h	SGMII PHY Identifier Lower (C45_SGMII_PHY_ID_L)	16	RO	E400h
8004h	SGMII Device Ability for 1000Base-X (C45_SGMII_DEV_ABIL_1KBX)	16	RW	01A0h
8004h	SGMII Device Ability for SGMII (C45_SGMII_DEV_ABIL_SGMII)	16	RW	01A0h
8005h	SGMII Partner Ability for 1000Base-X (C45_SGMII_LP_DEV_ABIL_1KBX)	16	RO	0000h
8005h	SGMII Partner Ability for SGMII (C45_SGMII_LP_DEV_ABIL_SGMII)	16	RO	0000h
8006h	SGMII AN Expansion (C45_SGMII_AN_EXP)	16	RO	0004h
8007h	SGMII Next Page Transmit (C45_SGMII_NP_TX)	16	RW	0000h
8008h	SGMII LP Next Page Receive (C45_SGMII_NP_RX)	16	RO	0000h
800Fh	SGMII Extended Status (C45_SGMII_XTND_STAT)	16	RU	0000h
8010h	SGMII Scratch (C45_SGMII_SCRATCH)	16	RW	0000h
8011h	SGMII Design Revision (C45_SGMII_REV)	16	RO	0001h
8012h	SGMII Link Timer Lower (C45_SGMII_LINK_TMR_L)	16	RW	12D0h
8013h	SGMII Link Timer Upper (C45_SGMII_LINK_TMR_H)	16	RW	0013h
8014h	SGMII IF Mode (C45_SGMII_IF_MODE)	16	RW	0000h

23.6.5.2 KX PCS Control (KX_PCS_CR)

23.6.5.2.1 Offset

Register	Offset
KX_PCS_CR	0h

23.6.5.2.2 Function

The 1000Base-KX PCS Control Register contains control bits used to enable/disable functions in the PCS, and to initiate commands such as reset.

23.6.5.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RS T	LPBK	SPEED_SEL 0	Reserved	PD	Reserved				SPEED_SEL 1	Reserved					
W																
Reset	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0

23.6.5.2.4 Fields

Field	Function
15 RST	Reset Self-Clearing Read / Write Reset Command Register. When set to 1, a synchronous reset pulse is generated which resets all the PCS state machines, the Comma detection function and the 8b/10b coder / decoder. Should be set to 0 (default) for normal operation.
14 LPBK	Loopback Loopback Command. 0b - Normal operation 1b - Reserved
13 SPEED_SEL0	Speed selection (LSB) MSB,LSB 1,0: 1000 Mb/s All others reserved Note: SGMII speed is controlled with the IF Mode register
12 —	Reserved
11 PD	Power down 0b - Normal operation 1b - Power down PCS. The PCS is held in internal soft reset until the bit is cleared again.
10-7 —	Reserved
6 SPEED_SEL1	Speed selection (MSB) See SPEED_SEL0
5-0 —	Reserved

23.6.5.3 KX PCS Status (KX_PCS_SR)

23.6.5.3.1 Offset

Register	Offset
KX_PCS_SR	1h

23.6.5.3.2 Function

The 1000Base-KX PCS Status Register contains status bits on the operation of the PCS.

23.6.5.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													LINK_STAT	Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

23.6.5.3.4 Fields

Field	Function
15-3 —	Reserved
2 LINK_STAT	Link status Link Status 0b - The link is not valid. If the link synchronization is lost a 0 is latched which is cleared only after a register read access 1b - This link is valid.
1-0 —	Reserved

23.6.5.4 KX PCS Device Identifier Upper (KX_PCS_DEV_ID)

23.6.5.4.1 Offset

Register	Offset
KX_PCS_DEV_ID	2h

23.6.5.4.2 Function

The 1000Base-KX PCS Device Identifier Upper Register contains the upper half of the 32-bit Device Identifier.

23.6.5.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCS_DEV_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.5.4.4 Fields

Field	Function
15-0 PCS_DEV_ID_U	PCS Device Identifier Upper PCS Device Identifier Upper: OUI[3:18]

23.6.5.5 KX PCS Device Identifier Lower (KX_PCS_DEV_ID_L)

23.6.5.5.1 Offset

Register	Offset
KX_PCS_DEV_ID_L	3h

23.6.5.5.2 Function

The 1000Base-KX PCS Identifier Lower Register contains the lower half of the 32-bit Device Identifier.

23.6.5.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCS_DEV_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.5.5.4 Fields

Field	Function
15-0 PCS_DEV_ID_L	PCS Device Identifier Lower PCS Device Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.5.6 KX PCS Devices In Package 0 (KX_PCS_DEV_PRES0)

23.6.5.6.1 Offset

Register	Offset
KX_PCS_DEV_PRES0	5h

23.6.5.6.2 Function

The 1000Base-KX PCS Devices in Package 0 Register contains half of the PCS devices in package status.

23.6.5.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AN_PRES	TC_PRES	DTE_XS_PRES	PHY_XS_PRES	PCS_PRES	WIS_PRES	PMD_PMA_PRES	CLAUSE_22_REG_PRES
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.6.4 Fields

Field	Function
15-8 —	Reserved
7 AN_PRES	AN present 0b - Auto-Negotiation not present in package 1b - Auto-Negotiation present in package
6 TC_PRES	TC present 0b - TC not present in package 1b - TC present in package
5 DTE_XS_PRES	DTE XS present 0b - DTE XS not present in package 1b - DTE XS present in package
4 PHY_XS_PRES	PHY XS present 0b - PHY XS not present in package 1b - PHY XS present in package
3 PCS_PRES	PCS present 0b - PCS not present in package 1b - PCS present in package
2 WIS_PRES	WIS present 0b - WIS not present in package 1b - WIS present in package
1 PMD_PMA_PRES	PMD/PMA present 0b - PMA/PMD not present in package 1b - PMA/PMD present in package
0 CLAUSE_22_REG_PRES	Clause 22 registers present 0b - Clause 22 registers not present in package 1b - Clause 22 registers present in package

23.6.5.7 KX PCS Devices In Package 1 (KX_PCS_DEV_PRES1)

23.6.5.7.1 Offset

Register	Offset
KX_PCS_DEV_PRES1	6h

23.6.5.7.2 Function

The 1000Base-KX PCS Devices in Package 1 Register contains half of the PCS devices in package status.

23.6.5.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VEND_SPEC_DEV2_PRES	VEND_SPEC_DEV1_PRES	CLAUSE22_EXT_PRES	Reserved												
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

23.6.5.7.4 Fields

Field	Function
15 VEND_SPEC_DEV2_PRES	Vendor specific device 2 present 0b - Vendor specific device 2 not present in package 1b - Vendor specific device 2 present in package
14 VEND_SPEC_DEV1_PRES	Vendor specific device 1 present 0b - Vendor specific device 1 not present in package 1b - Vendor specific device 1 present in package
13 CLAUSE22_EXT_PRES	Clause 22 extension present 0b - Clause 22 extension not present in package 1b - Clause 22 extension present in package
12-0 —	Reserved

23.6.5.8 KX PCS Package Identifier Upper (KX_PCS_PKG_ID_U)

23.6.5.8.1 Offset

Register	Offset
KX_PCS_PKG_ID_U	Eh

23.6.5.8.2 Function

The 1000Base-KX PCS Package Device Identifier Upper Register contains the upper half of the 32-bit Package Identifier.

23.6.5.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKG_DEV_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.5.8.4 Fields

Field	Function
15-0 PKG_DEV_ID_U	Package Device Identifier Upper Package Device Identifier Upper: OUI[3:18]

23.6.5.9 KX PCS Package Identifier Lower (KX_PCS_PKG_ID_L)

23.6.5.9.1 Offset

Register	Offset
KX_PCS_PKG_ID_L	Fh

23.6.5.9.2 Function

The KX PCS Package Identifier Lower Register contains the lower half of the 32-bit Package Identifier.

23.6.5.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKG_DEV_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.5.9.4 Fields

Field	Function
15-0 PKG_DEV_ID_L	Package Device Identifier Lower Package Device Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.5.10 SGMII Control (C45_SGMII_CR)

23.6.5.10.1 Offset

Register	Offset
C45_SGMII_CR	8000h

23.6.5.10.2 Function

The SGMII Control Register contains control bits used to enable/disable functions in the PCS, and to initiate commands such as reset.

23.6.5.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RS T	LPBK	SPEED_SEL 0	AN_EN	PD	ISOLATE	RESTART_AN	FD	COL_TEST	SPEED_SEL 1	Reserved					
Reset	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0

23.6.5.10.4 Fields

Field	Function
15 RST	Reset Self-Clearing Read / Write Reset Command Register. When set to 1, a synchronous reset pulse is generated which resets all the PCS state machines, the Comma detection function and the 8b/10b coder / decoder. Should be set to 0 (default) for normal operation.
14 LPBK	Loopback Loopback Command. 0b - Normal operation 1b - Reserved
13 SPEED_SEL0	Speed selection (LSB) Speed selection (LSB) All others reserved Note: SGMII speed is controlled with the IF Mode register MSB,LSB 1,0: 1000 Mb/s All others reserved
12 AN_EN	AN enable Auto-negotiation enable 0b - Disable auto-negotiation 1b - Enable auto-negotiation
11 PD	Power down Power down 0b - Normal operation 1b - Power down PCS. The PCS is held in internal soft reset until the bit is cleared again.
10 ISOLATE	Isolate Isolate 0b - Normal operation 1b - Reserved
9 RESTART_AN	Restart AN Restart auto-negotiation

Table continues on the next page...

Field	Function
	This bit is self-clearing 0b - Normal operation 1b - Restart auto-negotiation
8 FD	Full duplex Duplex mode. Read-only 0b - Reserved 1b - Full duplex
7 COL_TEST	Collision test Collision test. Read-only 0b - Disable COL signal test 1b - Reserved
6 SPEED_SEL1	Speed selection (MSB) Speed selection (MSB) See SPEED_SEL0
5-0 —	Reserved

23.6.5.11 SGMII Status (C45_SGMII_SR)

23.6.5.11.1 Offset

Register	Offset
C45_SGMII_SR	8001h

23.6.5.11.2 Function

The SGMII Status Register contains status bits on the operation of the PCS.

23.6.5.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ST_100BASET4	FD_100BASEX	HD_100BASEX	FD_10MBPS	HD_10MBPS	FD_100BASET2	HD_100BASET2	EXT_STAT	Reserved		AN_COMP	REM_FAULT	AN_ABIL	LINK_STAT	JAB_DE T	EXT_CAP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

23.6.5.11.4 Fields

Field	Function
15 ST_100BASET4	100Base-T4 Read Only bit set to 0 to indicate that the PCS does not support 100Base-T4 operation
14 FD_100BASEX	100Base-X Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-X operation
13 HD_100BASEX	100Base-X Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-X operation
12 FD_10MBPS	10Mbps Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 10Mbps operation
11 HD_10MBPS	10Mbps Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 10Mbps operation
10 FD_100BASET2	100BaseT2 Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-T2 operation.
9 HD_100BASET2	100BaseT2 Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-T2 operation.
8 EXT_STAT	Extended status Read Only bit always set to 0 to indicate that the PCS does not implement an extended status register.
7-6 —	Reserved
5 AN_COMP	AN complete Auto-negotiation complete. Read Only Bit 0b - The Auto Negotiation process is not completed or Auto Negotiation is disabled 1b - The Auto Negotiation process is completed and that the Auto Negotiation control registers are valid.
4 REM_FAULT	Remote fault Read Only Bit always set to 0. The PCS does not implement a PHY specific remote fault detection optional function.
3 AN_ABIL	AN ability Auto Negotiation Ability. Read Only Bit set to „1. to indicate that the PCS supports Auto-Negotiation.
2 LINK_STAT	Link status Link Status 0b - The link is not valid. If the link synchronization is lost a 0 is latched which is cleared only after a register read access 1b - This link is valid.
1 JAB_DET	Jabber detection Read Only bit always set to 0, the Core does not support the optional Jabber detection function
0 EXT_CAP	Extended capability Read Only bit set to 1. to indicate that the Core supports extended registers

23.6.5.12 SGMII PHY Identifier Upper (C45_SGMII_PHY_ID_H)

23.6.5.12.1 Offset

Register	Offset
C45_SGMII_PHY_ID_H	8002h

23.6.5.12.2 Function

The SGMII PHY Identifier Upper Register contains the upper half of the 32-bit PHY Identifier.

23.6.5.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_IDENTIFIER_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.5.12.4 Fields

Field	Function
15-0 PHY_IDENTIFIER_U	PHY Identifier Upper PHY Identifier Upper: OUI[3:18]

23.6.5.13 SGMII PHY Identifier Lower (C45_SGMII_PHY_ID_L)

23.6.5.13.1 Offset

Register	Offset
C45_SGMII_PHY_ID_L	8003h

23.6.5.13.2 Function

The SGMII PHY Identifier Lower Register contains the lower half of the 32-bit PHY Identifier.

23.6.5.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_IDENTIFIER_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.5.13.4 Fields

Field	Function
15-0 PHY_IDENTIFIER_L	PHY Identifier Lower PHY Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.5.14 SGMII Device Ability for 1000Base-X (C45_SGMII_DEV_ABIL_1KBX)

23.6.5.14.1 Offset

Register	Offset
C45_SGMII_DEV_ABIL_1KBX	8004h

23.6.5.14.2 Function

The SGMII Device Ability Register contains the control bits used to advertise the device abilities to the Link Partner during auto-negotiation for 1000Base-X mode.

23.6.5.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	RF2	RF1	Reserved			PS2	PS1	HD	FD	Reserved				
W																
Reset	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

23.6.5.14.4 Fields

Field	Function
15 NP	Next page Next page support. Set to 1 to indicate next page can be transferred following base page exchange
14 ACK	Ack Read only bit set to 1 when device has received three consecutive matching ability values from the link partner.
13 RF2	Remote fault 2 Fault condition advertised by the device: RF1=0 / RF2=0: No error, link is OK (Reset condition). RF1=0 / RF2=1: Device advertises that it is Off Line. RF1=1 / RF2=0: Device advertises a link failure condition.. RF1=1 / RF2=1: Device advertises an Auto Negotiation error. Note: the PCS does not interpret or set these bits. It is up to the application to implement the fault functions as needed.
12 RF1	Remote Fault 1 See RF2
11-9 —	Reserved
8 PS2	Pause 2 Pause bits both set to 1 (Reset Value) to advertise that the Core supports pause on both transmit and receive. Can be set to any other value to advertise other flow control capabilities.
7 PS1	Pause 1 See PS2
6 HD	Half Duplex Half Duplex Enable. Read only bit set to 1 when the device advertises that it supports Half Duplex Mode of operation.
5 FD	Full Duplex Full Duplex Enable. Set to 1 when the device advertises that it supports Full Duplex Mode of operation
4-0 —	Reserved

23.6.5.15 SGMII Device Ability for SGMII (C45_SGMII_DEV_ABIL_SGMII)

23.6.5.15.1 Offset

Register	Offset
C45_SGMII_DEV_ABIL_SGMII	8004h

23.6.5.15.2 Function

The SGMII Device Ability Register contains the control bits used to advertise the device abilities to the Link Partner during auto-negotiation for SGMII mode.

23.6.5.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ACK	Reserved					EEE_CLK_STP_EN	Reserved							SGMII
Reset	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

23.6.5.15.4 Fields

Field	Function
15 —	Reserved
14 ACK	Ack Read only bit set to 1 when device has received three consecutive matching ability values from the link partner
13-9 —	Reserved. Should be set to 0
8 EEE_CLK_STP_EN	EEE Clock Stop Enable EEE Clock Stop Enable

Table continues on the next page...

Field	Function
	Should be set to 0 before SGMII Auto-Negotiation enable/restart, as this device does not support EEE clock stop. 0b - EEE Clock Stop Disabled 1b - EEE Clock Stop Enabled
7-1 —	Reserved. Should be set to 0
0 SGMII	SGMII SGMII mode. Should be set to 1.

23.6.5.16 SGMII Partner Ability for 1000Base-X (C45_SGMII_LP_DEV_ABIL_1KBX)

23.6.5.16.1 Offset

Register	Offset
C45_SGMII_LP_DEV_ABIL_1KBX	8005h

23.6.5.16.2 Function

The SGMII Partner Ability Register contains the bits advertised by the Link Partner during auto-negotiation for 1000Base-X mode.

23.6.5.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	RF2	RF1	Reserved			PS2	PS1	HD	FD	Reserved				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.16.4 Fields

Field	Function
15 NP	Next Page Next page support.
14	Ack

Table continues on the next page...

Field	Function
ACK	Set to 1 when link partner has received three consecutive matching ability values from the device.
13 RF2	Remote Fault 2 Fault condition advertised by the link partner: RF1/RF2: 00: No error, link is OK (Reset condition). 01: Off Line. 10: Link failure 11: Auto Negotiation error.
12 RF1	Remote Fault 1 See RF2
11-9 —	Reserved
8 PS2	Pause 2 Pause capability of link partner (ASM_DIR:PAUSE) 0b00: No Pause 0b11: Symmetric Pause 0b10: Asymmetric Pause toward link partner 0b11: Both Symmetric Pause and Asymmetric Pause toward local device
7 PS1	Pause 1 See PS2
6 HD	Half Duplex Half Duplex support
5 FD	Full Duplex Full Duplex support
4-0 —	Reserved

23.6.5.17 SGMII Partner Ability for SGMII (C45_SGMII_LP_DEV_ABIL_SGMII)

23.6.5.17.1 Offset

Register	Offset
C45_SGMII_LP_DEV_ABIL_SGMII	8005h

23.6.5.17.2 Function

The SGMII Partner Ability Register contains the capability status of the SGMII link partner.

23.6.5.17.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COP_LNK_STAT	ACK	Reserved	COP_DUPL	COP_SPD		EEE_CAP	EEE_CLK_STOP_CAP	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.17.4 Fields

Field	Function
15 COP_LNK_STAT	Copper Link Status Read only bit, used by the SGMII PHY to advertise the Link Partner Copper status: 0b - Copper interface link is down 1b - Copper interface link is up
14 ACK	Ack Read only bit set to „1. when the Link Partner Copper Interface advertises that it has that received three consecutive matching ability values from the device
13 —	Always 0
12 COP_DUPL	Copper Duplex Read only bit, used by the SGMII PHY to advertise the Link Copper duplex capability: 0b - Copper Interface resolved to Half-Duplex 1b - Copper Interface resolved to Full-Duplex
11-10 COP_SPD	Copper speed Read only bits, used by the SGMII PHY to advertise the Link Partner Copper interface speed 00b - Copper Interface Speed is 10Mbps 01b - Copper Interface Speed is 100Mbps 10b - Copper Interface Speed is Gigabit 11b - Reserved
9 EEE_CAP	EEE Capability EEE Capability

Table continues on the next page...

Field	Function
	0b - EEE not supported 1b - EEE supported
8 EEE_CLK_STOP_CAP	EEE Clock Stop Capability EEE Clock Stop Capability 0b - EEE Clock Stop not supported 1b - EEE Clock Stop supported
7-0 —	Reserved

23.6.5.18 SGMII AN Expansion (C45_SGMII_AN_EXP)

23.6.5.18.1 Offset

Register	Offset
C45_SGMII_AN_EXP	8006h

23.6.5.18.2 Function

The SGMII AN Expansion Register contains status bits indicating the PCS next page auto-negotiation capability and status.

23.6.5.18.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													NP_ABLE	PG_RCV	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

23.6.5.18.4 Fields

Field	Function
15-3 —	Reserved

Table continues on the next page...

Field	Function
2 NP_ABLE	Next Page Able Read Only bit set to 1 to indicate the PCS does support the Next Page function
1 PG_RCV	Page Received Set to 1 to indicate that a new page has been received with new partner ability available in the PCS register PARTNER_ABILITY. The bit is set to 0 (Reset value) when the system management agent performs a read access.
0 —	Reserved

23.6.5.19 SGMII Next Page Transmit (C45_SGMII_NP_TX)

23.6.5.19.1 Offset

Register	Offset
C45_SGMII_NP_TX	8007h

23.6.5.19.2 Function

The SGMII NP TX Register contains next page data to transfer to the remote devide. Writing to this register initiates a next page exchange (sets mr_np_loaded variable).

23.6.5.19.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NP	ACK	MP	ACK2	TOGGLE	DATA										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.19.4 Fields

Field	Function
15 NP	Next page Next page indication. Indicates if additional next pages will follow (1) or this is the last page (0).
14	Ack

Table continues on the next page...

Field	Function
ACK	Acknowledge bit used by the autoneg function during message transfer. It has no meaning to the application and should be written with 0 always and ignored on read.
13 MP	Message page Message page (1) or unformatted page (0) format.
12 ACK2	Ack 2 Application level acknowledge to indicate acceptance of a message. Use is defined by application layer.
11 TOGGLE	Toggle The toggle bit is used by the auto-negotiation function to keep track of message exchange. It has no meaning to the application and should be written with 0 always and ignored on read.
10-0 DATA	Data 11 bits of data which can either be a message-id (if MP bit 13=1) or unformatted data (if MP bit 13=0). See IEEE 802.3 Annex 28C for message page encodings

23.6.5.20 SGMII LP Next Page Receive (C45_SGMII_NP_RX)

23.6.5.20.1 Offset

Register	Offset
C45_SGMII_NP_RX	8008h

23.6.5.20.2 Function

The SGMII NP RX Register contains the next page data received from the remote device during the latest next page exchange. The value is overridden with every new next page. Next page transfers are controlled by the application.

23.6.5.20.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	MP	ACK2	TOGGLE	DATA										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.20.4 Fields

Field	Function
15 NP	Next page Next page indication. Indicates if additional next pages will follow (1) or this is the last page (0).
14 ACK	Ack Acknowledge bit used by the autoneg function during message transfer. It has no meaning to the application and should be written with 0 always and ignored on read.
13 MP	Message page Message page (1) or unformatted page (0) format.
12 ACK2	Ack 2 Application level acknowledge to indicate acceptance of a message. Use is defined by application layer.
11 TOGGLE	Toggle The toggle bit is used by the auto-negotiation function to keep track of message exchange. It has no meaning to the application and should be written with 0 always and ignored on read.
10-0 DATA	Data 11 bits of data which can either be a message-id (if MP bit 13=1) or unformatted data (if MP bit 13=0). See IEEE 802.3 Annex 28C for message page encodings

23.6.5.21 SGMII Extended Status (C45_SGMII_XTND_STAT)

23.6.5.21.1 Offset

Register	Offset
C45_SGMII_XTND_STAT	800Fh

23.6.5.21.2 Function

The SGMII Extended Status Register is reserved, as this device does not implement the optional extended status registers.

23.6.5.21.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.21.4 Fields

Field	Function
15-0 —	Reserved. Always 0

23.6.5.22 SGMII Scratch (C45_SGMII_SCRATCH)

23.6.5.22.1 Offset

Register	Offset
C45_SGMII_SCRATCH	8010h

23.6.5.22.2 Function

The SGMII Scratch Register provides a memory location available to test register read and write operations.

23.6.5.22.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.22.4 Fields

Field	Function
15-0 SCRATCH	Scratch Scratch field.

23.6.5.23 SGMII Design Revision (C45_SGMII_REV)

23.6.5.23.1 Offset

Register	Offset
C45_SGMII_REV	8011h

23.6.5.23.2 Function

The SGMII Revision Register contains revision information for the PCS.

23.6.5.23.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								IP_MJ				IP_MN			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.5.23.4 Fields

Field	Function
15-8 —	Reserved
7-4 IP_MJ	Major revision Major revision
3-0 IP_MN	Minor revision Minor revision

23.6.5.24 SGMII Link Timer Lower (C45_SGMII_LINK_TMR_L)

23.6.5.24.1 Offset

Register	Offset
C45_SGMII_LINK_TMR_L	8012h

23.6.5.24.2 Function

The SGMII Link Timer Register Lower contains bits 15:1 of the link timer value.

23.6.5.24.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINK_TMR_L															Reserved
Reset	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0

23.6.5.24.4 Fields

Field	Function
15-1 LINK_TMR_L	Link Timer Lower Link timer[15:1]
0 —	Reserved

23.6.5.25 SGMII Link Timer Upper (C45_SGMII_LINK_TMR_H)

23.6.5.25.1 Offset

Register	Offset
C45_SGMII_LINK_TMR_H	8013h

23.6.5.25.2 Function

The SGMII Link Timer Register Upper contains bits 20:16 of the link timer value.

23.6.5.25.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											LINK_TMR_U				
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

23.6.5.25.4 Fields

Field	Function
15-5 —	Reserved
4-0 LINK_TMR_U	Link Timer Upper Link timer[20:16]

23.6.5.26 SGMII IF Mode (C45_SGMII_IF_MODE)

23.6.5.26.1 Offset

Register	Offset
C45_SGMII_IF_MODE	8014h

23.6.5.26.2 Function

The SGMII IF Mode Register contains control bits to set the interface mode.

23.6.5.26.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SGMII_DUPLEX	SGMII_SPEED		USE_SGMII_AN	SGMII_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.5.26.4 Fields

Field	Function
15-5 —	Reserved
4 SGMII_DUPLEX	SGMII Duplex SGMII Duplex Mode: Bit ignored when SGMII_EN is set to 0 or USE_SGMII_AN is set to 1. 0b - Full duplex enabled (default)
3-2 SGMII_SPEED	SGMII Speed SGMII Speed. When the PCS operates in SGMII mode (SGMII_EN set to 1) and is programmed not to be automatically configured (USE_SGMII_AN set to 0), sets the PCS speed of operation: Bits ignored when SGMII_EN is set to 0 or USE_SGMII_AN is set to 1. 00b - 10Mbps 01b - 100Mbps 10b - Gigabit (1G or 2.5G) 11b - Reserved
1 USE_SGMII_AN	Use SGMII AN Use the SGMII Auto-Negotiation Results to Program the PCS Speed. When set to 0 (Reset Value), the PCS operation should be programmed with the register bit SGMII_SPEED and SGMII_DUPLEX. When set to 1, the PCS operation is automatically programmed with the Partner abilities advertised during Auto-Negotiation. Ignored when SGMII_EN is set to 0.
0 SGMII_EN	SGMII Enable SGMII Mode Enable. When set to '0' (Reset Value), the PCS operates in standard 1000Base-X Gigabit mode, when set to '1', the PCS operates in SGMII Mode

23.6.6 MDIO_KX_AN register descriptions

The 1000Base-KX auto-negotiation register space is selected when the associated SGMIInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 07h.

23.6.6.1 MDIO_KX_AN memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	KX AN Control (KX_AN_CR)	16	RW	0000h
1h	KX AN Status (KX_AN_SR)	16	RO	0040h
2h	KX AN Device Identifier Upper (KX_AN_DEV_ID_H)	16	RO	0083h
3h	KX AN Device Identifier Lower (KX_AN_DEV_ID_L)	16	RO	E400h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
5h	KX AN Devices in Package 0 (KX_AN_DEV_PRES0)	16	RO	0088h
6h	KX AN Devices in Package 1 (KX_AN_DEV_PRES1)	16	RO	0000h
Eh	KX AN Package Identifier Upper (KX_AN_PKG_ID_H)	16	RO	0083h
Fh	KX AN Package Identifier Lower (KX_AN_PKG_ID_L)	16	RO	E400h
10h	KX AN Advertisement 0 (KX_AN_ADVERT0)	16	RW	0C01h
11h	KX AN Advertisement 1 (KX_AN_ADVERT1)	16	RW	003Fh
12h	KX AN Advertisement 2 (KX_AN_ADVERT2)	16	RW	0000h
13h	KX AN LP Base Page Ability 0 (KX_AN_LP_BASE_PG_ABIL0)	16	RO	0000h
14h	KX AN LP Base Page Ability 1 (KX_AN_LP_BASE_PG_ABIL1)	16	RO	0000h
15h	KX AN LP Base Page Ability 2 (KX_AN_LP_BASE_PG_ABIL2)	16	RO	0000h
16h	KX AN XNP Transmit 0 (KX_AN_XNP_TX0)	16	RO	2001h
17h	KX AN XNP Transmit 1 (KX_AN_XNP_TX1)	16	RO	0000h
18h	KX AN XNP Transmit 2 (KX_AN_XNP_TX2)	16	RO	0000h
19h	KX AN LP XNP Ability 0 (KX_AN_LP_XNP_ABIL0)	16	RO	0000h
1Ah	KX AN LP XNP Ability 1 (KX_AN_LP_XNP_ABIL1)	16	RO	0000h
1Bh	KX AN LP XNP Ability 2 (KX_AN_LP_XNP_ABIL2)	16	RO	0000h
30h	KX Backplane Ethernet Status (KX_BP_STAT)	16	RO	0001h
8000h	KX Millisecond Count (KX_MS_CNT)	16	RW	9896h

23.6.6.2 KX AN Control (KX_AN_CR)

23.6.6.2.1 Offset

Register	Offset
KX_AN_CR	0h

23.6.6.2.2 Function

The 1000Base-KX AN Control Register contains control bits used to enable/disable functions in the PCS, and to initiate commands such as reset.

23.6.6.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AN_RST	Reserved	EXT_NP_CTRL	BP_AN_EN	Reserved		RESTART_BP_AN	Reserved								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.2.4 Fields

Field	Function
15 AN_RST	AN Reset AN reset This bit is self-clearing 0b - AN normal operation 1b - AN reset
14 —	Reserved
13 EXT_NP_CTRL	Extended Next Page Control Extended Next Page Control When enabled (1) transmission of next page with non-null code field is possible. The next page registers should be initialized and must be set (handshaking) every time a next page is received. When disabled (0) only null next page is transmitted in response to received next pages from link partner. Note: the next page data registers (AN XNP) are writeable only if this bit is set. 0b - Extended next pages are disabled 1b - Extended next pages are enabled
12 BP_AN_EN	Backplane AN Enable Backplane auto-negotiation enable 0b - Disable auto-negotiation 1b - Enable auto-negotiation
11-10 —	Reserved
9 RESTART_BP_AN	Restart backplane AN Restart backplane auto-negotiation This bit is self-clearing 0b - Auto-Negotiation in process, disabled, or not supported 1b - Restart Auto-Negotiation process
8-0 —	Reserved

23.6.6.3 KX AN Status (KX_AN_SR)

23.6.6.3.1 Offset

Register	Offset
KX_AN_SR	1h

23.6.6.3.2 Function

The 1000Base-KX AN Status Register contains status bits for the 1000Base-KX auto-negotiate function.

23.6.6.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						PAR_DET_FAULT	Reserved	EXT_NP_STAT	PAGE_RCV	AN_COMP	REM_FAULT	AN_ABIL	LNK_STAT	Reserved	LP_AN_ABIL
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

23.6.6.3.4 Fields

Field	Function
15-10 —	Reserved
9 PAR_DET_FAULT	Parallel Detection Fault Parallel Detection Fault Once set, stays set until the register is read. 0b - A fault has not been detected via the parallel detection function. 1b - A fault has been detected via the parallel detection function.
8 —	Reserved
7	Extended Next Page Status

Table continues on the next page...

Field	Function
EXT_NP_STAT	Extended Next Page Status This bit is a copy of AN Control[EXT_NP_CTRL] 0b - Extended next pages disabled 1b - Extended next pages enabled
6 PAGE_RCV	Page received Page received Once set, stays set until the register is read 0b - Page has not been received 1b - Page has been received
5 AN_COMP	AN Complete Auto-negotiation complete 0b - Auto-negotiation process not completed 1b - Auto-negotiation process completed
4 REM_FAULT	Remote Fault Remote Fault Once set, stays set until the register is read 0b - No remote fault condition detected 1b - Remote fault condition detected
3 AN_ABIL	AN Ability Auto-negotiation Ability Always set to 1 to indicate that PCS is able to perform auto-negotiation
2 LNK_STAT	Link Status Link Status Once cleared, stays cleared until the register is read 0b - Link is down 1b - Link is up
1 —	Reserved
0 LP_AN_ABIL	Link Partner AN Ability Link Partner Auto-Negotiation Ability 0b - Link Partner is not able to perform auto-negotiation, or Link Partner ability not yet captured 1b - Link Parner is able to perform auto-negotiation

23.6.6.4 KX AN Device Identifier Upper (KX_AN_DEV_ID_H)

23.6.6.4.1 Offset

Register	Offset
KX_AN_DEV_ID_H	2h

23.6.6.4.2 Function

The 1000Base-KX AN Device Identifier Upper Register contains the upper half of the 32-bit device identifier.

23.6.6.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AN_DEV_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.6.4.4 Fields

Field	Function
15-0 AN_DEV_ID_U	AN Device Identifier Upper PCS Device Identifier Upper: OUI[3:18]

23.6.6.5 KX AN Device Identifier Lower (KX_AN_DEV_ID_L)

23.6.6.5.1 Offset

Register	Offset
KX_AN_DEV_ID_L	3h

23.6.6.5.2 Function

The 1000Base-KX AN Device Identifier Lower Register contains the lower half of the 32-bit Device Identifier.

23.6.6.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AN_DEV_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.6.5.4 Fields

Field	Function
15-0 AN_DEV_ID_L	AN Device ID Lower PCS Device Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.6.6 KX AN Devices in Package 0 (KX_AN_DEV_PRES0)

23.6.6.6.1 Offset

Register	Offset
KX_AN_DEV_PRES0	5h

23.6.6.6.2 Function

The 1000Base-KX AN Devices in Package 0 Register contains half of the AN devices in package status.

23.6.6.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								AN_PRES	TC_PRES	DTE_XS_PRES	PHY_XS_PRES	PCS_PRES	WIS_PRES	PMD_PMA_PRES	CLAUSE_22_REG_PRES
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

23.6.6.6.4 Fields

Field	Function
15-8 —	Reserved
7 AN_PRES	AN Present 0b - Auto-Negotiation not present in package 1b - Auto-Negotiation present in package
6 TC_PRES	TC Present 0b - TC not present in package 1b - TC present in package
5 DTE_XS_PRES	DTE XS Present 0b - DTE XS not present in package 1b - DTE XS present in package
4 PHY_XS_PRES	PHY XS Present 0b - PHY XS not present in package 1b - PHY XS present in package
3 PCS_PRES	PCS Present 0b - PCS not present in package 1b - PCS present in package
2 WIS_PRES	WIS Present 0b - WIS not present in package 1b - WIS present in package
1 PMD_PMA_PRES	PMD/PMA Present 0b - PMA/PMD not present in package 1b - PMA/PMD present in package
0 CLAUSE_22_REG_PRES	Clause 22 register present 0b - Clause 22 registers not present in package 1b - Clause 22 registers present in package

23.6.6.7 KX AN Devices in Package 1 (KX_AN_DEV_PRES1)

23.6.6.7.1 Offset

Register	Offset
KX_AN_DEV_PRES1	6h

23.6.6.7.2 Function

The 1000Base-KX AN Devices in Package 1 Register contains half of the AN devices in package status.

23.6.6.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VEND_SPEC_DEV2_PRES	VEND_SPEC_DEV1_PRES	CLAUSE22_EXT_PRES	Reserved												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.7.4 Fields

Field	Function
15 VEND_SPEC_DEV2_PRES	Vendor Specific Device 2 Present 0b - Vendor specific device 2 not present in package 1b - Vendor specific device 2 present in package
14 VEND_SPEC_DEV1_PRES	Vendor Specific Device 1 Present 0b - Vendor specific device 1 not present in package 1b - Vendor specific device 1 present in package
13 CLAUSE22_EXT_PRES	Clause 22 Extension Present 0b - Clause 22 extension not present in package 1b - Clause 22 extension present in package
12-0 —	Reserved

23.6.6.8 KX AN Package Identifier Upper (KX_AN_PKG_ID_H)

23.6.6.8.1 Offset

Register	Offset
KX_AN_PKG_ID_H	Eh

23.6.6.8.2 Function

The 1000Base-KX AN Package Device Identifier Upper Register contains the upper half of the 32-bit Package Identifier.

23.6.6.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKG_DEV_ID_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.6.8.4 Fields

Field	Function
15-0 PKG_DEV_ID_U	Package Device Identifier Upper Package Device Identifier Upper: OUI[3:18]

23.6.6.9 KX AN Package Identifier Lower (KX_AN_PKG_ID_L)

23.6.6.9.1 Offset

Register	Offset
KX_AN_PKG_ID_L	Fh

23.6.6.9.2 Function

The KX AN Package Identifier Lower Register contains the lower half of the 32-bit Package Identifier.

23.6.6.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKG_DEV_ID_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.6.9.4 Fields

Field	Function
15-0 PKG_DEV_ID_L	Package Device Identifier Lower Package Device Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.6.10 KX AN Advertisement 0 (KX_AN_ADVERT0)

23.6.6.10.1 Offset

Register	Offset
KX_AN_ADVERT0	10h

23.6.6.10.2 Function

The 1000Base-KX AN Advertisement Register 0 contains bits 15:0 of the 48-bit page bits used during base page exchange. They should be programmed before auto-negotiation is started. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NP	ACK	REM_FAULT	XNP_CAP	PAUSE_CAP		ECHOED_NONCE					SELECT_FLD				
Reset	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1

23.6.6.10.4 Fields

Field	Function
15 NP	Next Page Next Page 0b - The device has no next pages to send 1b - The device has next pages to send
14 ACK	Ack Acknowledge Should always be set to 0
13 REM_FAULT	Remote Fault Remote Fault 0b - No remote fault advertised 1b - Advertise remote fault
12 XNP_CAP	Extended Next Page Capabiltity Extended Next Page Capability 0b - Device does not support extended next pages
11-10 PAUSE_CAP	Pause Capability Pause Capability (ASM_DIR:PAUSE) Default is 11 00b - No PAUSE 01b - Symmetric PAUSE 10b - Asymmetric PAUSE toward link partner 11b - Both Symmetric PAUSE and Asymmetric PAUSE toward local device
9-5 ECHOED_NONCE	Echoed Nonce Echoed Nonce field (E4:0) Contains the transmitted nonce field from the link partner Only valid if ACK is set
4-0 SELECT_FLD	Selector field Selector field (S4:0) All values not shown are reserved 00001b - IEEE Std 802.3

23.6.6.11 KX AN Advertisement 1 (KX_AN_ADVERT1)

23.6.6.11.1 Offset

Register	Offset
KX_AN_ADVERT1	11h

23.6.6.11.2 Function

The 1000Base-KX AN Advertisement Register 1 contains bits 31:16 of the 48-bit page bits used during base page exchange. They should be programmed before auto-negotiation is started. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TECH_A10_A0											TRANS_NONCE				
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

23.6.6.11.4 Fields

Field	Function
15-5 TECH_A10_A0	Technology A10:A0 Bits A10:A0 of the technology field. 5 (A0): 1000Base-KX 6 (A1): 10GBase-KX4 7 (A2): 10GBase-KR 8 (A3): reserved 9 (A4): reserved 10 (A5): reserved 11 (A6): reserved 12 (A7): reserved 13 (A8): reserved 14 (A9): reserved 15 (A10): reserved

Table continues on the next page...

Field	Function
	Must set all bits except A0 to 0
4-0 TRANS_NONCE	Transmitted Nonce Transmitted Nonce Two devices must have a different nonce for auto-negotiation to operation (i.e. a loopback will not allow auto-negotiation to complete.

23.6.6.12 KX AN Advertisement 2 (KX_AN_ADVERT2)

23.6.6.12.1 Offset

Register	Offset
KX_AN_ADVERT2	12h

23.6.6.12.2 Function

The 1000Base-KX AN Advertisement Register 0 contains bits 47:32 of the 48-bit page bits used during base page exchange. They should be programmed before auto-negotiation is started. The 48-bit value is distributed over registers 0, 1, 2. Register 2 must be written last.

23.6.6.12.3 Diagram

23.6.6.12.4 Fields

Field	Function
15-14 FEC_CAP	FEC Capability Capability bits (F1:0) for FEC support

Table continues on the next page...

Field	Function
	00b - PCS does not implement a FEC function
13-0 TECH_A24_A11	Technology A24:A11 Technology field A24:11 All bits reserved. Should be set to 0

23.6.6.13 KX AN LP Base Page Ability 0 (KX_AN_LP_BASE_PG_ABIL0)

23.6.6.13.1 Offset

Register	Offset
KX_AN_LP_BASE_PG_ABIL0	13h

23.6.6.13.2 Function

The 1000Base-KX AN LP Base Page Ability Register 0 contains bits 15:0 of the link partner base storage. The 48-bit value is stored over registers 0, 1 and 2.

23.6.6.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	REM_FAULT	XNP_CAP	PAUSE_CAP		ECHOED_NONCE					SELECT_FLD				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.13.4 Fields

Field	Function
15 NP	Next Page Next Page

Table continues on the next page...

Field	Function
	0b - The link partner has no next pages to send 1b - The link partner has next pages to send
14 ACK	Ack Acknowledge 0b - No link partner acknowledge 1b - Link partner acknowledge
13 REM_FAULT	Remote Fault Remote Fault 0b - No remote fault advertised by link partner 1b - Remote fault advertised by link partner
12 XNP_CAP	Extended Next Page Capability Extended Next Page Capability 0b - Link partner does not support extended next pages 1b - Link parnet supports extended next pages
11-10 PAUSE_CAP	Pause capability Pause Capability (ASM_DIR:PAUSE) 00b - No PAUSE 01b - Symmetric PAUSE 10b - Asymmetric PAUSE toward local device 11b - Both Symmetric PAUSE and Asymmetric PAUSE toward link partner
9-5 ECHOED_NONCE	Echoed nonce Echoed Nonce field (E4:0) Contains the transmitted nonce field from the device as seen by the link partner Only valid if ACK is set
4-0 SELECT_FLD	Selector field Selector field (S4:0) All values not shown are reserved 00001b - IEEE Std 802.3

23.6.6.14 KX AN LP Base Page Ability 1 (KX_AN_LP_BASE_PG_ABIL1)

23.6.6.14.1 Offset

Register	Offset
KX_AN_LP_BASE_PG_ABIL1	14h

23.6.6.14.2 Function

The 1000Base-KX AN LP Base Page Ability Register 1 contains bits 31:16 of the link partner base storage. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TECH_A10_A0											TRANS_NONCE				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.14.4 Fields

Field	Function
15-5 TECH_A10_A0	Technology A10:A0 Bits A10:A0 of the technology field. 5 (A0): 1000Base-KX 6 (A1): 10GBase-KX4 7 (A2): 10GBase-KR 8 (A3): reserved 9 (A4): reserved 10 (A5): reserved 11 (A6): reserved 12 (A7): reserved 13 (A8): reserved 14 (A9): reserved 15 (A10): reserved
4-0 TRANS_NONCE	Transmitted Nonce Transmitted Nonce

23.6.6.15 KX AN LP Base Page Ability 2 (KX_AN_LP_BASE_PG_ABIL2)

23.6.6.15.1 Offset

Register	Offset
KX_AN_LP_BASE_PG_ABIL2	15h

23.6.6.15.2 Function

The 1000Base-KX AN LP Base Page Ability Register 2 contains bits 47:32 of the link partner base storage. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEC_CAP		TECH_A24_A11													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.15.4 Fields

Field	Function
15-14 FEC_CAP	FEC Capability Capability bits (F1:0) for FEC support F1: FEC ability F0: FEC requested
13-0 TECH_A24_A11	Technology A24:A11 Technology field A24:11 Reserved for future technology

23.6.6.16 KX AN XNP Transmit 0 (KX_AN_XNP_TX0)

23.6.6.16.1 Offset

Register	Offset
KX_AN_XNP_TX0	16h

23.6.6.16.2 Function

The 1000Base-KX AN XNP Transmit Register 0 contains bits 15:0 of the XNP Transmit Register. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	Reserved	MSG_PAGE	ACK2	TOGGLE	MSG_UNF_CODE_FLD										
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.6.16.4 Fields

Field	Function
15 NP	Next Page Next Page 0b - Device does not have any more Next Pages to send 1b - Device has more Next Pages to Send
14 —	Reserved
13 MSG_PAGE	Message Page Message Page 0b - Next page is an unformatted page 1b - Next page is a message page
12 ACK2	Ack 2 Acknowledge 2 0b - The receiver is not able to act on the information defined in the message 1b - The receiver is able to act on the information defined in the message

Table continues on the next page...

Field	Function
11 TOGGLE	Toggle Toggle
10-0 MSG_UNF_CODE_FLD	Message/Unformatted Code Field Message/Unformatted Code Field When MSG_PAGE=1: Message code field, else unformatted code field. For the null message code, the value is 0x1

23.6.6.17 KX AN XNP Transmit 1 (KX_AN_XNP_TX1)

23.6.6.17.1 Offset

Register	Offset
KX_AN_XNP_TX1	17h

23.6.6.17.2 Function

The 1000Base-KX AN XNP Transmit Register 1 contains bits 31:16 of the XNP Transmit Register. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.17.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNF_CODE_FLD1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.17.4 Fields

Field	Function
15-0 UNF_CODE_FLD1	Unformatted Code Field 1 Unformatted Code Field 1

23.6.6.18 KX AN XNP Transmit 2 (KX_AN_XNP_TX2)

23.6.6.18.1 Offset

Register	Offset
KX_AN_XNP_TX2	18h

23.6.6.18.2 Function

The 1000Base-KX AN XNP Transmit Register 2 contains bits 48:32 of the XNP Transmit Register. The 48-bit value is distributed over registers 0, 1, 2. Register 2 must be written last.

23.6.6.18.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNF_CODE_FLD2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.18.4 Fields

Field	Function
15-0 UNF_CODE_FLD2	Unformatted Code Field 2 Unformatted Code Field 2

23.6.6.19 KX AN LP XNP Ability 0 (KX_AN_LP_XNP_ABIL0)

23.6.6.19.1 Offset

Register	Offset
KX_AN_LP_XNP_ABIL0	19h

23.6.6.19.2 Function

The 1000Base-KX AN LP XNP Ability Register 0 contains bits 15:0 of the LP XNP Ability Register. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.19.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	Reserved	MSG_PAGE	ACK2	TOGGLE	MSG_UNF_CODE_FLD										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.19.4 Fields

Field	Function
15 NP	Next Page Next Page 0b - Device does not have any more Next Pages to send 1b - Device has more Next Pages to Send
14 —	Reserved
13 MSG_PAGE	Message Page Message Page 0b - Next page is an unformatted page 1b - Next page is a message page
12 ACK2	Ack 2 Acknowledge 2 0b - The receiver is not able to act on the information defined in the message 1b - The receiver is able to act on the information defined in the message
11 TOGGLE	Toggle Toggle
10-0 MSG_UNF_CODE_FLD	Message/Unformatted Code Field Message/Unformatted Code Field When MSG_PAGE=1: Message code field, else unformatted code field. For the null message code, the value is 0x1

23.6.6.20 KX AN LP XNP Ability 1 (KX_AN_LP_XNP_ABIL1)

23.6.6.20.1 Offset

Register	Offset
KX_AN_LP_XNP_ABIL1	1Ah

23.6.6.20.2 Function

The 1000Base-KX AN LP XNP Ability Register 1 contains bits 31:16 of the XNP Transmit Register. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.20.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNF_CODE_FLD1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.20.4 Fields

Field	Function
15-0 UNF_CODE_FLD1	Unformatted Code Field 1 Unformatted Code Field 1

23.6.6.21 KX AN LP XNP Ability 2 (KX_AN_LP_XNP_ABIL2)

23.6.6.21.1 Offset

Register	Offset
KX_AN_LP_XNP_ABIL2	1Bh

23.6.6.21.2 Function

The 1000Base-KX AN LP XNP Ability Register 2 contains bits 48:32 of the XNP Transmit Register. The 48-bit value is distributed over registers 0, 1, 2.

23.6.6.21.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UNF_CODE_FLD2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.6.21.4 Fields

Field	Function
15-0 UNF_CODE_FLD2	Unformatted Code Field 2 Unformatted Code Field 2

23.6.6.22 KX Backplane Ethernet Status (KX_BP_STAT)

23.6.6.22.1 Offset

Register	Offset
KX_BP_STAT	30h

23.6.6.22.2 Function

The 1000Base-KX AN Backplane Ethernet Status Register provides the result after auto-negotiation DME page exchange completed.

23.6.6.22.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							Reserved	Reserved	Reserved		ST_BASE_R_FEC	ST_10GBASE_KR	ST_10GBASE_KX4	ST_1000BASE_KX	BP_AN_ABIL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.6.22.4 Fields

Field	Function
15-9 —	Reserved
8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4 ST_BASE_R_FEC	Base-R FEC Always 0 after negotiation complete
3 ST_10GBASE_KR	10GBase-KR Always 0 after negotiation complete
2 ST_10GBASE_KX4	10GBase-KX4 Always 0 after negotiation complete
1 ST_1000BASE_KX	1000Base-KX 1000Base-KX 0b - Not auto-negotiated to 1000Base-KX 1b - Auto-negotiated to 1000Base-KX
0 BP_AN_ABIL	Backplane AN Ability 1000Base-KX capable PHY type is implemented Always 1

23.6.6.23 KX Millisecond Count (KX_MS_CNT)

23.6.6.23.1 Offset

Register	Offset
KX_MS_CNT	8000h

23.6.6.23.2 Function

The 1000Base-KX AN Millisecond Counter Register contains the upper 16 bits of the 18-bit counter value for counting 1 ms.

23.6.6.23.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNT_MS															
Reset	1	0	0	1	1	0	0	0	1	0	0	1	0	1	1	0

23.6.6.23.4 Fields

Field	Function
15-0 COUNT_MS	Count Milliseconds Count milliseconds Upper 16-bits value of the 18-bit counter is provided for counting 1ms. The milliseconds counter operates on 8bit samples (i.e. 6.4ns) incrementing by 4 with every sample (i.e. 0x9896*4*6.4ns=1ms)

23.6.7 MDIO_KX_VENDOR_SPEC register descriptions

The 1000Base-KX vendor-specific 1 register space is selected when the associated SGMIInCR1[MDEV_PORT] matches the Ethernet MAC port address (MDIO_CTL[PORT_ADDR]) and the device address (MDIO_CTL[DEV_ADDR]) is 1Dh.

23.6.7.1 MDIO_KX_VENDOR_SPEC memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	KX Revision (KX_VND_REV)	16	RO	0114h
1h	KX Scratch (KX_VND_SCRATCH)	16	RW	0000h
2h	KX PCS Interrupt Event (KX_VND_PCS_INT)	16	RW	0000h
3h	KX PCS Interrupt Mask (KX_VND_INT_MSK)	16	RW	0000h
4h	KX AN Interrupt Event (KX_VND_AN_INT)	16	RW	0000h
5h	KX AN Interrupt Mask (KX_VND_AN_INT_MSK)	16	RW	0000h

23.6.7.2 KX Revision (KX_VND_REV)

23.6.7.2.1 Offset

Register	Offset
KX_VND_REV	0h

23.6.7.2.2 Function

The 1000Base-KX version register contains version information for the KX PHY.

23.6.7.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CFG_VER								IP_VER				IP_REV			
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0

23.6.7.2.4 Fields

Field	Function
15-8 CFG_VER	Configuration/Integration Version Configuration/Integration Version
7-4 IP_VER	IP Version IP Version

Table continues on the next page...

Field	Function
3-0 IP_REV	IP revision IP Revision

23.6.7.3 KX Scratch (KX_VND_SCRATCH)

23.6.7.3.1 Offset

Register	Offset
KX_VND_SCRATCH	1h

23.6.7.3.2 Function

The 1000Base-KX scratch register may be used to test register reads and writes.

23.6.7.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.7.3.4 Fields

Field	Function
15-0 SCRATCH	Scratch Scratch register

23.6.7.4 KX PCS Interrupt Event (KX_VND_PCS_INT)

23.6.7.4.1 Offset

Register	Offset
KX_VND_PCS_INT	2h

23.6.7.4.2 Function

The 1000Base-KX PCS interrupt event register contains detect bits for KX PCS interrupt events.

23.6.7.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															PCS_LNK_LOS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.7.4.4 Fields

Field	Function
15-1 —	Reserved
0 PCS_LNK_LOS	PCS Link LOS PCS Link Loss Cleared on register read 0b - PCS has not detected a loss of link synchronization 1b - PCS has detected a loss of link synchronization

23.6.7.5 KX PCS Interrupt Mask (KX_VND_INT_MSK)

23.6.7.5.1 Offset

Register	Offset
KX_VND_INT_MSK	3h

23.6.7.5.2 Function

The 1000Base-KX PCS Interrupt Event Register contains mask bits for KX PCS interrupt events. See the Frame Manager reference manual for details on handling of PCS interrupts.

23.6.7.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															PCS_LNK_LOS_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.7.5.4 Fields

Field	Function
15-1 —	Reserved
0 PCS_LNK_LOS_EN	PCS Link LOS Enable PCS Link Loss Event Enable 0b - A PCS loss of link synchronization will not cause an interrupt event

23.6.7.6 KX AN Interrupt Event (KX_VND_AN_INT)

23.6.7.6.1 Offset

Register	Offset
KX_VND_AN_INT	4h

23.6.7.6.2 Function

The 1000Base-KX PCS Interrupt Event Register contains detect bits for KX auto-negotiation interrupt events.

23.6.7.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													AN_PARDET_FLT	AN_PAGE_RX	AN_DONE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.7.6.4 Fields

Field	Function
15-3 —	Reserved
2 AN_PARDET_FLT	AN Parallel Detection Fault Parallel detection fault Cleared on register read 0b - No fault detected 1b - Ambiguous link status detected while waiting on DME page receive
1 AN_PAGE_RX	AN Page Rx Auto-negotiate page receive Indicates the validity of the remote ability word (base page or next page) Cleared on register read 0b - An ability word was not received from the remote device 1b - An ability word was received from the remote device during backplane auto-negotiation
0 AN_DONE	AN Done Backplane auto-negotiation complete Note: not used for 1000Base-X/SGMII auto-negotiation Cleared on register read 0b - Backplane auto-negotiation not complete 1b - Backplane auto-negotiation complete

23.6.7.7 KX AN Interrupt Mask (KX_VND_AN_INT_MSK)

23.6.7.7.1 Offset

Register	Offset
KX_VND_AN_INT_MSK	5h

23.6.7.7.2 Function

The 1000Base-KX PCS Interrupt Event Register contains mask bits for KX AN interrupt events. See the Frame Manager reference manual for details on handling of AN interrupts.

23.6.7.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													AN_PARDET_FLT_EN	AN_PAGE_RX_EN	AN_DONE_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.7.7.4 Fields

Field	Function
15-3 —	Reserved
2 AN_PARDET_FLT_EN	AN Parallel Fault Enable AN Parallel Detection Fault Event Enable 0b - AN Parallel Dection Fault will not cause an interrupt event 1b - AN Parallel Dection Fault will cause an interrupt event
1 AN_PAGE_RX_EN	AN Page Rx Enable AN Page Recieve Event Enable 0b - AN Page Receive will not cause an interrupt event 1b - AN Page Receive will cause an interrupt event
0	AN Done Enable

Field	Function
AN_DONE_EN	AN Done Event Enable 0b - AN done will not cause an interrupt event 1b - AN done will cause an interrupt event

23.6.8 MDIO_SGMII register descriptions

The SGMII MDIO register space is selected when the associated SGMIInCR1[MDEV_PORT] matches the Ethernet MAC PHY address (MDIO_CTL[PHY_ADDR]).

23.6.8.1 MDIO_SGMII memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	SGMII Control (SGMII_CR)	16	RW	1140h
1h	SGMII Status (SGMII_SR)	16	RO	0009h
2h	SGMII PHY Identifier Upper (SGMII_PHY_ID_H)	16	RO	0083h
3h	SGMII PHY Identifier Lower (SGMII_PHY_ID_L)	16	RO	E400h
4h	SGMII Device Ability for 1000Base-X (SGMII_DEV_ABIL_1KBX)	16	RW	01A0h
4h	SGMII Device Ability for SGMII (SGMII_DEV_ABIL_SGMII)	16	RW	01A0h
5h	SGMII Partner Ability for 1000Base-X (SGMII_LP_DEV_ABIL_1KBX)	16	RO	0000h
5h	SGMII Partner Ability for SGMII (SGMII_LP_DEV_ABIL_SGMII)	16	RO	0000h
6h	SGMII AN Expansion (SGMII_AN_EXP)	16	RO	0004h
7h	SGMII Next Page Transmit (SGMII_NP_TX)	16	RW	0000h
8h	SGMII LP Next Page Receive (SGMII_NP_RX)	16	RO	0000h
Fh	SGMII Extended Status (SGMII_XTND_STAT)	16	RU	0000h
10h	SGMII Scratch (SGMII_SCRATCH)	16	RW	0000h
11h	SGMII Design Revision (SGMII_REV)	16	RO	0001h
12h	SGMII Link Timer Lower (SGMII_LINK_TMR_L)	16	RW	12D0h
13h	SGMII Link Timer Upper (SGMII_LINK_TMR_H)	16	RW	0013h
14h	SGMII IF Mode (SGMII_IF_MODE)	16	RW	0000h

23.6.8.2 SGMII Control (SGMII_CR)

23.6.8.2.1 Offset

Register	Offset
SGMII_CR	0h

23.6.8.2.2 Function

The SGMII Control Register contains control bits used to enable/disable functions in the PCS, and to initiate commands such as reset.

23.6.8.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RST	LPBK	SPEED_SEL 0	AN_EN	PD	ISOLATE	RESTART_AN	FD	COL_TEST	SPEED_SEL 1	Reserved					
W																
Reset	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0

23.6.8.2.4 Fields

Field	Function
15 RST	Reset Self-Clearing Read / Write Reset Command Register. When set to 1, a synchronous reset pulse is generated which resets all the PCS state machines, the Comma detection function and the 8b/10b coder / decoder. Should be set to 0 (default) for normal operation.
14 LPBK	Loopback Loopback Command. 0b - Normal operation 1b - Reserved
13 SPEED_SEL0	Speed selection (LSB) Speed selection (LSB) All others reserved Note: SGMII speed is controlled with the IF Mode register MSB,LSB 1,0: 1000 Mb/s All others reserved
12 AN_EN	AN Enable Auto-negotiation enable

Table continues on the next page...

Field	Function
	0b - Disable auto-negotiation 1b - Enable auto-negotiation
11 PD	Power down Power down 0b - Normal operation 1b - Power down PCS. The PCS is held in internal soft reset until the bit is cleared again.
10 ISOLATE	Isolate Isolate 0b - Normal operation 1b - Reserved
9 RESTART_AN	Restart AN Restart auto-negotiation This bit is self-clearing 0b - Normal operation 1b - Restart auto-negotiation
8 FD	Full Duplex Duplex mode. Read-only 0b - Reserved 1b - Full duplex
7 COL_TEST	Collision test Collision test. Read-only 0b - Disable COL signal test 1b - Reserved
6 SPEED_SEL1	Speed selection (MSB) Speed selection (MSB) See SPEED_SEL0
5-0 —	Reserved

23.6.8.3 SGMII Status (SGMII_SR)

23.6.8.3.1 Offset

Register	Offset
SGMII_SR	1h

23.6.8.3.2 Function

The SGMII Status Register contains status bits on the operation of the PCS.

23.6.8.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ST_100BASET4	FD_100BASEX	HD_100BASEX	FD_10MBPS	HD_10MBPS	FD_100BASET2	HD_100BASET2	EXT_STAT	Reserved		AN_COMP	REM_FAULT	AN_ABIL	LINK_STAT	JAB_DE T	EXT_CAP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

23.6.8.3.4 Fields

Field	Function
15 ST_100BASET4	100Base-T4 Read Only bit set to 0 to indicate that the PCS does not support 100Base-T4 operation
14 FD_100BASEX	100Base-X Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-X operation
13 HD_100BASEX	100Base-X Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-X operation
12 FD_10MBPS	10Mbps Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 10Mbps operation
11 HD_10MBPS	10Mbps Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 10Mbps operation
10 FD_100BASET2	100Base-T2 Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-T2 operation.
9 HD_100BASET 2	100Base-T2 Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-T2 operation.
8 EXT_STAT	Extended Status Read Only bit always set to 0 to indicate that the PCS does not implement an extended status register.
7-6 —	Reserved
5 AN_COMP	AN Complete Auto-negotiation complete. Read Only Bit 0b - The Auto Negotiation process is not completed or Auto Negotiation is disabled

Table continues on the next page...

Field	Function
	1b - The Auto Negotiation process is completed and that the Auto Negotiation control registers are valid.
4 REM_FAULT	Remote Fault Read Only Bit always set to 0. The PCS does not implement a PHY specific remote fault detection optional function.
3 AN_ABIL	AN Ability Auto Negotiation Ability. Read Only Bit set to „1. to indicate that the PCS supports Auto-Negotiation.
2 LINK_STAT	Link Status Link Status 0b - The link is not valid. If the link synchronization is lost a 0 is latched which is cleared only after a register read access 1b - This link is valid.
1 JAB_DET	Jabber Detection Read Only bit always set to 0, the Core does not support the optional Jabber detection function
0 EXT_CAP	Extended Capability Read Only bit set to „1. to indicate that the Core supports extended registers

23.6.8.4 SGMII PHY Identifier Upper (SGMII_PHY_ID_H)

23.6.8.4.1 Offset

Register	Offset
SGMII_PHY_ID_H	2h

23.6.8.4.2 Function

The SGMII PHY Identifier Upper Register contains the upper half of the 32-bit PHY Identifier.

23.6.8.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_IDENTIFIER_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.8.4.4 Fields

Field	Function
15-0 PHY_IDENTIFIER_U	PHY Identifier Upper PHY Identifier Upper: OUI[3:18]

23.6.8.5 SGMII PHY Identifier Lower (SGMII_PHY_ID_L)

23.6.8.5.1 Offset

Register	Offset
SGMII_PHY_ID_L	3h

23.6.8.5.2 Function

The SGMII PHY Identifier Lower Register contains the lower half of the 32-bit PHY Identifier.

23.6.8.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_IDENTIFIER_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.8.5.4 Fields

Field	Function
15-0 PHY_IDENTIFIER_L	PHY Identifier Lower PHY Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.8.6 SGMII Device Ability for 1000Base-X (SGMII_DEV_ABIL_1KBX)

23.6.8.6.1 Offset

Register	Offset
SGMII_DEV_ABIL_1KBX	4h

23.6.8.6.2 Function

The SGMII Device Ability Register contains the control bits used to advertise the device abilities to the Link Partner during auto-negotiation for 1000Base-X mode.

23.6.8.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ACK	Reserved					EEE_CLK_STP_EN	Reserved							SGMII
Reset	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

23.6.8.6.4 Fields

Field	Function
15 —	Reserved
14 ACK	Ack Read only bit set to 1 when device has received three consecutive matching ability values from the link partner
13-9 —	Reserved. Should be set to 0
8 EEE_CLK_STP_EN	EEE Clock Stop Enable EEE Clock Stop Enable

Table continues on the next page...

Field	Function
	Should be set to 0 before SGMII Auto-Negotiation enable/restart, as this device does not support EEE clock stop. 0b - EEE Clock Stop Disabled 1b - EEE Clock Stop Enabled
7-1 —	Reserved. Should be set to 0
0 SGMII	SGMII SGMII mode. Should be set to 1.

23.6.8.7 SGMII Device Ability for SGMII (SGMII_DEV_ABIL_SGMII)

23.6.8.7.1 Offset

Register	Offset
SGMII_DEV_ABIL_SGMII	4h

23.6.8.7.2 Function

The SGMII Device Ability Register contains the control bits used to advertise the device abilities to the Link Partner during auto-negotiation for SGMII mode.

23.6.8.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ACK	Reserved					EEE_CLK_STP_EN	Reserved							SGMII
Reset	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

23.6.8.7.4 Fields

Field	Function
15	Reserved

Table continues on the next page...

Field	Function
—	
14 ACK	Ack Read only bit set to 1 when device has received three consecutive matching ability values from the link partner
13-9 —	Reserved. Should be set to 0
8 EEE_CLK_STP_EN	EEE Clock Stop Enable EEE Clock Stop Enable Should be set to 0 before SGMII Auto-Negotiation enable/restart, as this device does not support EEE clock stop. 0b - EEE Clock Stop Disabled 1b - EEE Clock Stop Enabled
7-1 —	Reserved. Should be set to 0
0 SGMII	SGMII SGMII mode. Should be set to 1.

23.6.8.8 SGMII Partner Ability for 1000Base-X (SGMII_LP_DEV_ABIL_1KBX)

23.6.8.8.1 Offset

Register	Offset
SGMII_LP_DEV_ABIL_1KBX	5h

23.6.8.8.2 Function

The SGMII Partner Ability Register contains the capability status of the 1000Base-X link partner.

23.6.8.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COP_LNK_STAT	ACK	Reserved	COP_DUPL	COP_SPD		EEE_CAP	EEE_CLK_STOP_CAP	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.8.4 Fields

Field	Function
15 COP_LNK_STAT	Copper Link Status Read only bit, used by the SGMII PHY to advertise the Link Partner Copper status: 0b - Copper interface link is down 1b - Copper interface link is up
14 ACK	Ack Read only bit set to 1. when the Link Partner Copper Interface advertises that it has that received three consecutive matching ability values from the device
13 —	Always 0
12 COP_DUPL	Copper Duplex Read only bit, used by the SGMII PHY to advertise the Link Copper duplex capability: 0b - Copper Interface resolved to Half-Duplex 1b - Copper Interface resolved to Full-Duplex
11-10 COP_SPD	Copper Speed Read only bits, used by the SGMII PHY to advertise the Link Partner Copper interface speed 00b - Copper Interface Speed is 10Mbps 01b - Copper Interface Speed is 100Mbps 10b - Copper Interface Speed is Gigabit 11b - Reserved
9 EEE_CAP	EEE Capability EEE Capability 0b - EEE not supported 1b - EEE supported
8 EEE_CLK_STOP_CAP	EEE Clock Stop Capability EEE Clock Stop Capability 0b - EEE Clock Stop not supported 1b - EEE Clock Stop supported

Table continues on the next page...

Field	Function
7-0 —	Reserved

23.6.8.9 SGMII Partner Ability for SGMII (SGMII_LP_DEV_ABIL_SGMII)

23.6.8.9.1 Offset

Register	Offset
SGMII_LP_DEV_ABIL_SGMII	5h

23.6.8.9.2 Function

The SGMII Partner Ability Register contains the capability status of the SGMII link partner.

23.6.8.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COP_LNK_STAT	ACK	Reserved	COP_DUPL	COP_SPD		EEE_CAP	EEE_CLK_STOP_CAP	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.9.4 Fields

Field	Function
15 COP_LNK_STAT	Copper Link Status Read only bit, used by the SGMII PHY to advertise the Link Partner Copper status: 0b - Copper interface link is down

Table continues on the next page...

Field	Function
	1b - Copper interface link is up
14 ACK	Ack Read only bit set to 1. when the Link Partner Copper Interface advertises that it has that received three consecutive matching ability values from the device
13 —	Always 0
12 COP_DUPL	Copper Duplex Read only bit, used by the SGMII PHY to advertise the Link Copper duplex capability: 0b - Copper Interface resolved to Half-Duplex 1b - Copper Interface resolved to Full-Duplex
11-10 COP_SPD	Copper Speed Read only bits, used by the SGMII PHY to advertise the Link Partner Copper interface speed 00b - Copper Interface Speed is 10Mbps 01b - Copper Interface Speed is 100Mbps 10b - Copper Interface Speed is Gigabit 11b - Reserved
9 EEE_CAP	EEE Capability EEE Capability 0b - EEE not supported 1b - EEE supported
8 EEE_CLK_STOP_CAP	EEE Clock Stop Capability EEE Clock Stop Capability 0b - EEE Clock Stop not supported 1b - EEE Clock Stop supported
7-0 —	Reserved

23.6.8.10 SGMII AN Expansion (SGMII_AN_EXP)

23.6.8.10.1 Offset

Register	Offset
SGMII_AN_EXP	6h

23.6.8.10.2 Function

The SGMII AN Expansion Register contains status bits indicating the PCS next page auto-negotiation capability and status.

23.6.8.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													NP_ABLE	PG_RCV	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

23.6.8.10.4 Fields

Field	Function
15-3 —	Reserved
2 NP_ABLE	Next Page Able Read Only bit set to 1 to indicate the PCS does support the Next Page function
1 PG_RCV	Page Received Set to 1 to indicate that a new page has been received with new partner ability available in the PCS register PARTNER_ABILITY. The bit is set to 0 (Reset value) when the system management agent performs a read access.
0 —	Reserved

23.6.8.11 SGMII Next Page Transmit (SGMII_NP_TX)

23.6.8.11.1 Offset

Register	Offset
SGMII_NP_TX	7h

23.6.8.11.2 Function

The SGMII NP TX Register contains next page data to transfer to the remote devide. Writing to this register initiates a next page exchange (sets mr_np_loaded variable).

23.6.8.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NP	ACK	MP	ACK2	TOGGLE	DATA										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.11.4 Fields

Field	Function
15 NP	Next Page Next page indication. Indicates if additional next pages will follow (1) or this is the last page (0).
14 ACK	Ack Acknowledge bit used by the autoneg function during message transfer. It has no meaning to the application and should be written with 0 always and ignored on read.
13 MP	Message Page Message page (1) or unformatted page (0) format.
12 ACK2	Ack 2 Application level acknowledge to indicate acceptance of a message. Use is defined by application layer.
11 TOGGLE	Toggle The toggle bit is used by the auto-negotiation function to keep track of message exchange. It has no meaning to the application and should be written with 0 always and ignored on read.
10-0 DATA	Data 11 bits of data which can either be a message-id (if MP bit 13=1) or unformatted data (if MP bit 13=0). See IEEE 802.3 Annex 28C for message page encodings

23.6.8.12 SGMII LP Next Page Receive (SGMII_NP_RX)

23.6.8.12.1 Offset

Register	Offset
SGMII_NP_RX	8h

23.6.8.12.2 Function

The SGMII NP RX Register contains the next page data received from the remote device during the latest next page exchange. The value is overridden with every new next page. Next page transfers are controlled by the application.

23.6.8.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	MP	ACK2	TOGGLE	DATA										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.12.4 Fields

Field	Function
15 NP	Next Page Next page indication. Indicates if additional next pages will follow (1) or this is the last page (0).
14 ACK	Ack Acknowledge bit used by the autoneg function during message transfer. It has no meaning to the application and should be written with 0 always and ignored on read.
13 MP	Message Page Message page (1) or unformatted page (0) format.
12 ACK2	Ack 2 Application level acknowledge to indicate acceptance of a message. Use is defined by application layer.
11 TOGGLE	Toggle The toggle bit is used by the auto-negotiation function to keep track of message exchange. It has no meaning to the application and should be written with 0 always and ignored on read.
10-0 DATA	Data 11 bits of data which can either be a message-id (if MP bit 13=1) or unformatted data (if MP bit 13=0). See IEEE 802.3 Annex 28C for message page encodings

23.6.8.13 SGMII Extended Status (SGMII_XTND_STAT)

23.6.8.13.1 Offset

Register	Offset
SGMII_XTND_STAT	Fh

23.6.8.13.2 Function

The SGMII Extended Status Register is reserved, as this device does not implement the optional extended status registers.

23.6.8.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.13.4 Fields

Field	Function
15-0 —	Reserved. Always 0

23.6.8.14 SGMII Scratch (SGMII_SCRATCH)

23.6.8.14.1 Offset

Register	Offset
SGMII_SCRATCH	10h

23.6.8.14.2 Function

The SGMII Scratch Register provides a memory location available to test register read and write operations.

23.6.8.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.14.4 Fields

Field	Function
15-0 SCRATCH	Scratch Scratch field.

23.6.8.15 SGMII Design Revision (SGMII_REV)

23.6.8.15.1 Offset

Register	Offset
SGMII_REV	11h

23.6.8.15.2 Function

The SGMII Revision Register contains revision information for the PCS.

23.6.8.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								IP_MJ				IP_MN			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.8.15.4 Fields

Field	Function
15-8	Reserved

Table continues on the next page...

Field	Function
—	
7-4 IP_MJ	Major Revision Major revision
3-0 IP_MN	Minor Revision Minor revision

23.6.8.16 SGMII Link Timer Lower (SGMII_LINK_TMR_L)

23.6.8.16.1 Offset

Register	Offset
SGMII_LINK_TMR_L	12h

23.6.8.16.2 Function

The SGMII Link Timer Register Lower contains bits 15:1 of the link timer value.

23.6.8.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINK_TMR_L															Reserved
Reset	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0

23.6.8.16.4 Fields

Field	Function
15-1 LINK_TMR_L	Link Timer Lower Link timer[15:1]
0 —	Reserved

23.6.8.17 SGMII Link Timer Upper (SGMII_LINK_TMR_H)

23.6.8.17.1 Offset

Register	Offset
SGMII_LINK_TMR_H	13h

23.6.8.17.2 Function

The SGMII Link Timer Register Upper contains bits 20:16 of the link timer value.

23.6.8.17.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											LINK_TMR_U				
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

23.6.8.17.4 Fields

Field	Function
15-5 —	Reserved
4-0 LINK_TMR_U	Link Timer Upper Link timer[20:16]

23.6.8.18 SGMII IF Mode (SGMII_IF_MODE)

23.6.8.18.1 Offset

Register	Offset
SGMII_IF_MODE	14h

23.6.8.18.2 Function

The SGMII IF Mode Register contains control bits to set the interface mode.

23.6.8.18.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SGMII_DUPLEX	SGMII_SPEED		USE_SGMII_AN	SGMII_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.8.18.4 Fields

Field	Function
15-5 —	Reserved
4 SGMII_DUPLEX	SGMII Duplex SGMII Duplex Mode: Bit ignored when SGMII_EN is set to 0 or USE_SGMII_AN is set to 1. 0b - Full duplex enabled (default)
3-2 SGMII_SPEED	SGMII Speed SGMII Speed. When the PCS operates in SGMII mode (SGMII_EN set to 1) and is programmed not to be automatically configured (USE_SGMII_AN set to 0), sets the PCS speed of operation: Bits ignored when SGMII_EN is set to 0 or USE_SGMII_AN is set to 1. 00b - 10Mbps 01b - 100Mbps 10b - Gigabit 11b - Reserved
1 USE_SGMII_AN	Use SGMII Enable Use the SGMII Auto-Negotiation Results to Program the PCS Speed. When set to 0 (Reset Value), the PCS operation should be programmed with the register bit SGMII_SPEED and SGMII_DUPLEX. When set to 1, the PCS operation is automatically programmed with the Partner abilities advertised during Auto-Negotiation. Ignored when SGMII_EN is set to 0.
0 SGMII_EN	SGMII Enable SGMII Mode Enable. When set to '0' (Reset Value), the PCS operates in standard 1000Base-X Gigabit mode, when set to '1', the PCS operates in SGMII Mode

23.6.9 MDIO_QSGMII register descriptions

The QSGMII MDIO register space is selected when the associated QSGMIInCR1[MDEV_PORT] matches the most-significant 3 bits of the Ethernet MAC PHY address (MDIO_CTL[PHY_ADDR]). The least significant 2 bits of the PHY address are used to select the specific QSGMII PCS port (0, 1, 2, or 3).

23.6.9.1 MDIO_QSGMII memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	QSGMII Control (QSGMII_CR)	16	RW	1140h
1h	QSGMII Status (QSGMII_SR)	16	RO	0009h
2h	QSGMII PHY Identifier Upper (QSGMII_PHY_ID_H)	16	RO	0083h
3h	QSGMII PHY Identifier Lower (QSGMII_PHY_ID_L)	16	RO	E400h
4h	QSGMII Device Ability for SGMII (QSGMII_DEV_ABIL_SGMII)	16	RW	01A0h
5h	QSGMII Partner Ability for SGMII (QSGMII_LP_DEV_ABIL_SGMII)	16	RO	0000h
6h	QSGMII AN Expansion (QSGMII_AN_EXP)	16	RO	0004h
7h	QSGMII Next Page Transmit (QSGMII_NP_TX)	16	RW	0000h
8h	QSGMII LP Next Page Receive (QSGMII_NP_RX)	16	RO	0000h
Fh	QSGMII Extended Status (QSGMII_XTND_STAT)	16	RU	0000h
10h	QSGMII Scratch (QSGMII_SCRATCH)	16	RW	0000h
11h	QSGMII Design Revision (QSGMII_REV)	16	RO	0001h
12h	QSGMII Link Timer Lower (QSGMII_LINK_TMR_L)	16	RW	12D0h
13h	QSGMII Link Timer Upper (QSGMII_LINK_TMR_H)	16	RW	0013h
14h	QSGMII IF Mode (QSGMII_IF_MODE)	16	RW	0000h

23.6.9.2 QSGMII Control (QSGMII_CR)

23.6.9.2.1 Offset

Register	Offset
QSGMII_CR	0h

23.6.9.2.2 Function

The QSGMII Control Register contains control bits used to enable/disable functions in the PCS, and to initiate commands such as reset.

23.6.9.2.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RST	LPBK	SPEED_SEL0	AN_EN	PD	ISOLATE	RESTART_AN	FD	COL_TEST	SPEED_SEL1	Reserved					
W																
Reset	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0

23.6.9.2.4 Fields

Field	Function
15 RST	reset Self-Clearing Read / Write Reset Command Register. When set to 1, a synchronous reset pulse is generated which resets all the PCS state machines, the Comma detection function and the 8b/10b coder / decoder. Should be set to 0 (default) for normal operation.
14 LPBK	Loopback Loopback Command. 0b - Normal operation 1b - Reserved
13 SPEED_SEL0	Speed selection (LSB) Speed selection (LSB) All others reserved Note: SGMII speed is controlled with the IF Mode register MSB,LSB 1,0: 1000 Mb/s All others reserved
12 AN_EN	AN enable Auto-negotiation enable 0b - Disable auto-negotiation 1b - Enable auto-negotiation
11 PD	Power down Power down 0b - Normal operation 1b - Power down PCS. The PCS is held in internal soft reset until the bit is cleared again.
10 ISOLATE	Isolate Isolate 0b - Normal operation 1b - Reserved
9	Restart AN

Table continues on the next page...

Field	Function
RESTART_AN	Restart auto-negotiation This bit is self-clearing 0b - Normal operation 1b - Restart auto-negotiation
8 FD	Full Duplex Duplex mode. Read-only 0b - Reserved 1b - Full duplex
7 COL_TEST	Collision test Collision test. Read-only 0b - Disable COL signal test 1b - Reserved
6 SPEED_SEL1	Speed selection (MSB) Speed selection (MSB) See SPEED_SEL0
5-0 —	Reserved

23.6.9.3 QSGMII Status (QSGMII_SR)

23.6.9.3.1 Offset

Register	Offset
QSGMII_SR	1h

23.6.9.3.2 Function

The QSGMII Status Register contains status bits on the operation of the PCS.

23.6.9.3.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ST_100BASET4	FD_100BASEX	HD_100BASEX	FD_10MBPS	HD_10MBPS	FD_100BASET2	HD_100BASET2	EXT_STAT	Reserved		AN_COMP	REM_FAULT	AN_ABIL	LINK_STAT	JAB_DE T	EXT_CAP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

23.6.9.3.4 Fields

Field	Function
15 ST_100BASET4	100Base-T4 Read Only bit set to 0 to indicate that the PCS does not support 100Base-T4 operation
14 FD_100BASEX	100Base-X Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-X operation
13 HD_100BASEX	100Base-X Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-X operation
12 FD_10MBPS	10Mbps Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 10Mbps operation
11 HD_10MBPS	10Mbps Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 10Mbps operation
10 FD_100BASET2	100Base-T2 Full Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-T2 operation.
9 HD_100BASET 2	100Base-T2 Half Duplex Read Only bit set to 0 to indicate that the PCS does not support 100Base-T2 operation.
8 EXT_STAT	Extended Status Read Only bit always set to 0 to indicate that the PCS does not implement an extended status register.
7-6 —	Reserved
5 AN_COMP	AN Complete Auto-negotiation complete. Read Only Bit 0b - The Auto Negotiation process is not completed or Auto Negotiation is disabled 1b - The Auto Negotiation process is completed and that the Auto Negotiation control registers are valid.
4 REM_FAULT	remote Fault Read Only Bit always set to 0. The PCS does not implement a PHY specific remote fault detection optional function.

Table continues on the next page...

Field	Function
3 AN_ABIL	AN Ability Auto Negotiation Ability. Read Only Bit set to „1. to indicate that the PCS supports Auto-Negotiation.
2 LINK_STAT	Link Status Link Status 0b - The link is not valid. If the link synchronization is lost a 0 is latched which is cleared only after a register read access 1b - This link is valid.
1 JAB_DET	Jabber Detect Read Only bit always set to 0, the Core does not support the optional Jabber detection function
0 EXT_CAP	Extended Capability Read Only bit set to „1. to indicate that the Core supports extended registers

23.6.9.4 QSGMII PHY Identifier Upper (QSGMII_PHY_ID_H)

23.6.9.4.1 Offset

Register	Offset
QSGMII_PHY_ID_H	2h

23.6.9.4.2 Function

The QSGMII PHY Identifier Upper Register contains the upper half of the 32-bit PHY Identifier.

23.6.9.4.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_IDENTIFIER_U															
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

23.6.9.4.4 Fields

Field	Function
15-0	PHY Identifier Upper PHY Identifier Upper:

Field	Function
PHY_IDENTIFIER_U	OUI[3:18]

23.6.9.5 QSGMII PHY Identifier Lower (QSGMII_PHY_ID_L)

23.6.9.5.1 Offset

Register	Offset
QSGMII_PHY_ID_L	3h

23.6.9.5.2 Function

The QSGMII PHY Identifier Lower Register contains the lower half of the 32-bit PHY Identifier.

23.6.9.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_IDENTIFIER_L															
W																
Reset	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

23.6.9.5.4 Fields

Field	Function
15-0 PHY_IDENTIFIER_L	PHY Identifier Lower PHY Identifier Lower: 15:10 OUI[19:24] 9:4 Manufacturer's Model Number 3:0 Revision Number

23.6.9.6 QSGMII Device Ability for SGMII (QSGMII_DEV_ABIL_SGMII)

23.6.9.6.1 Offset

Register	Offset
QSGMII_DEV_ABIL_SGMII	4h

23.6.9.6.2 Function

The QSGMII Device Ability Register contains the control bits used to advertise the device abilities to the Link Partner during auto-negotiation for SGMII mode.

23.6.9.6.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	ACK	Reserved					EEE_CLK_STP_EN	Reserved							SGMII
Reset	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0

23.6.9.6.4 Fields

Field	Function
15 —	Reserved
14 ACK	Ack Read only bit set to 1 when device has received three consecutive matching ability values from the link partner
13-9 —	Reserved. Should be set to 0
8 EEE_CLK_STP_EN	EEE Clock Stop Enable EEE Clock Stop Enable Should be set to 0 before SGMII Auto-Negotiation enable/restart, as this device does not support EEE clock stop. 0b - EEE Clock Stop Disabled 1b - EEE Clock Stop Enabled
7-1 —	Reserved. Should be set to 0

Table continues on the next page...

Field	Function
0 SGMII	SGMII SGMII mode. Should be set to 1.

23.6.9.7 QSGMII Partner Ability for SGMII (QSGMII_LP_DEV_ABIL_SGMII)

23.6.9.7.1 Offset

Register	Offset
QSGMII_LP_DEV_ABIL_SGMII	5h

23.6.9.7.2 Function

The QSGMII Partner Ability Register contains the capability status of the SGMII link partner.

23.6.9.7.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COP_LNK_STAT	ACK	Reserved	COP_DUPL	COP_SPD		EEE_CAP	EEE_CLK_STOP_CAP	Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.9.7.4 Fields

Field	Function
15 COP_LNK_STAT	copper Link Status Read only bit, used by the SGMII PHY to advertise the Link Partner Copper status: 0b - Copper interface link is down

Table continues on the next page...

Field	Function
	1b - Copper interface link is up
14 ACK	Ack Read only bit set to 1. when the Link Partner Copper Interface advertises that it has that received three consecutive matching ability values from the device
13 —	Always 0
12 COP_DUPL	Copper Duplex Read only bit, used by the SGMII PHY to advertise the Link Copper duplex capability: 0b - Copper Interface resolved to Half-Duplex 1b - Copper Interface resolved to Full-Duplex
11-10 COP_SPD	Copper Speed Read only bits, used by the SGMII PHY to advertise the Link Partner Copper interface speed 00b - Copper Interface Speed is 10Mbps 01b - Copper Interface Speed is 100Mbps 10b - Copper Interface Speed is Gigabit 11b - Reserved
9 EEE_CAP	EEE Capability EEE Capability 0b - EEE not supported 1b - EEE supported
8 EEE_CLK_STOP_CAP	EEE Clock Stop Capability EEE Clock Stop Capability 0b - EEE Clock Stop not supported 1b - EEE Clock Stop supported
7-0 —	Reserved

23.6.9.8 QSGMII AN Expansion (QSGMII_AN_EXP)

23.6.9.8.1 Offset

Register	Offset
QSGMII_AN_EXP	6h

23.6.9.8.2 Function

The QSGMII AN Expansion Register contains status bits indicating the PCS next page auto-negotiation capability and status.

23.6.9.8.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													NP_ABLE	PG_RCV	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

23.6.9.8.4 Fields

Field	Function
15-3 —	Reserved
2 NP_ABLE	Next Page Able Read Only bit set to 1 to indicate the PCS does support the Next Page function
1 PG_RCV	Page received Set to 1 to indicate that a new page has been received with new partner ability available in the PCS register PARTNER_ABILITY. The bit is set to 0 (Reset value) when the system management agent performs a read access.
0 —	Reserved

23.6.9.9 QSGMII Next Page Transmit (QSGMII_NP_TX)

23.6.9.9.1 Offset

Register	Offset
QSGMII_NP_TX	7h

23.6.9.9.2 Function

The QSGMII NP TX Register contains next page data to transfer to the remote devide. Writing to this register initiates a next page exchange (sets mr_np_loaded variable).

23.6.9.9.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NP	ACK	MP	ACK2	TOGGLE	DATA										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.9.9.4 Fields

Field	Function
15 NP	Next Page Next page indication. Indicates if additional next pages will follow (1) or this is the last page (0).
14 ACK	Ack Acknowledge bit used by the autoneg function during message transfer. It has no meaning to the application and should be written with 0 always and ignored on read.
13 MP	Message page Message page (1) or unformatted page (0) format.
12 ACK2	Ack Application level acknowledge to indicate acceptance of a message. Use is defined by application layer.
11 TOGGLE	Toggle The toggle bit is used by the auto-negotiation function to keep track of message exchange. It has no meaning to the application and should be written with 0 always and ignored on read.
10-0 DATA	Data 11 bits of data which can either be a message-id (if MP bit 13=1) or unformatted data (if MP bit 13=0). See IEEE 802.3 Annex 28C for message page encodings

23.6.9.10 QSGMII LP Next Page Receive (QSGMII_NP_RX)

23.6.9.10.1 Offset

Register	Offset
QSGMII_NP_RX	8h

23.6.9.10.2 Function

The QSGMII NP RX Register contains the next page data received from the remote device during the latest next page exchange. The value is overridden with every new next page. Next page transfers are controlled by the application.

23.6.9.10.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NP	ACK	MP	ACK2	TOGGLE	DATA										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.9.10.4 Fields

Field	Function
15 NP	Next Pge Next page indication. Indicates if additional next pages will follow (1) or this is the last page (0).
14 ACK	Ack Acknowledge bit used by the autoneg function during message transfer. It has no meaning to the application and should be written with 0 always and ignored on read.
13 MP	Message Page Message page (1) or unformatted page (0) format.
12 ACK2	Ack Application level acknowledge to indicate acceptance of a message. Use is defined by application layer.
11 TOGGLE	Toggle The toggle bit is used by the auto-negotiation function to keep track of message exchange. It has no meaning to the application and should be written with 0 always and ignored on read.
10-0 DATA	Data 11 bits of data which can either be a message-id (if MP bit 13=1) or unformatted data (if MP bit 13=0). See IEEE 802.3 Annex 28C for message page encodings

23.6.9.11 QSGMII Extended Status (QSGMII_XTND_STAT)

23.6.9.11.1 Offset

Register	Offset
QSGMII_XTND_STAT	Fh

23.6.9.11.2 Function

The QSGMII Extended Status Register is reserved, as this device does not implement the optional extended status registers.

23.6.9.11.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.9.11.4 Fields

Field	Function
15-0 —	Reserved. Always 0

23.6.9.12 QSGMII Scratch (QSGMII_SCRATCH)

23.6.9.12.1 Offset

Register	Offset
QSGMII_SCRATCH	10h

23.6.9.12.2 Function

The QSGMII Scratch Register provides a memory location available to test register read and write operations.

23.6.9.12.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.9.12.4 Fields

Field	Function
15-0 SCRATCH	Scratch Scratch field.

23.6.9.13 QSGMII Design Revision (QSGMII_REV)

23.6.9.13.1 Offset

Register	Offset
QSGMII_REV	11h

23.6.9.13.2 Function

The QSGMII Revision Register contains revision information for the PCS.

23.6.9.13.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IP_MJ				IP_MN			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

23.6.9.13.4 Fields

Field	Function
15-8	Reserved

Table continues on the next page...

Field	Function
—	
7-4 IP_MJ	Major revision Major revision
3-0 IP_MN	Minor revision Minor revision

23.6.9.14 QSGMII Link Timer Lower (QSGMII_LINK_TMR_L)

23.6.9.14.1 Offset

Register	Offset
QSGMII_LINK_TMR_L	12h

23.6.9.14.2 Function

The QSGMII Link Timer Register Lower contains bits 15:1 of the link timer value.

23.6.9.14.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINK_TMR_L															Reserved
Reset	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0

23.6.9.14.4 Fields

Field	Function
15-1 LINK_TMR_L	Link Timer Lower Link timer[15:1]
0 —	Reserved

23.6.9.15 QSGMII Link Timer Upper (QSGMII_LINK_TMR_H)

23.6.9.15.1 Offset

Register	Offset
QSGMII_LINK_TMR_H	13h

23.6.9.15.2 Function

The QSGMII Link Timer Register Upper contains bits 20:16 of the link timer value.

23.6.9.15.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											LINK_TMR_U				
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

23.6.9.15.4 Fields

Field	Function
15-5 —	Reserved
4-0 LINK_TMR_U	Link Timer Upper Link timer[20:16]

23.6.9.16 QSGMII IF Mode (QSGMII_IF_MODE)

23.6.9.16.1 Offset

Register	Offset
QSGMII_IF_MODE	14h

23.6.9.16.2 Function

The QSGMII IF Mode Register contains control bits to set the interface mode.

23.6.9.16.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SGMII_DUPLEX	SGMII_SPEED		USE_SGMII_AN	SGMII_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

23.6.9.16.4 Fields

Field	Function
15-5 —	Reserved
4 SGMII_DUPLEX	SGMII Duplex SGMII Duplex Mode: Bit ignored when SGMII_EN is set to 0 or USE_SGMII_AN is set to 1. 0b - Full duplex enabled (default)
3-2 SGMII_SPEED	SGMII Speed SGMII Speed. When the PCS operates in SGMII mode (SGMII_EN set to 1) and is programmed not to be automatically configured (USE_SGMII_AN set to 0), sets the PCS speed of operation: Bits ignored when SGMII_EN is set to 0 or USE_SGMII_AN is set to 1. 00b - 10Mbps 01b - 100Mbps 10b - Gigabit 11b - Reserved
1 USE_SGMII_AN	Use SGMII AN Use the SGMII Auto-Negotiation Results to Program the PCS Speed. When set to 0 (Reset Value), the PCS operation should be programmed with the register bit SGMII_SPEED and SGMII_DUPLEX. When set to 1, the PCS operation is automatically programmed with the Partner abilities advertised during Auto-Negotiation. Ignored when SGMII_EN is set to 0.
0 SGMII_EN	SGMII Enable SGMII Mode Enable. When set to '0' (Reset Value), the PCS operates in standard 1000Base-X Gigabit mode, when set to '1', the PCS operates in SGMII Mode

23.7 Initialization/Application Information

23.7.1 Initialization

All initialization necessary for the SerDes module to start operation is done automatically based on RCW, with the exception of the following protocols:

23.7.1.1 1G SGMII

After initializing the MAC, and prior to initiating any SGMII traffic, perform the following SGMII protocol initialization:

1. SGMII IF Mode register:
   - SGMII_EN=1
   - USE_SGMII_AN=1 if SGMII auto-negotiation is desired
   - SGMII_SPEED (if USE_SGMII_AN=0)
   - SGMII_DUPLEX (if USE_SGMII_AN=0)
2. SGMII Device Ability Register:
   - EEE_CLK_STP_EN=0
   - SGMII=1

To enable SGMII auto-negotiation, perform the following sequence:

1. SGMII Link Timer Register Upper:
   - LINK_TMR_U=0h03
2. SGMII Link Timer Register Lower:
   - LINK_TMR_L=0h06A0
3. SGMII Control Register:
   - AN_EN=1
   - RESTART_AN=1

   Note: Half duplex is not supported.

23.7.1.2 2.5G SGMII

All default SerDes settings are for 1G SGMII rather than 2.5G SGMII. The following settings need to be updated for 2.5G SGMII, following the procedure in Lane Reset and Reconfiguration, prior to initializing the SGMII PCS:

- LNmGCR1[REIDL_TH]
- LNmGCR1[REIDL_EX_SEL]
- LNmGCR1[REIDL_ET_MSB]

- LNmGCR1[ISLEW_RCTL]
- LNmGCR1[OSLEW_RCTL]
- LNmRECR0[GK2OVD]
- LNmRECR0[GK3OVD]
- LNmRECR0[GK2OVD_EN]
- LNmRECR0[GK3OVD_EN]
- LNmTECR0[TEQ_TYPE]
- LNmTECR0[RATIO_PST1Q]
- LNmTECR0[AMP_RED]

See General Control Register 1 - Lane a (LNAGCR1 - LNDGCR1), Receive Equalization Control Register 0 - Lane a (LNARECR0 - LNDRECR0) and Transmit Equalization Control Register 0 - Lane a (LNATECR0 - LNDTECR0) for details.

The PCS initialization sequence for 2.5G SGMII is the same as for 1G SGMII with auto-negotiation disabled (see 1G SGMII). Auto-negotiation is not supported for 2.5G SGMII.

23.7.1.3 1000Base-KX

All default SerDes settings are for 1G SGMII rather than 1000Base-KX. The following settings need to be updated for 1000Base-KX, following the procedure in Lane Reset and Reconfiguration, prior to performing 1000Base-KX auto-negotiation:

- LNmGCR1[REIDL_TH]
- LNmGCR1[REIDL_EX_SEL]
- LNmGCR1[REIDL_ET_MSB]
- LNmTECR0[AMP_RED]

See General Control Register 1 - Lane a (LNAGCR1 - LNDGCR1) and Transmit Equalization Control Register 0 - Lane a (LNATECR0 - LNDTECR0) for details.

After initializing the MAC and SERDES lane(s), and prior to initiating any 1000Base-KX traffic, perform the following 1000Base-KX protocol initialization:

1. Enable the 1000Base-KX AN reference clock by setting PLLnCR0[DLYDIV_SEL]=01, for the PLLn that is the clock source for the SGMII PCS (see TPLL_LES in General Control Register 0 - Lane a (LNAGCR0 - LNDGCR0)).
2. Enable the 1000Base-KX AN module by setting PCCR8[SGMIIp_KX]=1, for each SGMIIp that will run in 1000Base-KX rather than SGMII mode.
3. Initialize SGMII IF Mode register to 0x0008:
   - SGMII_EN=0

- USE_SGMII_AN=0
- SGMII_SPEED=10

4. Initialize 1000Base-KX (Clause 45) AN Advertisement Register 1:
   - TX_NONCE=unique value per device
5. Read 1000Base-KX (Clause 45) AN Status Register to clear any previous status
6. Initialize 1000Base-KX (Clause 45) AN Control Register to 0x1200:
   - AN_ENAB=1
   - RST_AN=1

23.7.1.4 QSGMII

After initializing the MACs, and prior to initiating any QSGMII traffic, perform the following per-port SGMII protocol initialization in each corresponding register in the QSGMII PCS (4 per PCS):

1. QSGMII IF Mode register:
   - SGMII_EN=1
   - USE_SGMII_AN=1 if SGMII auto-negotiation is desired
   - SGMII_SPEED (if USE_SGMII_AN=0)
   - SGMII_DUPLEX (if USE_SGMII_AN=0)
2. QSGMII Device Ability Register:
   - SGMII=1
3. QSGMII Link Timer Register Upper:
   - SGMII: LINK_TMR_U=0h03
4. QSGMII Link Timer Register Lower:
   - SGMII: LINK_TMR_L=0h06A0
5. QSGMII Control Register:
   - RESTART_EN=1

   To initialize QSGMII with auto-negotiation disabled, instead of steps 2-4 above, do:

6. QSGMII Control Register:
   - AN_EN=0

23.7.1.5 XFI/10GBase-KR

All default SerDes settings are for XFI, and only settings that differ for 10GBase-KR are listed separately. There is no additional initialization required for XFI operation.

For 10GBase-KR, the following settings need to be modified, following the procedure in Lane Reset and Reconfiguration :

- LNmTECR0[TEQ_TYPE]
- LNmTECR0[SGN_PREQ]
- LNmTECR0[RATIO_PREQ]
- LNmTECR0[RATIO_PST1Q]
- LNmTECR0[AMP_RED]
- LNmTECR0[ADPT_EQ]

See Transmit Equalization Control Register 0 - Lane a (LNATECR0 - LNDTECR0) for details.

23.7.2 Unused Lanes

Unused lanes should be powered down to save power and avoid noise on adjacent lanes.

Power down the Rx portion of the lanes by setting LNmGCR0[RX_PD]=1 and LNmGCR0[RRST_B]=0.

Power down the Tx portion of the lanes by setting LNmGCR0[TX_PD]=1 and LNmGCR0[TRST_B]=0.

The Rx and Tx portions of the lanes may be independently powered down for uni-directional lanes or links.

23.7.3 Frequency Negotiation

PCI Express and SATA auto-negotiate frequency in hardware. For SATA and PCIe 2.5/5 Gbaud, that auto-negotiation involves overriding the by-n rate select function of the SerDes lane.

For PCIe 8 Gbaud, the auto-negotiation also involves a combination of switching PLL selects, and/or reconfiguring a PLL. There are two scenarios for PLL reconfiguration:

- PCIe starts on any PLL which runs at 5GHz, and the other PLL is off. PHY wrapper will switch to the other PLL as part of Gen 3 speed switch. In this case, both PLLs must be supplied with valid reference clocks.
- PCIe starts on any PLL which runs at 5GHz, and the other PLL is being used for other protocols. PHY wrapper will reconfigure the current 5GHz PLL as part of Gen 3 speed switch. PCIe stays on the same PLL in this case. This also applies to the scenario where one PCIe port runs on all lanes.

The current state of the by-n rate selects and PLL select per lane are reflected in SRDSxLNnGCR0, as defined in General Control Register 0 - Lane a (LNAGCR0 - LNDGCR0). The current state of the PLL full-rate frequency select is reflected in SRDSxPLL, as defined in SerDes PLLa Control Register 0 (PLL1CR0 - PLL2CR0).

Note: SGMII can operate at 10 Mbps, 100 Mbps, or 1000 Mbps or 2500 Mbps, but the frequency of operation of the serial link is 1.25 GHz for 10/100/1000, and 3.125 GHz for 2500. The lower effective data rates on SGMII 10/100 are implemented by replicating symbols 10 or 100 times.

Other protocols either have a static frequency configuration, or change frequencies via a software reconfiguring sequence. See Lane Reset and Reconfiguration, and PLL Reset and Reconfiguration, for details on the software reconfiguring sequences.

23.7.4 Soft Reset and Reconfiguring Procedures

23.7.4.1 Lane Reset and Reconfiguration

To reconfigure a lane (change settings including clock divider or PLL select), perform the following sequence:

1. Put the lane(s) into reset by setting LNmGCR0[TRST_B]=0 and LNmGCR0[RRST_B]=0.
2. Wait at least 50 ns
3. Change the desired per-lane settings
4. Wait at least 120 ns
5. Take the lane(s) out of reset by setting LNmGCR0[TRST_B]=1 and LNmGCR0[RRST_B]=1

Note that if the lanes being reconfigured are grouped for multi-lane or synchronous mode, then the master source clock lane for the group must have TRST_B set to 1 after all the other lanes in the group. The master source clock lane of the group is indicated by LNmGCR0[1STLANE]=1. All lanes p<m (if LNmGCR1[TRSTDIR]=0) or p>m (if LNmGCR1[TRSTDIR]=1) of the master source clock lane until the next lane with LNmGCR0[1STLANE]=1, or the end of the SerDes, are grouped with the master source clock lane. All grouped lanes must have the same setting of TRSTDIR.

It is recommended to disable any controller connected to a lane being reconfigured prior to starting the reconfiguration sequence.

23.7.4.2 Lane Enable After Powerdown

To enable a previously powered down lane, set LNmGCR0[nX_PD]=0 (n=R or T), wait 15 us, then set LNmGCR0[nRST]=1.

Note that if a Tx lane m with LNmGCR0[1STLANE]=1 (master Tx clock lane) is powered down or reset, all Tx lanes p<m (if LNmGCR1[TRSTDIR]=0), or all Tx lanes p>m (if LNmGCR1[TRSTDIR]=1), cannot be used.

23.7.4.3 PLL Reset and Reconfiguration

To reconfigure a PLL, perform the following sequence:

1. Disable all lanes using the PLL to be reconfigured by setting PLLnRSTCTL[SDRST_B]=0
2. Wait at least 50 ns
3. Disable the PLL by setting PLLnRSTCTL[SDEN]=0 and PLLnRSTCTL[PLLRST_B]=0
4. Wait at least 100 ns
5. Change the desired per-PLL settings (VCO frequency, refclk frequency, etc.).
6. Reset the PLL by setting PLLnRSTCTL[RSTREQ]=1
7. Set PLLnRSTCTL[SDEN]=1, PLLnRSTCTL[PLLRST_B]=1, and PLLnRSTCTL[SDRST_B]=1
    - Note this step is not to be combined with setting RSTREQ=1

Note: lane reconfiguration may also be performed while the PLL is disabled by following the first three steps of the lane reset sequence in Lane Reset and Reconfiguration after changing the per-PLL settings, before setting RSTREQ=1, and the last step of the lane reset sequence after the last step of the above PLL reset sequence.

It is recommended to disable any controller connected to a lane selecting the PLL being reconfigured prior to starting the reconfiguration sequence. In the case of PCI Express, the controllers must be disabled to prevent auto-negotiation to 8Gbaud, which can include autonomous PLL reset and reconfiguration.

23.7.5 Quiesce Sequences for System Sleep

To quiece the SerDes module in preparation for System Sleep, the user must first quiesce all controllers and disable transmission/reception of packets.

Chapter 24
Serial Peripheral Interface

24.1 Introduction

The SPI module provides a synchronous serial bus for communication between a chip and an external peripheral device.

24.1.1 Block diagram

The following figure represents the module's block diagram.

Figure 24-1. SPI block diagram

24.1.2 Features

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master mode
- Slave mode
- Data streaming operation in the Slave mode with continuous slave selection
- Buffered transmit operation using transmit first in first out (TX FIFO) with a depth of 4 entries
- Support for 8- or 16-bit accesses to PUSHR_SLAVE[TXDATA]
- Buffered receive operation using the Receive FIFO (RX FIFO) with a depth of 4 entries

- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues
- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
  - 2 transfer attribute registers
  - 2 extended transfer attribute registers
  - Serial clock (SCK) with programmable polarity and phase
  - Various programmable delays
  - Programmable serial frame size: 4 to 32
    - SPI frames longer than 32 bits can be supported using the continuous selection format.
  - Continuously held chip-select capability
  - Parity control
- 4 peripheral chip selects (PCSes), expandable to 2 ^ (4 – 1) with an external demultiplexer
- Deglitching support for up to 4 peripheral chip selects (PCSes) with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO when:
  - TX FIFO is not full (TFFF)
  - RX FIFO is not empty (RFDF)
  - Command (CMD) FIFO is not full (CMDFFF)
- Interrupt conditions:
  - End of queue reached (EOQF)
  - TFFF
  - CMDFFF
  - Transfer of current frame complete (TCF)
  - Transfers due from current command frame complete (CMDTCF)
  - Attempt to transmit with an empty transmit FIFO (TFUF)
  - RFDF

- Frame received while RX FIFO is full (RFOF)
- SPI Parity Error (SPEF)
- Data present in TX FIFO while CMD FIFO is empty (TFIWF)

- Global interrupt request line
- Power-saving architectural features:
  - Support for Stop mode

24.1.3 Interface configurations

24.1.3.1 SPI configuration

The SPI configuration allows the module to send and receive serial data. This configuration allows the module to operate as a basic SPI block with internal FIFOs supporting the external queue operation. Transmitted data and received data are included in separate FIFOs. The host CPU or a DMA controller reads the received data from the RX FIFO and writes transmit data to the transmit FIFO.

For queued operations, the SPI queues can be included in system RAM, external to the module. A DMA controller or host CPU accomplishes data transfers between the queues and the module FIFOs. The following figure shows a system example with DMA, SPI, and external queues in system RAM.

Figure 24-2. SPI with queues and DMA

24.1.4 Modes of operation

The module supports the following modes of operation that can be divided into two categories:

- Module-specific modes:
  - Master mode
  - Slave mode
  - Module Disable mode
- Chip-specific modes:
  - External Stop mode
  - Debug mode

The module enters module-specific modes when the host writes to a module register. The signals external to the module control the chip-specific modes. The chip may enter these modes in parallel to the module-specific modes.

24.1.4.1 Master mode

The Master mode allows the module to initiate and control serial communication. In this mode, the module controls the following signals, which the module configures as outputs:

- SCK
- SOUT
- PCS[*x*]

24.1.4.2 Slave mode

The Slave mode allows the module to communicate with SPI bus masters. In this mode, the module responds to externally controlled serial transfers. The module configures the SCK signal and the PCS[0]/SS_b signals as inputs, which are driven by the SPI bus master.

24.1.4.3 Module Disable mode

The Module Disable mode can be used for chip-power management. In this mode, you can stop the clock to the non-memory mapped logic in the module.

24.1.4.4 External Stop mode

The External Stop mode is used for chip-power management. The module supports the Peripheral Bus Stop mode mechanism. When a request is made to enter the External Stop mode, it acknowledges the request and completes the transfer that is in progress. When the module reaches the frame boundary, it signals that the protocol clock of the module may be shut off.

NOTE

In the Slave mode, this module waits until its chip select input signal goes high before it is ready to have its clocks shut off. Note that the CS0 pad must be correctly configured to allow it to return to high when no transfer is taking place. If the pad is tied low, the module may not be correctly disabled.

24.1.4.5 Debug mode

The Debug mode is used for system development and debugging. MCR[FRZ] controls module behavior when the chip is in this mode.

- If you write a 1 to this field, the module stops all serial transfers .
- If you write a 0 to the field, this mode has no effect on the module.

24.2 Module signal descriptions

This table describes the signals on the boundary of the module that may connect off chip (in alphabetical order).

Table 24-1. Module signal descriptions

Signal	Master mode	Slave mode	I/O
HT	Hardware Trigger	Hardware Trigger	I
PCS0/SS_b	Peripheral Chip Select 0 (O)	Slave Select (I)	I/O
SCK	Serial Clock (O)	Serial Clock (I)	I/O
SIN	Serial Data In	Serial Data In	I
SOUT	Serial Data Out	Serial Data Out	O

24.2.1 HT

Master mode: Hardware Trigger (I)—

- Is honored only when the module is configured for the DSI or CSI configuration (MCR[DCONF]), the value for DSICR0[TRRE] or TRIG[TRRE] is 1, and the module is in the Running state (SR[TXRXS])
- Receives hardware triggers that initiate data transfers
- DSICR0[TPOL] specifies whether the module uses the rising or falling edges of HT as hardware triggers

Slave mode: Hardware Trigger (I)—

- Is honored only when the module is configured for the DSI or CSI configuration (MCR[DCONF]), the value for DSICR0[TRRE] or TRIG[TRRE] is 1, and the module is in the Running state (SR[TXRXS])
- Receives hardware triggers that initiate the assertion of MTRIG_b
- DSICR0[TPOL] specifies whether the module uses rising or falling edges of HT as hardware triggers.

24.2.2 PCS0/SS_b

Master mode: Peripheral Chip Select 0 (O)—Selects SPI slave to receive data transmitted from the module

Slave mode: Slave Select (I)—Selects the module to receive data transmitted from the SPI master

NOTE

Do not tie the SPI slave select pin to ground. Otherwise, SPI cannot function properly.

24.2.3 SCK

Master mode: Serial Clock (O)—Supplies a clock signal from the module to SPI slaves

Slave mode: Serial Clock (I)—Supplies a clock signal to the module from the SPI master

24.2.4 SIN

Master mode: Serial Input (I)—Receives serial data

Slave mode: Serial Input (I)—Receives serial data

24.2.5 SOUT

Master mode: Serial Output (O)—Transmits serial data

Slave mode: Serial Output (O)—Transmits serial data

24.3 SPI register descriptions

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Any write access to the POPR and RXFR*n* also results in a transfer error.

NOTE

While the module is in the running state, do not write to:

- CTAREn

24.3.1 SPI memory map

SPI base address: 210_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Module Configuration Register (MCR)	32	RW	0000_4001h
8h	Transfer Count Register (TCR)	32	RW	0000_0000h
Ch - 10h	Clock and Transfer Attributes Register (in Master mode) (CTAR0 - CTAR1)	32	RW	7800_0000h
Ch	Clock and Transfer Attributes Register (in Slave mode) (CTAR0_SLAVE)	32	RW	7800_0000h
2Ch	Status Register (SR)	32	W1C	0201_0000h
30h	DMA/Interrupt Request Select and Enable Register (RSER)	32	RW	0000_0000h
34h	PUSH TX FIFO Register In Master Mode (PUSHR)	32	RW	0000_0000h
34h	PUSH TX FIFO Register In Slave Mode (PUSHR_SLAVE)	32	RW	0000_0000h
38h	POP RX FIFO Register (POPR)	32	RO	0000_0000h
3Ch - 48h	Transmit FIFO Registers (TXFR0 - TXFR3)	32	RO	0000_0000h
7Ch - 88h	Receive FIFO Registers (RXFR0 - RXFR3)	32	RO	0000_0000h
11Ch - 120h	Clock and Transfer Attributes Register Extended (CTARE0 - CTARE1)	32	RW	0000_0001h
13Ch	Status Register Extended (SREX)	32	RO	0000_0000h

24.3.2 Module Configuration Register (MCR)

24.3.2.1 Offset

Register	Offset
MCR	0h

24.3.2.2 Function

This register includes fields to configure various attributes associated with module operations. The HALT and MDIS fields of this register can be changed any time, but the effect takes place only on the next frame boundary. Also, only these two fields can be changed when the module is in the Running state.

24.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSTR	CONT_SCKE	DCONF		FRZ	Reserved	Reserved	ROOE	Reserved				PCSIS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	MDIS	DIS_TXF	DIS_RXF	0	0	0		0				XSPI	FCPCS	PES	HALT
W					CLR_TXF	CLR_RXF										
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

24.3.2.4 Fields

Field	Function
31	Master/Slave Mode Select

Table continues on the next page...

Field	Function
MSTR	Enables either Master mode (if supported) or Slave mode (if supported) operation 0b - Enables Slave mode 1b - Enables Master mode
30 CONT_SCKE	Continuous SCK Enable Enables the SCK to run continuously 0b - Continuous SCK disabled 1b - Continuous SCK enabled
29-28 DCONF	SPI Configuration Selects among the different configurations of the module 00b - SPI 01b - Reserved 10b - Reserved 11b - Reserved
27 FRZ	Freeze Enables transfers to be stopped on the next frame boundary when the chip enters the Debug mode 0b - Do not halt serial transfers in Debug mode. 1b - Halt serial transfers in Debug mode.
26 —	Reserved
25 —	Reserved
24 ROOE	Receive FIFO Overflow Overwrite Enable In case of RX FIFO overflow, configure the module to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted to the Shift register. 0b - Incoming data is ignored. 1b - Incoming data is shifted to the Shift register.
23-20 —	Always write the reset value to this field.
19-16 PCSIS	Peripheral Chip Select x Inactive State Determines the inactive state of PCS*x*. **NOTE:** The effect of this field only takes place when the module is enabled. Ensure that the field is configured correctly before enabling the SPI interface. 0000b - The inactive state of PCSx is low. 0001b - The inactive state of PCSx is high.
15 —	Reserved
14 MDIS	Module Disable Allows the clock to be stopped to the non-memory mapped logic in the module, effectively putting it in a software-controlled, power-saving state. The reset value of this field is parametrized with a default reset value of 1. When the module is used in the Slave mode, it is recommended to retain 0 as the value of this field. 0b - Enables the module clocks 1b - Allows external logic to disable the module clocks
13 DIS_TXF	Disable Transmit FIFO When the TX FIFO is disabled, the transmit part of the module operates as a simplified, double-buffered SPI. This field can be written to only when the value of the MDIS field is 0.

Table continues on the next page...

Field	Function
	0b - TX FIFO is enabled. 1b - TX FIFO is disabled.
12 DIS_RXF	Disable Receive FIFO When the RX FIFO is disabled, the receive part of the module operates as a simplified double-buffered SPI. This field can only be written when the value of the MDIS field is 0. 0b - RX FIFO is enabled. 1b - RX FIFO is disabled.
11 CLR_TXF	Clear TX FIFO Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO counter. The value of this field is always read as 0. 0b - Do not clear the TX FIFO counter. 1b - Clear the TX FIFO counter.
10 CLR_RXF	Clear RX FIFO Flushes the RX FIFO. Writing a 1 to this field clears the RX counter. The value of this field is always read as 0. **NOTE:** After every RX FIFO clear operation (MCR[CLR_RXF] = 0b1), followed by an RX FIFO overflow (SR[RFOF] = 0b1) scenario, you must immediately perform a single POP from the RX FIFO and discard the read data. The POP and discard operations must be completed before receiving a new incoming frame. 0b - Do not clear the RX FIFO counter. 1b - Clear the RX FIFO counter.
9-8 —	Reserved
7-4 —	Reserved
3 XSPI	Extended SPI Mode Enables the usage of Command and Transfer Attribute Register Extended (CTARE) registers. These registers allow you to send up to 32-bit SPI frames. Command cycling is also enabled, which allows you to send multiple data frames using a single command frame. When MCR[DIS_TXF] is asserted, the Extended SPI mode cannot be used to transmit SPI frames that are more than 16 bits in size. 0b - Normal SPI mode. Frame size can be up to 16 bits. Command cycling is not available in this mode. 1b - Extended SPI mode. Up to 32-bit SPI frames, along with command cycling, are enabled.
2 FCPCS	Fast Continuous PCS Mode Enables the masking of “After SCK (t_{ASC})” and “PCS to SCK (t_{CSC})” delays when operating in the Continuous PCS mode. This masking is not available if Continuous SCK mode is enabled. The individual delay masks are selected via PUSHR[MASC] and PUSHR[MCSC]. The firmware should select appropriate masks when providing continuous frames via the PUSHR. 0b - Normal or slow continuous PCS mode. Masking of delays is disabled. 1b - Fast continuous PCS mode. Delays are masked via control fields in the PUSHR.
1 PES	Parity Error Stop Controls SPI operation when a parity error is detected in a received SPI frame 0b - SPI frame transmission continues. 1b - SPI frame transmission stops.
0 HALT	Halt Starts and stops frame transfers. See Start and Stop of Module transfers for details. 0b - Starts transfers

Field	Function
	1b - Stops transfers

24.3.3 Transfer Count Register (TCR)

24.3.3.1 Offset

Register	Offset
TCR	8h

24.3.3.2 Function

This register contains a counter that indicates the number of SPI transfers made. The transfer counter is intended to assist in queue management. You must not write to the TCR when the module is in the running state.

24.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPI_TCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.3.4 Fields

Field	Function
31-16 SPI_TCNT	SPI Transfer Counter Counts the number of SPI transfers that the module performs

Table continues on the next page...

Field	Function
	This field increments every time the last bit of an SPI frame is transmitted, and a value written to this field presets the counter to that value. The field is reset to 0 at the beginning of a frame if the value of PUSHR[CTCNT] is 1. If the counter is not overwritten or cleared, it eventually overflows at value 65535 to 0.
15-0 —	Reserved

24.3.4 Clock and Transfer Attributes Register (in Master mode) (CTAR0 - CTAR1)

24.3.4.1 Offset

Register	Offset
CTAR0	Ch
CTAR1	10h

24.3.4.2 Function

This set of registers is used to define different transfer attributes. You must not write to the CTARs when the module is in the running state.

In Master mode, the CTARs define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In Slave mode, a subset of the fields in CTAR0 is used to configure the slave transfer attributes.

When the module is configured as an SPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTARs is used. When the module is configured as an SPI bus slave, it uses the CTAR0 register.

24.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DBR	FMSZ				CPOL	CPHA	LSBFE	PCSSCK		PASC		PDT		PBR	
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSSCK				ASC				DT				BR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.4.4 Fields

Field	Function
31 DBR	Double Baud Rate Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in Master mode. It effectively halves the baud rate division ratio, supporting faster frequencies, and odd division ratios for SCK. If the value of this field is 1, the duty cycle of the SCK depends on the value in the baud rate prescaler and the CPHA field, as listed in the following table. See the description of the VR field in this register for details on how to compute the baud rate.

Table 24-2. SPI SCK duty cycle

DBR	CPHA	PBR	SCK duty cycle
0	Any	Any	50/50
1	0	00	50/50
1	0	01	33/66
1	0	10	40/60
1	0	11	43/57
1	1	00	50/50
1	1	01	66/33
1	1	10	60/40
1	1	11	57/43

0b - The baud rate is computed normally with a 50/50 duty cycle.
1b - The baud rate is doubled with the duty cycle depending on the baud rate prescaler.

30-27 FMSZ	Frame Size The number of bits transferred per frame is equal to the FMSZ value plus 1. Regardless of the transmission mode, the minimum valid frame size value is 4.
26	Clock Polarity

Table continues on the next page...

Field	Function
CPOL	Selects the inactive state of the SCK. This field is used in both Master and Slave modes. For successful communication between serial devices, the devices must have identical clock polarities. When the continuous selection format is selected, switching between clock polarities without stopping the module can cause errors in the transfer because of the peripheral device interpreting the switch of clock polarity as a valid clock edge. **NOTE:** In case of Continuous SCK mode, when the module goes in low-power mode (disabled), the inactive state of SCK is not guaranteed. 0b - The inactive state value of SCK is low. 1b - The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This field is used in both Master and Slave modes. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the field value is ignored and the transfers are done considering 1 as the value of this field. 0b - Data is captured on the leading edge of SCK and changed on the following edge. 1b - Data is changed on the leading edge of SCK and captured on the following edge.
24 LSBFE	LSB First Specifies whether the LSB or MSB of the frame is transferred first. 0b - Data is transferred to MSB first. 1b - Data is transferred to LSB first.
23-22 PCSSCK	PCS to SCK Delay Prescaler Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK delay. See PCS to SCK Delay (t_{CSC}) for details. 00b - PCS to SCK prescaler value is 1. 01b - PCS to SCK prescaler value is 3. 10b - PCS to SCK prescaler value is 5. 11b - PCS to SCK prescaler value is 7.
21-20 PASC	After SCK Delay Prescaler Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the "After SCK Delay". See After SCK Delay (t_{ASC}) for details. 00b - Delay after Transfer Prescaler value is 1. 01b - Delay after Transfer Prescaler value is 3. 10b - Delay after Transfer Prescaler value is 5. 11b - Delay after Transfer Prescaler value is 7.
19-18 PDT	Delay after Transfer Prescaler Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. This field is only used in Master mode. See the DT field description for details on how to compute the "Delay after Transfer". See Delay after Transfer (t_{DT}) for details. 00b - Delay after Transfer Prescaler value is 1. 01b - Delay after Transfer Prescaler value is 3. 10b - Delay after Transfer Prescaler value is 5. 11b - Delay after Transfer Prescaler value is 7.
17-16 PBR	Baud Rate Prescaler Selects the prescaler value for the baud rate. This field is used only in Master mode. The baud rate is the frequency of the SCK. The protocol clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate. 00b - Baud Rate Prescaler value is 2. 01b - Baud Rate Prescaler value is 3.

Table continues on the next page...

Field	Function
	10b - Baud Rate Prescaler value is 5. 11b - Baud Rate Prescaler value is 7.
15-12 CSSCK	PCS to SCK Delay Scaler Selects the scaler value for the PCS to SCK delay. This field is used only in master mode. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the protocol clock period, and it is computed according to the following equation: $t_{CSC} = (1/f_P) \times PCSSCK \times CSSCK$. The following table lists the delay scaler values. **Table 24-3. Delay scaler encoding** See PCS to SCK Delay (t_{CSC}) for details.

Field value	Delay scaler value
0000	2
0001	4
0010	8
0011	16
0100	32
0101	64
0110	128
0111	256
1000	512
1001	1024
1010	2048
1011	4096
1100	8192
1101	16384
1110	32768
1111	65536

11-8 ASC	After SCK Delay Scaler Selects the scaler value for the "After SCK Delay". This field is used only in master mode. The "After SCK Delay" is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the protocol clock period, and it is computed according to the following equation: $t_{ASC} = (1/f_P) \times PASC \times ASC$ See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. See After SCK Delay (t_{ASC}) for details.
7-4 DT	Delay After Transfer Scaler Selects the "Delay after Transfer" scaler. This field is used only in Master mode. The "Delay after Transfer" is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. In the continuous serial communications clock operation, the DT value is fixed to one SCK clock period. The "Delay after Transfer" is a multiple of the protocol clock period, and it is computed according to the following equation:

Table continues on the next page...

Field	Function
	$t_{DT} = (1/f_P) \times PDT \times DT$ See Delay Scaler Encoding table in the CTAR*n*[CSSCK] field description for scaler values.
3-0 BR	Baud Rate Scaler Selects the scaler value for the baud rate. This field is used only in master mode. The prescaled protocol clock is divided by the baud rate scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation: SCK baud rate = $(f_P/PBR) \times [(1+DBR)/BR]$ The following table lists the baud rate scaler values.

Table 24-4. Baud rate scaler

CTARn[BR]	Baud rate scaler value
0000	2
0001	4
0010	6
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1024
1011	2048
1100	4096
1101	8192
1110	16384
1111	32768

24.3.5 Clock and Transfer Attributes Register (in Slave mode) (CTAR0_SLAVE)

24.3.5.1 Offset

Register	Offset
CTAR0_SLAVE	Ch

24.3.5.2 Function

When the module is configured as an SPI bus slave, the CTAR0 register is used.

24.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FMSZ					CPOL	CPHA	PE	PP	Reserved	Reserved					
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.5.4 Fields

Field	Function
31-27 FMSZ	Frame Size The number of bits transferred per frame is equal to the value of this field plus 1. Note that the minimum valid value of frame size is 4.
26 CPOL	Clock Polarity Selects the inactive state of the SCK. **NOTE:** In case of Continuous SCK mode, when the module goes in low-power mode (disabled), the inactive state of SCK is not guaranteed. 0b - The inactive state value of SCK is low. 1b - The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This field is used in both Master and Slave modes. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the field value is ignored and the transfers are done considering 1 as the value for this field. 0b - Data is captured on the leading edge of SCK and changed on the following edge. 1b - Data is changed on the leading edge of SCK and captured on the following edge.
24 PE	Parity Enable Enables parity bit transmission and reception for the frame 0b - No parity bit included/checked 1b - Parity bit is transmitted instead of last data bit in frame; and parity is checked for the received frame

Table continues on the next page...

Field	Function
23 PP	Parity Polarity Controls polarity of the parity bit transmitted and checked 0b - Even parity: the number of 1 bits in the transmitted frame is even. SR[SPEF] is set if the number of 1 bits is odd in the received frame. 1b - Odd parity: the number of 1 bits in the transmitted frame is odd. SR[SPEF] is set if the number of 1 bits is even in the received frame.
22 —	Reserved
21-0 —	Reserved

24.3.6 Status Register (SR)

24.3.6.1 Offset

Register	Offset
SR	2Ch

24.3.6.2 Function

This register includes status and flag fields, which reflect the status of the module and indicate the occurrence of events that can generate interrupts or DMA requests. The software can clear the fields of this register by writing a 1, and writing a 0 does not effect the fields. The register may not be writeable in the Module Disable mode because of the use of power-saving mechanisms.

24.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	BSYF	CMDTCF	0	SPEF	0	RFOF	TFIWF	RFDF	CMDFFF
W	W1C			W1C	W1C		W1C		W1C		W1C		W1C	W1C		W1C
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCTR				TXNXTPTR				RXCTR				POPNXTPTR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.6.4 Fields

Field	Function
31 TCF	Transfer Complete Flag Indicates that all bits in a frame are shifted out. This field remains set until it is cleared by writing a 1 to it. 0b - Transfer is incomplete. 1b - Transfer is complete.
30 TXRXS	TX and RX Status Reflects the run status of the module 0b - Transmit and receive operations are disabled when the module is in the Stopped state. 1b - Transmit and receive operations are enabled when the module is in the Running state.
29 —	Reserved
28 EOQF	End of Queue Flag Indicates that the last entry in a queue had been transmitted when the module was in Master mode. This field is set when the TX FIFO entry has the EOQ field set in the command halfword and the end of the transfer is reached. The field remains set until cleared by writing a 1 to it. When this field is set, the TXRXS field is automatically cleared. 0b - EOQ is not set in the executing command. 1b - EOQ is set in the executing SPI command.
27 TFUF	Transmit FIFO Underflow Flag Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for SPI blocks operating in Slave mode and SPI configuration. This field is set when the TX FIFO of the module operating in SPI Slave mode is empty and an external SPI master initiates a transfer. The field remains set until cleared by writing a 1 to it. 0b - No TX FIFO underflow occurred. 1b - TX FIFO underflow had occurred.
26	Reserved

Table continues on the next page...

Field	Function
—	
25 TFFF	Transmit FIFO Fill Flag Indicates whether there is an available location to be filled in the FIFO. Either DMA or an interrupt can be used to add another entry to the FIFO. Note that this field is set if at least one location is free in the FIFO. The field can be cleared by writing a 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request, when the TX FIFO is full. If FIFO is filled manually, and not by DMA, see Transmit FIFO fill interrupt or DMA request **NOTE:** The reset value of this field is 0 when the module is disabled(MCR[MDIS]=1). 0b - TX FIFO is full. 1b - TX FIFO is not full.
24 BSYF	Busy Flag Indicates that the current command frame is being used for transmitting multiple data frames. This field is valid only when the value of MCR[XSPI] is 1, and is not set for the last data frame of a cyclic command transfer or when the value of CTARE[DTCP] is 1. See CMD FIFO buffering mechanism for details. 0b - No cyclic command transfer is in progress. 1b - Cyclic command transfer is in progress. Current data frame is not the last data frame for on-going cyclic command transfer.
23 CMDTCF	Command Transfer Complete Flag Indicates that the last data frame for the current cyclic command has been transmitted. Therefore, this field is set only for the last data frame of a cyclic command transfer or when the value of CTARE[DTCP] is 1. The field remains set until it is cleared by writing a 1 to it. 0b - Data transfer by current command is incomplete. 1b - Data transfer by current command is complete.
22 —	Reserved
21 SPEF	SPI Parity Error Flag Indicates that SPI frame with parity error had been received. The field remains set until it is cleared by writing a 1 to it. 0b - No parity error occurred. 1b - Parity error had occurred.
20 —	Reserved
19 RFOF	Receive FIFO Overflow Flag Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and Shift register are full and a transfer is initiated. The field remains set until it is cleared by writing a 1 to it. 0b - No RX FIFO overflow occurred. 1b - RX FIFO overflow has occurred.
18 TFIWF	Transmit FIFO Invalid Write Flag Indicates data write on TX FIFO when CMD FIFO is empty. Without a command, the data entries present in TX FIFO are invalid. This field remains set until it is cleared by writing a 1 to it. 0b - No invalid data is present in TX FIFO. 1b - Invalid data is present in TX FIFO because CMD FIFO is empty.
17 RFDF	Receive FIFO Drain Flag Provides a method for the module to request that entries be removed from the RX FIFO. Note that this bit is set if at least one location can be read from the FIFO. The RFDF bit can be cleared by acknowledgement from the DMA controller when the RX FIFO is empty. 0b - RX FIFO is empty. 1b - This bit auto-clears on every RXFR read performed.

Table continues on the next page...

Field	Function
16 CMDFFF	Command FIFO Fill Flag Indicates whether there is an available location to be filled in the FIFO. Either a DMA request or an interrupt indication can be used to add another entry to the FIFO. Note that this field is set if at least one location is free in the FIFO. The CMDFFF is cleared by writing a 1 to it or by acknowledgement from the DMA controller to the CMD FIFO full request. 0b - CMD FIFO is full. 1b - CMD FIFO is not full.
15-12 TXCTR	TX FIFO Counter Indicates the number of valid entries in the TX FIFO. This field increments every time the PUSHR is written and decrements every time SPI command is executed and the SPI data is transferred to the Shift register.
11-8 TXNXTPTR	Transmit Next Pointer Indicates which TX FIFO entry is transmitted during the next transfer. This field updates every time SPI data is transferred from the TX FIFO to the Shift register. **NOTE:** If TX FIFO is cleared by writing 1 to MCR[CLR_TXF], the value of the TXNXTPTR field does not change immediately to 0. Only when the next transfer starts, the field reflects the latest value, TXNXTPTR = 1.
7-4 RXCTR	RX FIFO Counter Indicates the number of entries in the RX FIFO. This field decrements every time the POPR is read and increments every time data is transferred from the Shift register to the RX FIFO.
3-0 POPNXTPTR	Pop Next Pointer Contains a pointer to the RX FIFO entry to be returned when the POPR is read. This field is updated when the POPR is read.

24.3.7 DMA/Interrupt Request Select and Enable Register (RSER)

24.3.7.1 Offset

Register	Offset
RSER	30h

24.3.7.2 Function

This register controls DMA and interrupt requests. You must not write to the register when the module is in the Running state.

24.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCF_RE	CMDFFF_RE	Reserved	EOQF_RE	TFUF_RE	Reserved	TFFF_RE	TFFF_DIRS	CMDTCF_RE	Reserved	SPEF_RE	Reserved	RFOF_RE	TFIWF_RE	RFDF_RE	RFDF_DIRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDFFF_DIRS	Reserved	0													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.7.4 Fields

Field	Function
31 TCF_RE	Transmission Complete Request Enable Enables SR[TCF] to generate an interrupt request. 0b - TCF interrupt requests are disabled. 1b - TCF interrupt requests are enabled.
30 CMDFFF_RE	Command FIFO Fill Flag Request Enable Enables SR[CMDFFF] to generate a request. The CMDFFF_DIRS field of this register selects between generating an interrupt request or a DMA request. 0b - CMDFFF interrupts or DMA requests are disabled. 1b - CMDFFF interrupts or DMA requests are enabled.
29 —	Always write the reset value to this field.
28 EOQF_RE	Finished Request Enable Enables SR[EOQF] to generate an interrupt request 0b - EOQF interrupt requests are disabled. 1b - EOQF interrupt requests are enabled.
27 TFUF_RE	Transmit FIFO Underflow Request Enable Enables SR[TFUF] to generate an interrupt request. 0b - TFUF interrupt requests are disabled. 1b - TFUF interrupt requests are enabled.
26 —	Always write the reset value to this field.
25	Transmit FIFO Fill Request Enable

Table continues on the next page...

Field	Function
TFFF_RE	Enables SR[TFFF] to generate a request. The TFFF_DIRS field of this register selects between generating an interrupt request or a DMA request. 0b - TFFF interrupts or DMA requests are disabled. 1b - TFFF interrupts or DMA requests are enabled.
24 TFFF_DIRS	Transmit FIFO Fill DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request. 0b - TFFF flag generates interrupt requests. 1b - TFFF flag generates DMA requests.
23 CMDTCF_RE	Command Transmission Complete Request Enable Enables SR[CMDTCF] to generate an interrupt request 0b - CMDTCF interrupt requests are disabled. 1b - CMDTCF interrupt requests are enabled.
22 —	Always write the reset value to this field.
21 SPEF_RE	SPI Parity Error Request Enable Enables SR[SPEF] to generate an interrupt request 0b - SPEF interrupt requests are disabled. 1b - SPEF interrupt requests are enabled.
20 —	Always write the reset value to this field.
19 RFOF_RE	Receive FIFO Overflow Request Enable Enables SR[RFOF] to generate an interrupt request 0b - RFOF interrupt requests are disabled. 1b - RFOF interrupt requests are enabled.
18 TFIWF_RE	Transmit FIFO Invalid Write Request Enable Enables SR[TFIWF] to generate an interrupt request 0b - TFIWF interrupt requests are disabled. 1b - TFIWF interrupt requests are enabled.
17 RFDF_RE	Receive FIFO Drain Request Enable Enables SR[RFDF] to generate a request. The RFDF_DIRS field of this register selects between generating an interrupt request or a DMA request. 0b - RFDF interrupt or DMA requests are disabled. 1b - RFDF interrupt or DMA requests are enabled.
16 RFDF_DIRS	Receive FIFO Drain DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. If SR[RFDF] is set, and the value of the RFDF_RE field in this register is 1, the RFDF_DIRS field selects between generating an interrupt request or a DMA request. 0b - Interrupt request 1b - DMA request
15 CMDFFF_DIRS	Command FIFO Fill DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. If SR[CMDFFF] is set, and the value of the CMDFFF_RE field in this register is 1, the CMDFFF_DIRS field selects between generating an interrupt request or a DMA request. 0b - CMDFFF flag generates interrupt requests. 1b - CMDFFF flag generates DMA requests.
14	Always write the reset value to this field.

Table continues on the next page...

Field	Function
—	
13-0 —	Reserved

24.3.8 PUSH TX FIFO Register In Master Mode (PUSHR)

24.3.8.1 Offset

Register	Offset
PUSHR	34h

24.3.8.2 Function

Specifies data to be transferred to the TX FIFO and CMD FIFO. You must write 16 bits of data into the TXDATA field. An 8- or 16-bit write access to the TXDATA field transfers 16 bits of data bus to the TX FIFO. A write access to the command fields transfers the 16 bits of command information to the CMD FIFO. In Master mode, the register transfers 16 bits of data to the TX FIFO and 16 bits of command information to the CMD FIFO.

If Extended SPI Mode is disabled (MCR[XSPI]), the TX FIFO and CMD FIFO must be filled simultaneously. In other words, you must perform write accesses to both the data and command fields for every PUSHR operation. With Extended SPI Mode disabled and both the TX FIFO and CMD FIFO written to and read from simultaneously, they behave as a single 32-bit FIFO. When Extended SPI mode is enabled (MCR[XSPI]), the TX FIFO and CMD FIFO can be written to independently.

A read access of PUSHR returns the top-most TX FIFO and CMD FIFO entries concatenated.

When the module is disabled, writing to this register does not update the FIFO. Therefore, any reads performed while the module is disabled return the last PUSHR write performed while the module was still enabled.

24.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CONT	CTAS			EOQ	CTCNT	PE_MASC	PP_MCSC	Reserved				PCS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.8.4 Fields

Field	Function
31 CONT	Continuous Peripheral Chip Select Enable Selects a continuous selection format. This field is used in SPI Master mode. The field enables the selected PCS signals to remain asserted between transfers. 0b - Return PCSn signals to their inactive state between transfers. 1b - Keep PCSn signals asserted between transfers.
30-28 CTAS	Clock and Transfer Attributes Select Selects which CTAR to use in Master mode to specify the transfer attributes for the associated SPI frame. In SPI Slave mode, CTAR0 is used. See the chip-specific section for details to determine how many CTARs this chip has. You should not program a value in this field for a register that is not present. 000b - CTAR0 001b - CTAR1 010b - Reserved 011b - Reserved 100b - Reserved 101b - Reserved 110b - Reserved 111b - Reserved
27 EOQ	End of Queue The host software uses this field to signal to the module that the current SPI transfer is the last in a queue. At the end of the transfer, SR[EOQF] is set. 0b - SPI data is not the last data to transfer. 1b - SPI data is the last data to transfer.
26 CTCNT	Clear Transfer Counter Writes a 0 to TCR[TCNT], and the TCNT field is cleared before the module starts transmitting the current SPI frame. 0b - Do not clear TCR[TCNT] 1b - Clear TCR[TCNT]
25	Parity Enable or Mask TASC Delay in Current Frame

Table continues on the next page...

Field	Function
PE_MASC	PE – This bit enables parity bit transmission and parity reception check for the SPI frame. MASC - The current frame has the “after SCK” delay masked if this bit is asserted. See Fast Continuous Selection Format for more details. **NOTE:** This field is used as Mask T_{ASC} in the Fast Continuous PCS mode when MCR[FCPCS] is set. 0b - PE - No parity bit is included/checked. MASC - T_{ASC} delay is not masked and the current frame has the "After SCK delay". 1b - PE - Parity bit is transmitted instead of the last data bit in the frame; parity is checked for the received frame. MASC - T_{ASC} delay is masked in the current frame.
24 PP_MCSC	Parity Polarity or Mask TCSC Delay in Next Frame PP: It controls the polarity of the parity bit transmitted and checked. MCSC: The next frame has the “PCS to SCK” delay masked if this bit is asserted. See Fast Continuous Selection Format for details. **NOTE:** This field is used as Mask T_{CSC} in the Fast Continuous PCS mode when MCR[FCPCS] is set. 0b - PP - Even Parity: the number of 1 bits in the transmitted frame is even. SR[SPEF] is set if the number of 1 bits is odd in the received frame. MCSC - T_{CSC} delay is not masked and the next frame has the PCS to SCK delay. 1b - PP - Odd Parity: the number of 1 bits in the transmitted frame is odd. SR[SPEF] is set if the number of 1 bits is even in the received frame. MCSC - T_{ASC} delay is masked in the next frame.
23-20 —	Always write the reset value to this field.
19-16 PCS	PCS Selects which PCS signals are to be asserted for the transfer. 0000b - Negate the PCS[x] signal. 0001b - Assert the PCS[x] signal.
15-0 TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command

24.3.9 PUSH TX FIFO Register In Slave Mode (PUSHR_SLAVE)

24.3.9.1 Offset

Register	Offset
PUSHR_SLAVE	34h

24.3.9.2 Function

Specifies the data to be transferred to the TX FIFO in Slave mode. An 8- or 16-bit write access to PUSHR transfers the 16-bit TXDATA field to the TX FIFO.

If the value of MCR[XSPI] is 1, up to 32-bit DSPI frames may be queued for transmission or reception.

24.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.9.4 Fields

Field	Function
31-16 —	Reserved
15-0 TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command

24.3.10 POP RX FIFO Register (POPR)

24.3.10.1 Offset

Register	Offset
POPR	38h

24.3.10.2 Function

This register is used to read the RX FIFO. An 8- or 16-bit access to the register has the same effect on the RX FIFO as 32-bit read accesses. A write to this register generates a transfer error.

24.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.10.4 Fields

Field	Function
31-0 RXDATA	Received Data Contains the SPI data from the RX FIFO entry to which the pop next data pointer points

24.3.11 Transmit FIFO Registers (TXFR0 - TXFR3)

24.3.11.1 Offset

Register	Offset
TXFR0	3Ch
TXFR1	40h
TXFR2	44h
TXFR3	48h

24.3.11.2 Function

These registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers can only be read and not modified. Reading the TXFR*x* registers does not alter the state of the TX FIFO. Also, reading these registers is invalid if the module is operating in the Extended SPI mode.

24.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TXCMD_TXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.11.4 Fields

Field	Function
31-16 TXCMD_TXDATA	Transmit Command or Transmit Data In Master mode, the TXCMD field contains the command that sets the transfer attributes for the SPI data. In Slave mode, this field is reserved.
15-0 TXDATA	Transmit Data Contains the SPI data to be shifted out.

24.3.12 Receive FIFO Registers (RXFR0 - RXFR3)

24.3.12.1 Offset

Register	Offset
RXFR0	7Ch
RXFR1	80h

Table continues on the next page...

Register	Offset
RXFR2	84h
RXFR3	88h

24.3.12.2 Function

These read-only registers provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. Reading the RXFR*x* registers does not alter the state of the RX FIFO. However, the value of MCR[MDIS] must be 0 when RXFR is read.

24.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.12.4 Fields

Field	Function
31-0 RXDATA	Receive Data Contains the received SPI data

24.3.13 Clock and Transfer Attributes Register Extended (CTARE0 - CTARE1)

24.3.13.1 Offset

Register	Offset
CTARE0	11Ch
CTARE1	120h

24.3.13.2 Function

CTARE registers are used to define the extended transfer attributes for an SPI frame. These registers are valid only when Extended SPI mode is enabled (MCR[XSPI]).

When the module is configured as a SPI master, the CTAS field in CMD FIFO entry selects which of the CTARE registers is used.

24.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															FMSZE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DTCP										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

24.3.13.4 Fields

Field	Function
31-17 —	Reserved
16 FMSZE	Frame Size Extended This field is valid only when MCR[XSPI] is set. This field concatenated with CTAR[FMSZ] defines the Frame size of the SPI frames to be transmitted. Effective frame size would be the concatenation of {CTARE[FMSZE], CTAR[FMSZ]} plus 1. 0b - Default Mode. Up to 16 bit SPI frames can be transfered.

Table continues on the next page...

Field	Function
	1b - Up to 32 bit SPI frames can be transferred. Each Frame transfer will be a result of 2 simultaneous TX FIFO Pop Operation.
15-11 —	Reserved
10-0 DTCP	Data Transfer Count Preload This field is valid only when SPIx_MCR[XSPI] is set. This field defines the number of data frames (whose size is defined by CTARE[FMSZE] and CTAR[FMSZ]) to be transmitted using the Command frame which selected this SPIx_CTARE register. The value 0 is reserved and should not be written in this field. The default value of this field is 1.

24.3.14 Status Register Extended (SREX)

24.3.14.1 Offset

Register	Offset
SREX	13Ch

24.3.14.2 Function

The register contains status fields. The fields reflect the status of the module and indicate the occurrence of events. This register is not writeable.

24.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TXCTR4	0		RXCTR4	0		CMDCTR					CMDNXTPTR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.3.14.4 Fields

Field	Function
31-15 —	Reserved
14 TXCTR4	TX FIFO Counter[4] This bit is an extension of SR[TXCTR]. The concatenated field {TXCTR4, TXCTR} indicates the number of valid entries in the TX FIFO. This field is incremented every time the PUSHR is written. And this field is decremented every time an SPI command is executed and the SPI data is transferred to the Shift register.
13-12 —	Reserved
11 RXCTR4	RX FIFO Counter[4] This bit is an extension of SR[RXCTR]. The concatenated field {RXCTR4, RXCTR} indicates the number of entries in the RX FIFO. This field is decremented every time the POPR is read. And this field is incremented every time data is transferred from the Shift register to the RX FIFO.
10-9 —	Reserved
8-4 CMDCTR	CMD FIFO Counter Indicates the number of entries in the CMD FIFO. The CMDCTR is incremented every time the command part of PUSHR is written. The CMDCTR is decremented every time a SPI command is executed (all data frames due to current command frame have been transmitted).
3-0 CMDNXTPTR	Command Next Pointer Indicates which CMD FIFO Entry is used during the next transfer. The CMDNXTPTR field is updated every time SPI data due to current command have been transmitted.

24.4 Functional description

The module supports full-duplex, synchronous serial communications between chips and peripheral devices. The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes.

The module includes the following configuration:

- SPI configuration in which the module operates as a basic SPI or a queued SPI

MCR[DCONF] determines the module configuration. SPI configuration is selected when the value of MCR[DCONF] is 0.

The CTAR*n* registers hold clock and transfer attributes. The SPI configuration allows to select which CTAR to use, on a frame-by-frame basis, by setting a field in the SPI command. MCR[XSPI] further allows the usage of CTARE*n* registers. This allows you to send multiple data frames using a single command frame.

Typical master-to-slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted to a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was included in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, if the value of SR[TCF] is 1, it indicates a completed frame transfer.

Figure 24-3. Serial protocol overview

Generally, more than one slave device can be connected to the module master, and 4 PCS signals of the module masters can be used to select which of the slaves to communicate with.

The SPI configuration shares transfer protocol and timing properties that are described independently of the configuration in Transfer formats. The transfer rate and delay settings are described in Module baud rate and clock delay generation.

24.4.1 Starting and stopping module transfers

The module has two operating states: Stopped and Running. Both the states are independent of their configuration. The default state of the module is Stopped. In this state, the module initiates no serial transfers in the Master mode and responds to no transfers in the Slave mode. The Stopped state is also a safe state for writing to the various configuration registers of the module, without causing undetermined results. In the Running state, serial transfers take place.

SR[TXRXS] indicates the state of the module. The value of this field is 1 if the module is in the Running state.

The module starts or transitions to the Running state when all of the following conditions are true:

- The value of SR[EOQF] = 0.
- The chip is not in the Debug mode or the value of MCR[FRZ] = 0.
- The value of MCR[HALT] = 0.

The module stops or transitions from the Running state to the Stopped state after the current frame, when any one of the following conditions exist:

- The value of SR[EOQF] = 1.
- The chip is in Debug mode and the value of MCR[FRZ] = 1.
- The value of MCR[HALT] = 1.

State transitions from running to stopped occur on the next frame boundary if a transfer is in progress, and occur immediately if no transfers are in progress.

24.4.2 SPI configuration

The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The module is in SPI configuration when the value of MCR[DCONF] is 0. The SPI frames can be 32-bit long. The host CPU or a DMA controller transfers the external SPI data to the module RAM queues and to a TX FIFO buffer. The received data is stored in the entries of the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to a memory external to the module. The operation of FIFO buffers is described in the following sections:

- TX FIFO buffering mechanism
- CMD FIFO buffering mechanism
- RX FIFO buffering mechanism

The interrupt and DMA request conditions are described in Interrupts/DMA requests.

The SPI configuration supports two module-specific modes—Master mode and Slave mode.In the Master mode, the module initiates and controls the transfer according to the fields of the executing SPI command. In the Slave mode, the module responds only to transfers initiated by a bus master external to it, and the SPI command field is reserved.

24.4.2.1 Master mode

In the SPI Master mode, the module initiates serial transfers by controlling the SCK and the PCS signals. The executing SPI command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert. The command field also contains various bits that help with queue management and transfer protocol. See PUSH TX FIFO Register In Master Mode (PUSHR) for details on the SPI command fields. The data in the executing TX FIFO entry is loaded into the shift register and shifted out on the SOUT pin. In the SPI Master mode, each SPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis. In the Extended SPI Master mode, multiple SPI frames can have a single command associated with them allowing efficient SPI frame transfers requiring common transfer attributes. In addition, the Extended SPI mode allows larger frame sizes of up to 32 bits.

24.4.2.2 Slave mode

In the SPI Slave mode, the module responds to transfers that an SPI bus master initiates. It does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be configured for successful communication with SPI master. The SPI Slave mode transfer attributes are configured in the CTAR0 and CTARE0 registers. The data shifts out with MSB first, and the module does not support shifting out of LSB.

24.4.2.3 FIFO disable operation

The FIFO disable mechanisms allow SPI transfers without using the TX FIFO, CMD FIFO, or RX FIFO. The module operates as a double-buffered, simplified SPI when the FIFOs are disabled. The transmit and receive sides of the FIFOs are disabled separately. Writing a 1 to MCR[DIS_TXF] disables the TX FIFO and CMD FIFO, and writing a 1 to MCR[DIS_RXF] disables the RX FIFO.

The FIFO disable mechanisms are transparent to you and to the host software. Transmit data and commands are written to the PUSHR and received data is read from the POPR.

When the TX FIFO and CMD FIFO are disabled:

- SR[TFFF], SR[TFUF], SR[CMDFFF], SREX[CMDCTR], and SR[TXCTR] behave as if there is a one-entry FIFO.
- The contents of TXFRs, SR[TXNXTPTR], and SREX[CMDNXTPTR] are undefined.

Similarly, when the RX FIFO is disabled, SR[RFDF], SR[RFOF], and SR[RXCTR] behave as if there is a one-entry FIFO, but the contents of the RXFR registers and SR[POPNXTPTR] are undefined.

24.4.2.4 TX FIFO buffering mechanism

The TX FIFO functions as a buffer of SPI data for transmission. It holds 4 words, each consisting of SPI data. The number of entries in the TX FIFO is chip-specific. SPI data is added to the TX FIFO by writing to the data field of PUSHR. TX FIFO entries can only be removed from the TX FIFO if they are shifted out or by flushing the TX FIFO.

SR[TXCTR] indicates the number of valid entries in the TX FIFO. The field updates every time an 8- or 16-bit write takes place on PUSHR[TXDATA] or SPI data is transferred into the Shift register from the TX FIFO.

SR[TXNXTPTR] indicates the TX FIFO entry that will be transmitted during the next transfer. The TXFR*n* registers are invalid in the Extended SPI mode because the TX FIFO and CMD FIFO can be used independently. The TXNXTPTR field increments every time SPI data is transferred from the TX FIFO to the Shift register. The maximum value of this field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

24.4.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO and CMD FIFO by writing to the PUSHR. When the TX FIFO is not full, the value of DR[TFFF] is set to 1, and the value is set to 0 when TX FIFO is full and the DMA controller indicates that a write to PUSHR is complete. You can also write a 1 to clear the field, which can generate a DMA request or an interrupt request. See Transmit FIFO fill interrupt or DMA request for details.

The module ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition occurs.

24.4.2.4.2 Draining the TX FIFO

The module removes (drains) the TX FIFO entries by shifting SPI data out through the Shift register. It transfers entries from the TX FIFO to the Shift register, and the entries are shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the Shift register, the TX FIFO counter decrements by 1. When the value of MCR[XSPI] is 1, and if the frame size of SPI data to be transmitted is more than 16 bits, then it causes two data entries to be popped from TX FIFO

simultaneously. These entries are transferred to the Shift register. The first of the two popped entries forms the 16 least significant bits of the SPI frame to be transmitted. Such an operation also causes the TX FIFO counter to decrement by 2. At the end of a transfer, the value of SR[TCF] is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a 1 to MCR[CLR_TXF].

If an external bus master initiates a transfer with a module slave while the slave's TX FIFO is empty, the value of SR[TFUF] is configured as 1. See Transmit FIFO underflow interrupt request for details.

24.4.2.5 CMD FIFO buffering mechanism

The CMD FIFO functions as a buffer of SPI command used for SPI data transmission. When Extended SPI Mode (MCR[XSPI]) is disabled, the TX FIFO and CMD FIFO must be filled together, i.e. write enables should be given for both the Data and Command fields while performing a PUSHR operation.When Extended SPI Mode (MCR[XSPI]) is enabled, the TX FIFO and CMD FIFO can be filled independently.

The CMD FIFO holds 4 words, each representing SPI command fields. The number of entries in the CMD FIFO is chip-specific. SPI command is added to the CMD FIFO by writing to the command field of PUSHR. CMD FIFO entries can only be removed from the CMD FIFO by being shifted out (to help transmit SPI data) or by flushing the CMD FIFO.

When Extended SPI Mode (MCR[XSPI]) is disabled, every CMD FIFO entry has a corresponding single TX FIFO entry attached to it because both these FIFO's are filled simultaneously.

When Extended SPI Mode (MCR[XSPI]) is enabled, every CMD FIFO entry can have multiple TX FIFO entries attached to it. Thus a single CMD FIFO entry can be used to transmit multiple TX FIFO entries. The CTARE[DTCP] field decides the number of SPI Data Frames having frame size as {FMSZE, FMSZ} to be transmitted using the current Command Entry. The CTAR/CTARE registers pointed by the CTAS field in the Command frame gives the FMSZ and FMSZE fields respectively. The time for which a command entry is in use is known as a Command Cycle. The Busy Flag SR[BSYF] is asserted for the duration of the Command Cycle except for the last SPI frame in the Command Cycle.

The CMD FIFO Counter field (CMDCTR) in the SPI Status Register extended(SREX) indicates the number of valid entries in the CMD FIFO. The CMDCTR field is updated every time a 8- or 16-bit write takes place on the lower half of SPI_PUSHR or SPI data is transferred into the shift register from the TX FIFO.

The TXFRn Registers are invalid in the Extended SPI Mode, since the TX FIFO and CMD FIFO can be used independently. The CMDNXTPTR field indicates which CMD FIFO Entry will be used during the next command cycle. The CMDNXTPTR field is incremented every time the last SPI data in the command cycle is transferred from the TX FIFO to the shift register and it rolls over after reaching the maximum.

24.4.2.6 RX FIFO buffering mechanism

The RX FIFO functions as a buffer for data received on the SIN pin. The FIFO holds 4 received SPI data frames. The number of entries in the RX FIFO is chip-specific. SPI data is added to the RX FIFO at the completion of a transfer when the received data in the Shift register is transferred into the RX FIFO. SPI data is removed (popped) from the RX FIFO by reading the POPR. RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

SR[RXCTR] indicates the number of valid entries in the RX FIFO. The field is updated every time the POPR is read or SPI data is copied from the Shift register to the RX FIFO.

SR[POPNXTPTR] points to the RX FIFO entry that is returned when the POPR is read. The field contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR = 2 means that the RXFR2 contains the received SPI data that is returned after the POPR is read. The POPNXTPTR field increments every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

24.4.2.6.1 Filling the RX FIFO

The RX FIFO is filled with the received SPI data from the Shift register. While the RX FIFO is not full, SPI frames from the Shift register are transferred to the RX FIFO. Every time a SPI frame is transferred to the RX FIFO, the RX FIFO counter is incremented by 1.

If the RX FIFO and Shift register are full and a transfer is initiated, the value of SR[RFOF] equals 1, indicating an overflow condition. Depending on the state of MCR[ROOE], the data from the transfer that generated the overflow is either ignored or shifted to the Shift register. If the value of the ROOE field is 1, the incoming data is shifted in to the Shift register. If the value of ROOE is 0, the incoming data is ignored.

24.4.2.6.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the POPR. A read of the POPR decrements the RX FIFO counter by 1. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the value of SR[RFDF] is 1. The value of the RFDF field turns 0 when the RX FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

24.4.3 Module baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the system clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

Figure 24-4. Communications clock prescalers and scalers

24.4.3.1 Baud rate generator

The baud rate is the frequency of the SCK. The protocol clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

Table 24-5. Baud rate computation example

f_P	PBR	Prescaler	BR	Scaler	DBR	Baud rate
100 MHz	0b00	2	0b0000	2	0	25 Mb/s
20 MHz	0b00	2	0b0000	2	1	10 Mb/s

NOTE

The clock frequencies mentioned in the preceding table are provided as an example. See the Clocking chapter for the frequency used to drive this module in the chip.

24.4.3.2 PCS to SCK delay (t_{CSC})

The "PCS to SCK delay" is the duration from assertion of the PCS signal to the first SCK edge. See Figure 24-5 for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR*x* registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

Table 24-6. PCS to SCK delay computation example

f_{SYS}	PCSSCK	Prescaler	CSSCK	Scaler	PCS to SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 µs

NOTE

The clock frequency mentioned in the preceding table is given as an example. See the Clocking chapter for the frequency used to drive this module in the chip.

24.4.3.3 After SCK delay (t_{ASC})

The "After SCK Delay" is the duration between the last edge of SCK and the negation of PCS. See Figure 24-5 and Figure 24-6 for illustrations of this type of delay. The PASC and ASC fields in the CTAR*x* registers select the "After SCK Delay" by the formula in the ASC field description. The following table shows an example of how to compute the "After SCK Delay".

Table 24-7. After SCK Delay computation example

f_P	PASC	Prescaler	ASC	Scaler	After SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 µs

NOTE

The clock frequency mentioned in the preceding table is given as an example. See the Clocking chapter for the frequency used to drive this module in the chip.

24.4.3.4 Delay after transfer (t_{DT})

The "Delay after Transfer" is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See Figure 24-5 for an illustration of this type of delay. The PDT and DT fields in the CTAR*x* registers select the "Delay after Transfer" by the formula in the DT field description. The following table shows an example of how to compute the "Delay after Transfer".

Table 24-8. "Delay after Transfer" computation example

f_P	PDT	Prescaler	DT	Scaler	Delay after Transfer
100 MHz	0b01	3	0b1110	32768	0.98 ms

NOTE

The clock frequency mentioned in the preceding table is given as an example. See the Clocking chapter for the frequency used to drive this module in the chip.

When in Non-Continuous Clock mode, the t_{DT} delay is configured according to the equation specified in the CTAR[DT] field description. When in Continuous Clock mode, the delay is fixed at 1 SCK period.

24.4.4 Transfer formats

The SPI serial communication is controlled by the SCK signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, CTAR*n*[CPOL] and CTAR*n*[CPHA] select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even if the bus slave does not control the SCK signal, in Slave mode, the values of CPOL and CPHA must be identical to the master device settings to ensure proper transmission. In SPI Slave mode, only CTAR0 is used.

The module supports two different transfer formats:

- Classic SPI with CPHA = 0
- Classic SPI with CPHA = 1

In the interface configurations, the module provides the option of having the PCS signals asserted between frames. See Continuous selection format for details.

24.4.4.1 Classic SPI transfer format (CPHA = 0)

The transfer format shown in the following figure is used to communicate with peripheral SPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

Figure 24-5. Module transfer timing diagram (CPHA = 0, FMSZ = 8)

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate PCS signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the t_{CSC} delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and slave devices place their second data bit on their serial data output signals. For the rest of the frame, the master and the slave sample their SIN pins on the odd-numbered clock edges and change the data on their SOUT pins for the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signals. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

24.4.4.2 Classic SPI transfer format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral SPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins for the even-numbered SCK edges.

Figure 24-6. Module transfer timing diagram (CPHA = 1, FMSZ = 8)

The master initiates the transfer by asserting the PCS signal to the slave. After the t_{CSC} delay elapses, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin. The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK, the master and slave sample their SIN pins. For the rest of the frame, the master and the slave change the data for their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signal. A delay of t_{DT} is inserted before the master can initiate a new frame transfer.

24.4.4.3 Continuous selection format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The continuous selection format provides the flexibility to handle the following case. The format is enabled for the SPI configuration by setting the CONT bit in the SPI command.

When the value of CONT is 0, the module drives the asserted Chip Select signals to their idle states in between frames. The idle states of these signals are selected by the PCSIS*n*n fields in the MCR. The following figure shows the timing diagram for two 4-bit transfers with CPHA = 1 and CONT = 1.

Figure 24-7. Example of non-continuous format (CPHA=1, CONT=0)

When CONT = 1, the PCS signal remains asserted for the duration of the two transfers. The "Delay between Transfers" (t_{DT}) is not inserted between the transfers. The following figure shows the timing diagram for two 4-bit transfers with CPHA = 1 and CONT = 1.

Figure 24-8. Example of continuous transfer (CPHA=1, CONT=1)

When using the module with continuous selection, follow these rules:

- All transmit commands must have the same PCS*n* field programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only the FMSZ field can be programmed differently in these CTARs.
- When transmitting multiple frames in this mode, the software must ensure that the last frame has USHR[CONT] deasserted in Master mode and the software must provide sufficient frames in the TX_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- PUSHR[CONT] must be deasserted before asserting MCR[HALT] in master mode. This ensures that the PCSn signals are deasserted. Asserting MCR[HALT] during continuous transfer causes the PCSn signals to remain asserted and hence the slave device cannot transition from Running to Stopped state.

NOTE

You must fill the TX FIFO with the number of entries that are concatenated together under one PCS assertion for both master and slave before the TX FIFO is empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

24.4.4.4 Fast continuous selection format

The fast continuous selection format has a functioning similar to the continuous selection format except that the inter command delays, t_{ASC} and t_{CSC}, can be masked out and are not inserted by the hardware.

NOTE

The fast continuous selection format is available in the SPI configuration only when the Continuous SCK mode is disabled. Masking of delays is not allowed if the transfer is non-continuous.

The fast continuous selection format is enabled by writing a 1 to MCR[FCPCS]. When this field is asserted, the MASC and MCSC fields of the PUSHR perform the function of mask bits for the transmit frame. These fields individually mask the t_{ASC} and t_{CSC} delays as programmed by the software. A normal continuous selection format includes these two delays for each frame, which is transmitted with the CONT field asserted. To avoid these delays and to speed up the transfer process, the software can mask these delays while programming the command in the PUSHR.

While masking the delays, the software must follow these masking rules, otherwise, correct operation is not guaranteed.

- The MASC field masks the “After SCK” delay for the current frame.
- The MCSC field masks the “PCS to SCK” delay for the next frame.
- The t_{ASC} delay must not be masked when the current frame is the last frame in the continuous selection format.
- The "PCS to SCK" delay for the first frame in the continuous selection format cannot be masked.
- Masking of only t_{ASC} is not allowed. If it is masked, then t_{CSC} must be masked too.
- Masking of both t_{ASC} and t_{CSC} delays is allowed. In this case, the delay between two frames is equal to half the baud rate (baud rate is 1 SCK cycle duration) configured by the software.
- Masking of only t_{CSC} is allowed. In this case, the delay between two frames is equal to the t_{ASC} time and thus the software must ensure that the t_{ASC} time is greater than the baud rate.

- The software must not mask these delays if the continuous selection format is not used and MCR[FCPCS] is asserted.
- Rules applicable to the continuous selection format are applicable here too.

The following figure shows the timing for a fast continuous selection format transfer. Here, seven frames are transferred with both t_{ASC} and t_{CSC} delays masked except for the last frame that terminated the transfer. The last frame has t_{ASC} delay at its end.

Figure 24-9. Example of fast continuous selection format

In case any chip select is to be changed, first the fast continuous selection format should be terminated and appropriate delays must be introduced.

24.4.5 Continuous SCK

The module provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by writing a 1 to MCR[CONT_SCKE]. Enabling this field generates the Continuous SCK only if MCR[HALT] is low. Continuous SCK is valid for all configurations.

Continuous SCK is only supported for CPHA = 1. Clearing CPHA is ignored if the value of the CONT_SCKE field is 1.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the module is in SPI configuration, CTAR0 is used initially. At the start of each SPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using continuous SCK. Switching clock polarity between frames while using continuous SCK can cause errors in the transfer. The Continuous SCK operation is not guaranteed if the module is put into the External Stop mode or Module Disable mode.

Enabling continuous SCK disables the PCS to SCK delay, and the Delay after Transfer (t_{DT}) is fixed to one SCK cycle. The following figure is the timing diagram for the Continuous SCK format with continuous selection disabled.

NOTE

In the Continuous SCK mode, for the SPI transfer, CTAR0 should always be used and the TX FIFO must be cleared through MCR[CLR_TXF] before initiating a transfer.

Figure 24-10. Continuous SCK timing diagram (CONT = 0)

If the CONT bit in the TX FIFO entry is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, which means when SOUT is pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO

- Continuous SCK with CONT bit set and entering the Stopped state (see Starting and stopping module transfers)
- Continuous SCK with CONT bit set and entering the Stop mode or Module Disable mode

The following figure shows the timing diagram for the Continuous SCK format with continuous selection enabled.

Figure 24-11. Continuous SCK timing diagram (CONT = 1)

24.4.6 Slave mode operation constraints

The slave mode logic shift register is buffered. This allows the data streaming operation when the module is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at the second SCK clock edge of each frame to the Shift register if the SS_b signal is asserted, and at and any time when transmit data is ready and the SS_b signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR.

If the SS_b signal negates before that last SCK edge, the data from the Shift register is lost.

24.4.7 Parity generation and check

The module can generate and check parity in the serial frame. The parity bit replaces the last transmitted bit in the frame. The parity is calculated for all transmitted data bits in frame, not including the last data bit that would be transmitted. The parity generation and control are done on a frame basis. The registers field setting frame size defines the total number of bits in the frame, including the parity bit. Thus, to transmit or receive the same number of data bits with parity check, increase the frame size by one versus the same data size frame without the parity check.

Parity can be selected as odd or even. Parity errors in the received frame set parity error flags in the Status register. The parity error interrupt requests are generated, if enabled. The module can be programmed to stop frame transmission in case of a frame reception with parity error.

24.4.7.1 Parity for SPI frames

When the module is in the master mode the parity generation is controlled by PE and PP bits of the CMD FIFO entries (PUSHR). Setting the PE bit enables parity generation for transmitted SPI frames and parity check for received frames. The PP bit defines polarity of the parity bit.

When continuous PCS selection is used to transmit SPI data, two parity generation scenarios are available:

- Generate or check parity for the whole frame
- Generate or check parity for each sub-frame separately

To generate or check parity for the whole frame, set PE bit only in the last command or TX FIFO entry, forming this frame with the PUSHR.

To generate or check parity for each sub-frame, set PE bit in each command or TX FIFO entry, forming this frame.

If the parity error occurs for received SPI frame, SR[SPEF] is set. If the value of MCR[PES] is 1, the module stops SPI frames transmission. To resume SPI operation, clear SR[SPEF] or MCR[PES].

In Slave mode, the parity is controlled by CTAR0[PE] and CTAR0[PP], as in the case of master mode parity generation without continuous PCS selection.

24.4.8 Interrupts/DMA requests

The module has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

Table 24-9. Interrupt and DMA request conditions

Condition	Flag	Interrupt	DMA
End of Queue (EOQ)	EOQF	Yes	-
TX FIFO Fill	TFFF	Yes	Yes
CMD FIFO Fill	CMDFFF	Yes	Yes
TX FIFO Invalid Write	TFIWF	Yes	-
Transfer Complete	TCF	Yes	-
CMD Transfer Complete	CMDTCF	Yes	-
TX FIFO Underflow	TFUF	Yes	-
RX FIFO Drain	RFDF	Yes	Yes
RX FIFO Overflow	RFOF	Yes	-
SPI Parity Error	SPEF	Yes	-

Each condition has a corresponding field in the SR and a Request Enable field in the RSER. Certain fields, as shown in the above table, generate interrupt requests or DMA requests depending on the configuration of this register.

The module also provides a global interrupt request line, which is asserted when any of individual interrupt requests lines is asserted.

24.4.8.1 EOQ interrupt request

The EOQ interrupt request indicates that the end of a transmit queue is reached. The module generates the interrupt request when the value of RSER[EOQF_RE] is 1 and the value of the EOQ field in the executing SPI command is 1.

If the value of MCR[XSPI] is 1 and the value of the EOQ field in the executing SPI command is 1, the module generates the EOQ interrupt request after the last bit of the last data frame in the command cycle is transmitted.

When the value of MCR[XSPI] is 0, the module generates the interrupt request after the last bit of the SPI frame, with the value of EOQ as 1, is transmitted.

24.4.8.2 Transmit FIFO fill interrupt or DMA request

The transmit FIFO fill request indicates that the TX FIFO is not full. This request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the value of RSER[TFFF_RE] is 1.

RSER[TFFF_DIRS] selects whether a DMA request or an interrupt request is generated.

NOTE

TFFF clears automatically when DMA is used to fill TX FIFO. You must configure the DMA to fill only one FIFO location per transfer.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until the value of TFFF is 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, the flag does not clear.

24.4.8.3 CMD FIFO fill interrupt or DMA request

The CMD FIFO fill request indicates that the CMD FIFO is not full. This request is generated when the number of entries in the CMD FIFO is less than the maximum number of possible entries, and the value of RSER[CMDFFF_RE] is 1. RSER[CMDFFF_DIRS] selects whether a DMA request or an interrupt request is generated.

The request is useful when the value of MCR[XSPI] is 1 because TX FIFO and CMD FIFO can be filled independently. If the value of MCR[XSPI] is 0, the TX FIFO fill interrupt or DMA request suffices to fill both the FIFOs, which must be filled simultaneously.

NOTE

The CMDFFF flag clears automatically when DMA is used to fill CMD FIFO.

To clear CMDFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill CMD FIFO:

1. Wait until the value of CMDFFF is 1.
2. Write data to PUSHR using CPU.

3. Clear CMDFFF by writing a 1 to its location. If CMD FIFO is not full, this flag does not clear.

24.4.8.4 Transmit FIFO invalid write interrupt request

The transmit FIFO invalid write request is valid only when the value of MCR[XSPI] is 1. This request indicates that data exists in the TX FIFO while the CMD FIFO is empty. Because no command fields are associated with the data present in TX FIFO, this data is considered invalid until a command entry is available. The transmit FIFO invalid write request is generated for this condition when the value of RSER[TFIWF_RE] is 1.

24.4.8.5 Transfer complete interrupt request

The transfer complete request indicates the end of the transfer of a serial frame. This request is generated at the end of each frame transfer when the value of RSER[TCF_RE] is 1.

24.4.8.6 Command transfer complete interrupt request

The command transfer complete request indicates the end of transfer of the last SPI frame in a command cycle. This request is generated when the value of RSER[CMDTCF_RE] is 1.

24.4.8.7 Transmit FIFO underflow interrupt request

The transmit FIFO underflow request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the module operating in the Slave mode and according to the SPI configuration. The value of SR[TFUF] is 1 when the TX FIFO of the module is empty, and a transfer is initiated from an external SPI master. If the value of SR[TFUF] is 1 while the value of RSER[TFUF_RE] is 1, an interrupt request is generated.

24.4.8.8 Receive FIFO drain interrupt or DMA request

The receive FIFO drain request indicates that the RX FIFO is not empty. This request is generated when the number of entries in the RX FIFO is not 0, and the value of RSER[RFDF_RE] is 1.

RSER[RFDF_DIRS] selects whether a DMA request or an interrupt request is generated. You must configure the DMA to drain only one FIFO location per transfer.

24.4.8.9 Receive FIFO overflow interrupt request

The receive FIFO overflow request indicates that an overflow condition in the RX FIFO has occurred. This request is generated when RX FIFO and the Shift register are full, and a transfer is initiated. The value of RSER[RFOF_RE] must be equal to 1 for the interrupt request to be generated.

Depending on the state of MCR[ROOE], the data from the transfer that generated the overflow is either ignored or shifted to the Shift register. If the value of the ROOE field is 1, the incoming data is shifted in to the Shift register. If the value of the ROOE field is 0, the incoming data is ignored.

24.4.8.10 SPI frame parity error interrupt request

The SPI Frame Parity Error Flag indicates that a SPI frame with parity error had been received. The SPEF_RE bit in the RSER must be set for the interrupt request to be generated.

24.4.9 Power saving features

This module supports the following power-saving strategies:

- External Stop mode
- Module Disable mode: clock gating of non-memory mapped logic

All power-saving features require a logic external to SPI.

24.4.9.1 Stop mode (External Stop mode)

This module supports the Stop mode protocol. When a request is made to enter the External Stop mode, the module acknowledges the request. If a serial transfer is in progress, the module waits until it reaches the frame boundary before it is ready to have its clocks shut off . While the clocks are shut off, the module's memory-mapped logic is

not accessible. This also puts the module in the Stopped state. SR[TXRXS] is cleared to indicate the Stopped state. The states of the interrupt and DMA request signals cannot be changed in the External Stop mode.

24.4.9.2 Module Disable mode

This is a block-specific mode that the module can enter to save power. The host CPU can initiate the Module Disable mode by writing a 1 to MCR[MDIS]. The hardware can also initiate this mode.

When the value of the MDIS field is 1, the module negates the Clock Enable signal at the next frame boundary. After the Clock Enable signal is negated, it is said to have entered the Module Disable Mode. This also puts the module in the Stopped state. The value of SR[TXRXS] must be 0 to indicate the Stopped state. If implemented, the Clock Enable signal can stop the clock to the non-memory mapped logic. When Clock Enable is negated, the module is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the module is in the Module Disable mode. Reading the POPR does not change the state of the RX FIFO. Similarly, writing to the PUSHR does not change the state of the TX FIFO or CMD FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Also, changes to MCR[DIS_TXF] and MCR[DIS_RXF] have no effect in the Module Disable mode. In this mode, all status bits and register flags in the module return the correct values when read, but writing to them has no effect. Also, writing to the TCR during the Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared in the Module Disable mode.

24.5 Initialization/application information

This section describes how to initialize the module.

24.5.1 How to manage queues

Queues are not a part of this module, but the module includes features to support queue management.

Queues are primarily supported in SPI configuration, and here are the steps to manage them.

1. When the module executes the last command word from a queue, the EOQ bit in the command word indicates that it is the last entry in the queue.
2. At the end of the transfer, the value of SR[EOQF] is configured as 1.
3. If the value of SR[EOQF] is 1, the field disables serial transmission and reception of data, putting the module in the Stopped state. The value 0 is written to SR[TXRXS] indicating the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO, CMD FIFO, and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA controller.
6. Ensure that all the received data in RX FIFO has been transferred to the memory receive queue by reading SR[RXCNT] or by checking SR[RFDF] after each read operation of the POPR.
7. Modify the DMA descriptor of TX and RX channels for new queues.
8. Flush TX FIFO and CMD FIFO by writing a 1 to MCR[CLR_TXF] and flushing RX FIFO by writing a 1 to MCR[CLR_RXF].
9. Clear the transfer count either by setting the CTCNT bit in the command word of the first entry in the new queue or via the CPU writing directly to TCR[SPI_TCNT].
10. Enable the DMA channel through the DMA enable request. DMA channels are assigned to modules TX FIFO, CMD FIFO, and RX FIFO. For each module, a separate DMA enable request bit is available.
11. Enable serial transmission and serial reception of data by clearing the EOQF field.

24.5.2 Switching Master and Slave modes

When changing modes in the module, follow these steps to guarantee a proper operation.

1. Halt it by writing a 1 to MCR[HALT].
2. Clear the transmit and receive FIFOs by writing a 1 to MCR[CLR_TXF] and MCR[CLR_RXF].
3. Set the appropriate mode in MCR[MSTR] and enable it by writing a 0 to MCR[HALT].

24.5.3 Initializing module in Master/Slave modes

After the appropriate mode in MCR[MSTR] is configured, the module is enabled by writing a 0 to MCR[HALT]. The module's slave must be enabled before you enable its master. This ensures that the slave is ready to be communicated with, before the master initializes communication.

24.5.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler, PBR, and the baud rate scaler, BR, in the CTARs. The values calculated assume a 100 MHz protocol frequency, and the value of the DBR field is 0..

NOTE

The clock frequency provided in this chapter is an example. See the Clocking chapter for details on the frequency used to drive this module in the chip.

Table 24-10. Baud rate values (bit/s)

		Baud rate divider prescaler values			
		2	3	5	7
Baud rate scaler values	2	25.0M	16.7M	10.0M	7.14M
	4	12.5M	8.33M	5.00M	3.57M
	6	8.33M	5.56M	3.33M	2.38M
	8	6.25M	4.17M	2.50M	1.79M
	16	3.12M	2.08M	1.25M	893k
	32	1.56M	1.04M	625k	446k
	64	781k	521k	312k	223k
	128	391k	260k	156k	112k
	256	195k	130k	78.1k	55.8k
	512	97.7k	65.1k	39.1k	27.9k
	1024	48.8k	32.6k	19.5k	14.0k
	2048	24.4k	16.3k	9.77k	6.98k
	4096	12.2k	8.14k	4.88k	3.49k
	8192	6.10k	4.07k	2.44k	1.74k
	16384	3.05k	2.04k	1.22k	872
	32768	1.53k	1.02k	610	436

24.5.5 Delay settings

The following table shows the values for the Delay after Transfer (t_{DT}) and CS to SCK Delay (T_{CSC}) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz protocol frequency.

NOTE

The clock frequency mentioned in this section is only an example. See the clocking chapter for the frequency used to drive this module in the chip.

Table 24-11. Delay values

		Delay scaler values (CTAR[PDT])			
		1	3	5	7
Delay scaler values (CTAR[DT])	2	20.0 ns	60.0 ns	100.0 ns	140.0 ns
	4	40.0 ns	120.0 ns	200.0 ns	280.0 ns
	8	80.0 ns	240.0 ns	400.0 ns	560.0 ns
	16	160.0 ns	480.0 ns	800.0 ns	1.1 µs
	32	320.0 ns	960.0 ns	1.6 µs	2.2 µs
	64	640.0 ns	1.9 µs	3.2 µs	4.5 µs
	128	1.3 µs	3.8 µs	6.4 µs	9.0 µs
	256	2.6 µs	7.7 µs	12.8 µs	17.9 µs
	512	5.1 µs	15.4 µs	25.6 µs	35.8 µs
	1024	10.2 µs	30.7 µs	51.2 µs	71.7 µs
	2048	20.5 µs	61.4 µs	102.4 µs	143.4 µs
	4096	41.0 µs	122.9 µs	204.8 µs	286.7 µs
	8192	81.9 µs	245.8 µs	409.6 µs	573.4 µs
	16384	163.8 µs	491.5 µs	819.2 µs	1.1 ms
	32768	327.7 µs	983.0 µs	1.6 ms	2.3 ms
	65536	655.4 µs	2.0 ms	3.3 ms	4.6 ms

24.5.6 Calculation of FIFO pointer addresses

Complete visibility of the FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO, the first-in pointer is Transmit Next Pointer (TXNXTPTR), and for the CMD FIFO, the first-in pointer is the Command Next Pointer (CMDNXTPTR). For the RX FIFO, the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure

illustrates the concept of first-in and last-in FIFO entries along with the FIFO counter. The TX FIFO is chosen for illustration, but the concepts carry over. See TX FIFO buffering mechanism, CMD FIFO buffering mechanism, and RX FIFO buffering mechanism for details on the FIFO operation.

Figure 24-12. TX FIFO pointers and counter

24.5.6.1 Address calculation for the first-in entry and last-in entry in the TX FIFO

Use the following equation to compute the memory address of the first-in entry in the TX FIFO:

- First-in EntryAddress = TXFIFOBase+[4xTXNXTPTR]

Use the following equation to compute the memory address of the last-in entry in the TX FIFO:

- Last-in EntryAddress = TXFIFOBase+4x[TXCTR+TXNXTPTR - 1]mod[TXFIFOdepth]

Here is an explanation of the different components of these equations:

- TX FIFO base: Base address of TX FIFO
- TXCTR: TX FIFO Counter
- TXNXTPTR: Transmit Next Pointer
- TX FIFO Depth: Transmit FIFO depth, implementation specific

24.5.6.2 Address calculation for the first-in entry and last-in entry in the CMD FIFO

Use the following equation to compute the memory address of the first-in entry in the CMD FIFO:

- First-in EntryAddress = CMDFIFOBase+[4xCMDNXTPTR]

Use the following equation to compute the memory address of the last-in entry in the CMD FIFO:

- Last-in EntryAddress = CMDFIFOBase+4x[CMDCTR+CMDNXTPTR - 1]mod[CMDFIFOdepth]

Here is an explanation of the different components of these equations:

- CMD FIFO Base: Base address of CMD FIFO
- CMDCTR: CMD FIFO Counter
- CMDNXTPTR: Command Next Pointer
- CMD FIFO Depth: Command FIFO depth, implementation specific

24.5.6.3 Address calculation for the first-in entry and last-in entry in the RX FIFO

Use the following equation to compute the memory address of the first-in entry in the RX FIFO:

- First-in EntryAddress = RXFIFOBase+[4xPOPNXTPTR]

Use the following equation to compute the memory address of the last-in entry in the RX FIFO:

- Last-in EntryAddress = RXFIFOBase+4x[RXCTR+POPNXTPTR - 1]mod[RXFIFOdepth]

Here is an explanation of the different components of these equations:

- RX FIFO Base: Base address of RX FIFO
- RXCTR: TX FIFO Counter
- POPNXTPTR: Pop Next Pointer
- RX FIFO Depth: Receive FIFO depth, implementation specific

Chapter 25
Trust Zone Modules

25.1 Overview

NXP is responsible for the overall security architecture of Layerscape SoCs. Arm TrustZone IP is listed in its own section as this IP is mainly responsible for creating and enforcing resource reservation for the trusted execution environment known as TrustZone. TrustZone is a concept users may be familiar with from other Arm based SoC families, including NXP's own iMX product line. In some cases, NXP's SoC reference manuals and LSDK documentation may cross reference Arm public documentation.

25.2 TrustZone as implemented on the chip

The chip includes the following TrustZone modules:

- TrustZone Address Space Controller (TZC-400)
- TrustZone Protection Controller (TZPC)
- TrustZone Memory Adapter (TZMA)

The following table describes the TrustZone as implemented in LS1088A:

Table 25-1. TrustZone address map

Module	Module Base address
TZC-400	0x110_0000
TZPC	0x220_0000

LS1088A includes one TZC-400 unit. TZC-400 and TZMA protect only DDR and OCRAM memory space, respectively.

The following table describes the integration of TZC-400 into this chip:

Table 25-2. Configurable parameters

Feature	TZC-400
Number of filter units	1
Number of fast path outstanding read accesses **NUM_OUTSTAND**	32
Fast Path (FPID0=1) for all reads	Yes
NS Access ID (NSAID) support[1]	No

1. NS Access ID is not supported. The NSAID[3:0] TZC-400 input is strapped to 0 and therefore does not vary with initiator or transaction.

25.3 TrustZone Address Space Controller (TZC-400)

25.3.1 Introduction

This chapter introduces the TZC-400. It contains the following sections:

- About the TZC-400.
- Features.

25.3.1.1 About the TZC-400

25.3.1.1.1 TZC-400 overview

The TZC-400 performs filtering based on security requirements specified for address regions. You can program the eight TZC-400 address regions with varying security requirements for your intended application.

The figure below shows a high-level view of the TZC-400 with a control unit and one filter unit.

Figure 25-1. TZC-400 overview

The APB® interface of the control unit enables you to program the security setting and the addresses of each region.

25.3.1.2 Features

The TZC-400 provides the following key features:

25.3.2 TZC400 register descriptions

25.3.2.1 TZC400 memory map

TZC400 base address: 110_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Build Configuration Register (BUILD_CONFIG)	32	RO	0000_2708h
4h	Action Register (ACTION)	32	RW	0000_0000h
8h	Gate Keeper Register (GATE_KEEPER)	32	RW	0000_0000h
Ch	Speculation Control Register (SPECULATION_CTRL)	32	RW	0000_0000h
10h	Interrupt Status Register (INT_STATUS)	32	RO	0000_0000h
14h	Interrupt Clear Register (INT_CLEAR)	32	WO	0000_0000h
20h	Fail Address Low Register (FAIL_ADDRESS_LOW_0)	32	RO	0000_0000h
24h	Fail Address High Register (FAIL_ADDRESS_HIGH_0)	32	RO	0000_0000h
28h	Fail Control register (FAIL_CONTROL_0)	32	RO	0000_0000h
100h	Region 0 Base Address Low Register (REGION_BASE_LOW_0)	32	RO	0000_0000h
104h	Region 0 Base Address High Register (REGION_BASE_HIGH_0)	32	RO	0000_0000h
108h	Region 0 Top Address Low Register (REGION_TOP_LOW_0)	32	RO	FFFF_FFFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
10Ch	Region 0 Top Address High Register (REGION_TOP_HIGH_0)	32	RO	0000_00FFh
110h	Region 0 Attribute Register (REGION_ATTRIBUTES_0)	32	RW	0000_0001h
120h	Region 1 Base Address Low Register (REGION_BASE_LOW_1)	32	RW	0000_0000h
124h	Region 1 Base Address High Register (REGION_BASE_HIGH_1)	32	RW	0000_0000h
128h	Region 1 Top Address Low Register (REGION_TOP_LOW_1)	32	RW	0000_0FFFh
12Ch	Region 1 Top Address High Register (REGION_TOP_HIGH_1)	32	RW	0000_0000h
130h	Region 1 Attribute Register (REGION_ATTRIBUTES_1)	32	RW	0000_0000h
140h	Region 2 Base Address Low Register (REGION_BASE_LOW_2)	32	RW	0000_0000h
144h	Region 2 Base Address High Register (REGION_BASE_HIGH_2)	32	RW	0000_0000h
148h	Region 2 Top Address Low Register (REGION_TOP_LOW_2)	32	RW	0000_0FFFh
14Ch	Region 2 Top Address High Register (REGION_TOP_HIGH_2)	32	RW	0000_0000h
150h	Region 2 Attribute Register (REGION_ATTRIBUTES_2)	32	RW	0000_0000h
160h	Region 3 Base Address Low Register (REGION_BASE_LOW_3)	32	RW	0000_0000h
164h	Region 3 Base Address High Register (REGION_BASE_HIGH_3)	32	RW	0000_0000h
168h	Region 3 Top Address Low Register (REGION_TOP_LOW_3)	32	RW	0000_0FFFh
16Ch	Region 3 Top Address High Register (REGION_TOP_HIGH_3)	32	RW	0000_0000h
170h	Region 3 Attribute Register (REGION_ATTRIBUTES_3)	32	RW	0000_0000h
180h	Region 4 Base Address Low Register (REGION_BASE_LOW_4)	32	RW	0000_0000h
184h	Region 4 Base Address High Register (REGION_BASE_HIGH_4)	32	RW	0000_0000h
188h	Region 4 Top Address Low Register (REGION_TOP_LOW_4)	32	RW	0000_0FFFh
18Ch	Region 4 Top Address High Register (REGION_TOP_HIGH_4)	32	RW	0000_0000h
190h	Region 4 Attribute Register (REGION_ATTRIBUTES_4)	32	RW	0000_0000h
1A0h	Region 5 Base Address Low Register (REGION_BASE_LOW_5)	32	RW	0000_0000h
1A4h	Region 5 Base Address High Register (REGION_BASE_HIGH_5)	32	RW	0000_0000h
1A8h	Region 5 Top Address Low Register (REGION_TOP_LOW_5)	32	RW	0000_0FFFh
1ACh	Region 5 Top Address High Register (REGION_TOP_HIGH_5)	32	RW	0000_0000h
1B0h	Region 5 Attribute Register (REGION_ATTRIBUTES_5)	32	RW	0000_0000h
1C0h	Region 6 Base Address Low Register (REGION_BASE_LOW_6)	32	RW	0000_0000h
1C4h	Region 6 Base Address High Register (REGION_BASE_HIGH_6)	32	RW	0000_0000h
1C8h	Region 6 Top Address Low Register (REGION_TOP_LOW_6)	32	RW	0000_0FFFh
1CCh	Region 6 Top Address High Register (REGION_TOP_HIGH_6)	32	RW	0000_0000h
1D0h	Region 6 Attribute Register (REGION_ATTRIBUTES_6)	32	RW	0000_0000h
1E0h	Region 7 Base Address Low Register (REGION_BASE_LOW_7)	32	RW	0000_0000h
1E4h	Region 7 Base Address High Register (REGION_BASE_HIGH_7)	32	RW	0000_0000h
1E8h	Region 7 Top Address Low Register (REGION_TOP_LOW_7)	32	RW	0000_0FFFh
1ECh	Region 7 Top Address High Register (REGION_TOP_HIGH_7)	32	RW	0000_0000h
1F0h	Region 7 Attribute Register (REGION_ATTRIBUTES_7)	32	RW	0000_0000h
200h	Region 8 Base Address Low Register (REGION_BASE_LOW_8)	32	RW	0000_0000h
204h	Region 8 Base Address High Register (REGION_BASE_HIGH_8)	32	RW	0000_0000h
208h	Region 8 Top Address Low Register (REGION_TOP_LOW_8)	32	RW	0000_0FFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
20Ch	Region 8 Top Address High Register (REGION_TOP_HIGH_8)	32	RW	0000_0000h
210h	Region 8 Attribute Register (REGION_ATTRIBUTES_8)	32	RW	0000_0000h
FE0h	Peripheral Identification 0 register (PID0)	32	RO	0000_0060h
FE4h	Peripheral Identification 1 register (PID1)	32	RO	0000_00B4h
FE8h	Peripheral Identification 2 register (PID2)	32	RO	0000_001Bh
FECh	Peripheral Identification 3 register (PID3)	32	RO	0000_0000h
FF0h	Component Identification 0 (CID0)	32	RO	0000_000Dh
FF4h	Component Identification 1 (CID1)	32	RO	0000_00F0h
FF8h	Component Identification 2 (CID2)	32	RO	0000_0005h
FFCh	Component Identification 3 (CID3)	32	RO	0000_00B1h

25.3.2.2 Build Configuration Register (BUILD_CONFIG)

25.3.2.2.1 Offset

Register	Offset
BUILD_CONFIG	0h

25.3.2.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						no_of_filters		Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		address_width						Reserved			no_of_regions				
W																
Reset	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0

25.3.2.2.3 Fields

Field	Function
31-26 —	Reserved
25-24 no_of_filters	Number of filters -minus one
23-14 —	Reserved
13-8 address_width	Address width minus one
7-5 —	Reserved
4-0 no_of_regions	Number of regions minus one

25.3.2.3 Action Register (ACTION)

25.3.2.3.1 Offset

Register	Offset
ACTION	4h

25.3.2.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														reaction_value	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.3.3 Fields

Field	Function
31-2 —	Reserved
1-0 reaction_value	Failure Reaction Controls how the TZC-400 uses the BRESPS[1:0], RRESPS[1:0], and TZCINT signals when a region permission failure occurs, excluding region overlap errors. When a region overlap for region 1 and higher occurs, this field also determines how TZCINT is set: 0b00, 0b01 - TZCINT LOW 0b10, 0b11 - TZCINT HIGH 00b - Sets TZCINT LOW and issues an OKAY response. 01b - Sets TZCINT LOW and issues a DECERR response. 10b - Sets TZCINT HIGH and issues an OKAY response. 11b - Sets TZCINT HIGH and issues a DECERR response.

25.3.2.4 Gate Keeper Register (GATE_KEEPER)

25.3.2.4.1 Offset

Register	Offset
GATE_KEEPER	8h

25.3.2.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															open_status
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															open_request
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.4.3 Fields

Field	Function
31-17 —	Reserved
16 open_status	Gate keeper open status This bit is set to 0 when both of the following conditions are fulfilled: • The gate keeper no longer permits access to its associated filter. • All outstanding accesses through the filter unit are complete. This means that the gate keeper always waits for outstanding accesses to complete. 0b - The gate keeper no longer permits access to its associated filter, that is, it is closed. 1b - The gate keeper permits access to its associated filter, that is, it is open.
15-1 —	Reserved
0 open_request	Gate keeper open request 0b - Requests the gate to be closed. 1b - Requests the gate to be open.

25.3.2.5 Speculation Control Register (SPECULATION_CTRL)

25.3.2.5.1 Offset

Register	Offset
SPECULATION_CTRL	Ch

25.3.2.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														write_spec_disable	read_spec_disable
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.5.3 Fields

Field	Function
31-2 —	Reserved
1 write_spec_disable	Write speculation disable control This bit controls write access speculation. **NOTE:** This bit is ignored and assumed to be zero at a filter unit if the corresponding QVNENABLE<x> signal is HIGH. 0b - Enables write access speculation. (Default) 1b - Disables write access speculation.
0 read_spec_disable	Read speculation disable control This bit controls read access speculation. **NOTE:** This bit is ignored and assumed to be zero at a filter unit if the corresponding QVNENABLE<x> signal is HIGH. 0b - Enables read access speculation. (Default) 1b - Disables read access speculation.

25.3.2.6 Interrupt Status Register (INT_STATUS)

25.3.2.6.1 Offset

Register	Offset
INT_STATUS	10h

25.3.2.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															overlap
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							overrun	Reserved							status
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.6.3 Fields

Field	Function
31-17 —	Reserved
16 overlap	Interrupt status - overlap When this bit is 1, the interrupt status bit is also set to 1. Clear the interrupt status of the associated bit in the Interrupt clear register to also clear this field. 0b - This bit is set even if the ACTION register is set to not drive the interrupt. 1b - When this bit is set to 1, it indicates a violation of the overlap region configuration rules for the associated filter unit. This occurs when an access matches with two enabled regions at the same time unless the overlap is only with Region 0.
15-9 —	Reserved
8 overrun	Interrupt status - overrun Clear the interrupt status of the associated bit in the Interrupt Clear Register (INT_CLEAR) to also clear this field.

Table continues on the next page...

Field	Function
	0b - This bit is set even if the ACTION register is set to not drive the interrupt. 1b - When a bit is set to 1, it indicates the occurrence of two or more region permission or region overlapping failures at the associated filter unit after the interrupt was cleared by the associated bit in the Interrupt clear register.
7-1 —	Reserved
0 status	Interrupt status This bit is set even if the ACTION register is set to not drive the interrupt output TZCINT HIGH. See Action register. See Action Register (ACTION) for details. Therefore, the status acts as an indicator that a region permission check failure or an overlap error has occurred at the filter unit. 0b - Interrupt is not asserted. 1b - Interrupt is asserted and waiting to be cleared.

25.3.2.7 Interrupt Clear Register (INT_CLEAR)

25.3.2.7.1 Offset

Register	Offset
INT_CLEAR	14h

25.3.2.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																clear
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.7.3 Fields

Field	Function
31-1 —	Reserved
0 clear	Interrupt clear Write a 1 to this bit to clear the associated status, overrun, and overlap bits in the INT_STATUS register. See Interrupt Status Register (INT_STATUS).

25.3.2.8 Fail Address Low Register (FAIL_ADDRESS_LOW_0)

25.3.2.8.1 Offset

Register	Offset
FAIL_ADDRESS_LOW_0	20h

25.3.2.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	addr_status_low															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	addr_status_low															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.8.3 Fields

Field	Function
31-0 addr_status_low	Failure access's low address If a region permission check fails or a region overlap occurs, this field returns the ACE-Lite address bits [31:0] of the first failed access. You must clear the associated interrupt status before this field can return the address of accesses of subsequent permission checks or region overlap failures. This occurs even if the ACTION register does not enable the interrupt. See Action Register (ACTION). If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in that filter unit do not update the associated fail status group of registers.

25.3.2.9 Fail Address High Register (FAIL_ADDRESS_HIGH_0)

25.3.2.9.1 Offset

Register	Offset
FAIL_ADDRESS_HIGH_0	24h

25.3.2.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								addr_status_high							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.9.3 Fields

Field	Function
31-8 —	Reserved
7-0 addr_status_high	Failure access's high address If a region permission check fails or a region overlap occurs, this field returns the ACE-Lite address bits [N+32:32] of the first failed access. You must clear the associated interrupt status before this field can return the addresses of the accesses of subsequent permission checks or region overlap failures. This occurs even if the ACTION register does not enable the interrupt. See Action Register (ACTION). If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in that filter unit do not update the associated fail status group of registers.

25.3.2.10 Fail Control register (FAIL_CONTROL_0)

25.3.2.10.1 Offset

Register	Offset
FAIL_CONTROL_0	28h

25.3.2.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved							direction	Reserved		nonsecure	privileged	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.10.3 Fields

Field	Function
31-25 —	Reserved
24 direction	Failure access's direction If a region permission check fails or a region overlap occurs, this field indicates whether the failed access was a read or write access attempt. You must clear the associated interrupt status before this field can return the direction of accesses of subsequent permission checks or region overlap failures. If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in that filter unit do not update the associated fail status group of registers. 0b - Read access 1b - Write access
23-22 —	Reserved
21 nonsecure	Failure access's security value If a region permission check fails or a region overlap occurs, this field indicates whether it was a Secure or Non-secure access attempt. You must clear the associated interrupt status before this field can return the direction of accesses of subsequent permission checks or region overlap failures. If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in that filter unit do not update the associated fail status group of registers.

Table continues on the next page...

Field	Function
	0b - Secure access 1b - Non-secure access
20 privileged	Failure access's privileged value If a region permission check fails or a region overlap occurs, this field indicates whether it was an unprivileged or privileged access attempt. You must clear the associated interrupt status before this field can return the values of accesses of subsequent permission checks or region overlap failures. If the status flag for the filter unit in the INT_STATUS register is already set, new region permission check failures in that filter unit do not update the associated fail status group of registers. 0b - Unpriviledged access 1b - Priviledged access
19-0 —	Reserved

25.3.2.11 Region 0 Base Address Low Register (REGION_BASE_LOW_0)

25.3.2.11.1 Offset

Register	Offset
REGION_BASE_LOW_0	100h

25.3.2.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	base_address_low_n															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	base_address_low_n				Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.11.3 Fields

Field	Function
31-12 base_address_low_n	Region base address lower 32 bits. This bit controls the base address bits[31:12] of Region <n>a. The TZC-400 only permits a region to start at a 4 KB aligned address and address bits[11:0] are zeros.
11-0 —	Reserved

25.3.2.12 Region 0 Base Address High Register (REGION_BASE_HIGH_0)

25.3.2.12.1 Offset

Register	Offset
REGION_BASE_HIGH_0	104h

25.3.2.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								base_address_high_n							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.12.3 Fields

Field	Function
31-8 —	Reserved
7-0	Region base address higher bits

Field	Function
base_address_high_n	This bit controls the base address bits 32 and higher of Region <n>b. For Region 0, this field is read-only. The TZC-400 sets the base address of Region 0 to 0x0.

25.3.2.13 Region 0 Top Address Low Register (REGION_TOP_LOW_0)

25.3.2.13.1 Offset

Register	Offset
REGION_TOP_LOW_0	108h

25.3.2.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	top_address_low_n															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	top_address_low_n				HIGH_4KB											
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

25.3.2.13.3 Fields

Field	Function
31-12 top_address_low_n	Region top address lower 32 bits. This bit controls the region top address bits[31:12] of region <n>a. This address points to the start of the next 4 KB aligned address immediately outside the region. For Region 0, this field is read-only and all bits are set HIGH.

Table continues on the next page...

Field	Function
11-0 HIGH_4KB	HIGH_4KB 0xFFF This bit points to the highest byte of a 4KB-aligned area.

25.3.2.14 Region 0 Top Address High Register (REGION_TOP_HIGH_0)

25.3.2.14.1 Offset

Register	Offset
REGION_TOP_HIGH_0	10Ch

25.3.2.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								top_address_high_n							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

25.3.2.14.3 Fields

Field	Function
31-8 —	Reserved
7-0 top_address_high_n	Region top address higher bits This bit controls the region top address bits 32 and higher of region <n>b. This address points to the highest address of the region. For region 0, this field is read-only and always returns 0xFFFFFFFF.

25.3.2.15 Region 0 Attribute Register (REGION_ATTRIBUTES_0)

25.3.2.15.1 Offset

Register	Offset
REGION_ATTRIBUTES_0	110h

25.3.2.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	s_wr_en	s_rd_en	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															filter_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

25.3.2.15.3 Fields

Field	Function
31 s_wr_en	Region secure write enable Secure global write enable. This control bit defines the permissions for the secure writes in the region. 0b - Secure write to the region is not permitted. 1b - Secure write to the region is permitted.
30 s_rd_en	Region secure read enable 0b - Secure read to the region is not permitted. 1b - Secure read to the region is permitted.
29-1 —	Reserved
0 filter_en	Region filter enable When set HIGH, it enables the use of the current region programming for the associated filter. For Region 0, this bit is set HIGH and cannot be modified.

25.3.2.16 Region a Base Address Low Register (REGION_BASE_LOW_1 - REGION_BASE_LOW_8)

25.3.2.16.1 Offset

For a = 1 to 8:

Register	Offset
REGION_BASE_LOW_a	100h + (a × 20h)

25.3.2.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	base_address_low_n															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	base_address_low_n				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.16.3 Fields

Field	Function
31-12 base_address_low_n	Region base address lower 32 bits. This bit controls the base address bits[31:12] of Region <n>a. The TZC-400 only permits a region to start at a 4 KB aligned address and address bits[11:0] are zeros.
11-0 —	Reserved

25.3.2.17 Region a Base Address High Register (REGION_BASE_HIGH_1 - REGION_BASE_HIGH_8)

25.3.2.17.1 Offset

For a = 1 to 8:

Register	Offset
REGION_BASE_HIGH_a	104h + (a × 20h)

25.3.2.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								base_address_high_n							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.17.3 Fields

Field	Function
31-8 —	Reserved
7-0 base_address_high_n	Region base address higher bits This bit controls the base address bits 32 and higher of Region <n>b. For Region 0, this field is read-only. The TZC-400 sets the base address of Region 0 to 0x0.

25.3.2.18 Region a Top Address Low Register (REGION_TOP_LOW_1 - REGION_TOP_LOW_8)

25.3.2.18.1 Offset

For a = 1 to 8:

Register	Offset
REGION_TOP_LOW_a	108h + (a × 20h)

25.3.2.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	top_address_low_n															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	top_address_low_n				HIGH_4KB											
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

25.3.2.18.3 Fields

Field	Function
31-12 top_address_low_n	Region top address lower 32 bits. This bit controls the region top address bits[31:12] of region <n>a. This address points to the start of the next 4 KB aligned address immediately outside the region. For Region 0, this field is read-only and all bits are set HIGH.
11-0 HIGH_4KB	HIGH_4KB 0xFFF This bit points to the highest byte of a 4KB-aligned area.

25.3.2.19 Region a Top Address High Register (REGION_TOP_HIGH_1 - REGION_TOP_HIGH_8)

25.3.2.19.1 Offset

For a = 1 to 8:

Register	Offset
REGION_TOP_HIGH_a	10Ch + (a × 20h)

25.3.2.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								top_address_high_n							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.19.3 Fields

Field	Function
31-8 —	Reserved
7-0 top_address_high_n	Region top address higher bits This bit controls the region top address bits 32 and higher of region <n>b. This address points to the highest address of the region. For region 0, this field is read-only and always returns 0xFFFFFFFF.

25.3.2.20 Region a Attribute Register (REGION_ATTRIBUTES_1 - REGION_ATTRIBUTES_8)

25.3.2.20.1 Offset

For a = 1 to 8:

Register	Offset
REGION_ATTRIBUTES_a	110h + (a × 20h)

25.3.2.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	s_wr_en	s_rd_en	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															filter_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.20.3 Fields

Field	Function
31 s_wr_en	Region secure write enable Secure global write enable. This control bit defines the permissions for the secure writes in the region. 0b - Secure write to the region is not permitted. 1b - Secure write to the region is permitted.
30 s_rd_en	Region secure read enable 0b - Secure read to the region is not permitted. 1b - Secure read to the region is permitted.
29-1 —	Reserved
0 filter_en	Region filter enable When set HIGH, it enables the use of the current region programming for the associated filter. For Region 0, this bit is set HIGH and cannot be modified.

25.3.2.21 Peripheral Identification 0 register (PID0)

25.3.2.21.1 Offset

Register	Offset
PID0	FE0h

25.3.2.21.2 Function

Peripheral Identification (PID) registers contain TZC-400 IP block identity information. These are not operational registers.

25.3.2.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								part_number_0							
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

25.3.2.21.4 Fields

Field	Function
31-8 —	Reserved
7-0 part_number_0	Part number 8 bits. These bits read back as 0x60.

25.3.2.22 Peripheral Identification 1 register (PID1)

25.3.2.22.1 Offset

Register	Offset
PID1	FE4h

25.3.2.22.2 Function

Peripheral Identification (PID) registers contain TZC-400 IP block identity information. These are not operational registers.

25.3.2.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								jep106_id_3_0				part_number_0			
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0

25.3.2.22.4 Fields

Field	Function
31-8 —	Reserved
7-4 jep106_id_3_0	JEP106 Arm's identification code JEP106 Arm's identification code JEP106 identity code [3:0]. See the JEP106, Standard Manufacturer Identification Code. These bits read back as 0xB because Arm is the peripheral designer.
3-0 part_number_0	Part number. These bits read back as 0x4.

25.3.2.23 Peripheral Identification 2 register (PID2)

25.3.2.23.1 Offset

Register	Offset
PID2	FE8h

25.3.2.23.2 Function

Peripheral Identification (PID) registers contain TZC-400 IP block identity information. These are not operational registers.

25.3.2.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								revision				jedec_used	jep106_id_6_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

25.3.2.23.4 Fields

Field	Function
31-8 —	Reserved
7-4 revision	Revision. Identifies the revision of the TZC-400.
3 jedec_used	JEP106 manufacturer's identity code. Indicates the use of JEP106 manufacturer's identity code. This indicates that the TZC-400 uses a manufacturer identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1.
2-0 jep106_id_6_4	JEP106 Arm's identification code bits 6 down to 4. These bits read back as 0x3 because Arm is the peripheral designer.

25.3.2.24 Peripheral Identification 3 register (PID3)

25.3.2.24.1 Offset

Register	Offset
PID3	FECh

25.3.2.24.2 Function

Peripheral Identification (PID) registers contain TZC-400 IP block identity information. These are not operational registers.

25.3.2.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								revand				mod_number			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.3.2.24.4 Fields

Field	Function
31-8 —	Reserved
7-4 revand	TZC-400 revision number
3-0 mod_number	Modification Number. This bit reads back as 0x0.

25.3.2.25 Component Identification 0 (CID0)

25.3.2.25.1 Offset

Register	Offset
CID0	FF0h

25.3.2.25.2 Function

This is one of four 8-bit registers that together hold a 32 bit component ID value. This register is used for automatic BIOS configuration. The full 32-bit component_id is set to 0xB105F00D.

25.3.2.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								comp_id_0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

25.3.2.25.4 Fields

Field	Function
31-8 —	Reserved
7-0 comp_id_0	Component identity 0

25.3.2.26 Component Identification 1 (CID1)

25.3.2.26.1 Offset

Register	Offset
CID1	FF4h

25.3.2.26.2 Function

This is one of four 8-bit registers that together hold a 32 bit component ID value. This register is used for automatic BIOS configuration. The full 32-bit component_id is set to 0xB105F00D.

25.3.2.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								comp_id_1							
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

25.3.2.26.4 Fields

Field	Function
31-8 —	Reserved
7-0 comp_id_1	Conponent identity 1

25.3.2.27 Component Identification 2 (CID2)

25.3.2.27.1 Offset

Register	Offset
CID2	FF8h

25.3.2.27.2 Function

This is one of four 8-bit registers that together hold a 32 bit component ID value. This register is used for automatic BIOS configuration. The full 32-bit component_id is set to 0xB105F00D.

25.3.2.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								comp_id_2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

25.3.2.27.4 Fields

Field	Function
31-8 —	Reserved
7-0 comp_id_2	Conponent identity 2

25.3.2.28 Component Identification 3 (CID3)

25.3.2.28.1 Offset

Register	Offset
CID3	FFCh

25.3.2.28.2 Function

This is one of four 8-bit registers that together hold a 32 bit component ID value. This register is used for automatic BIOS configuration. The full 32-bit component_id is set to 0xB105F00D.

25.3.2.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								comp_id_3							
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

25.3.2.28.4 Fields

Field	Function
31-8 —	Reserved
7-0 comp_id_3	Conponent identity 3

25.3.3 Functional Description

This chapter describes the functionality of the TZC-400. It contains the following sections:

- TZC-400 operation.
- Constraints of use.

25.3.3.1 TZC-400 operation

This section describes the operation of the TZC-400. It contains the following sections:

- Access filtering with regions.

- Speculative accesses.
- Fast path selection.
- Gate keeper.
- Barriers.
- Exclusive accesses.
- Denied ACE-Lite transactions.

25.3.3.1.1 Access filtering with regions

The TZC-400 enables you to program the access permissions for each region. A region is a contiguous area in the address space with a defined start and end. Each region has a security level setting. A bus access must meet this security level to be given access to the required resource. The TZC-400 supports nine regions:

- One base region, Region 0, that is partially programmable from software.
- Eight other regions, Regions 1-8, that are fully programmable from software.

The security state of transactions is indicated by the *ARPROTS[1]* or *AWPROTS[1]* signals, where a value of 0 indicates a TZ Secure access.

Through the control unit, you can program the security setting and addresses of each region. The filter unit performs security checking when a master tries to access the memory region of a slave.

Region 0 is the default region. It covers any memory space that is not part of another region.

You can define a region and then disable checking of that region in software.

25.3.3.1.1.1 Region access rules

The TZC-400 implements the following region rules:

- A region must be enabled for a filter unit to provide a valid match.
- Region 0 is always enabled.
- TZC-400 checks the access against Region 0 security settings only if the access cannot be found in any of the other enabled regions for that filter unit.

- Where an address maps to region 1 or higher, TZC-400 checks the access against the security settings of that region.
- For region 1 and higher, any enabled region in a filter unit must not overlap memory areas from another enabled region.

NOTE

- Other regions can overlap the address area of Region 0. You can use an overlap between Region 0 and a higher region to manage all access rights for a Secure OS, by defining the security of the default region, Region 0, and then defining security for higher regions overlapping the default base region. For example, you can make Region 0 inaccessible to any Non-secure masters. The Secure OS can then selectively release regions for global access later.
-
- The behavior of the TZC-400 is *Undefined* for configurations where Regions 1 and higher overlap when enabled on the same filter unit. When an access to an overlapping region occurs, the TZC-400 sets a status bit to indicate an overlapping access. The TZC-400 can generate an interrupt when this occurs. Interrupt generation is a programmable feature of the TZC-400. See Action Register (ACTION).

25.3.3.1.1.2 Example region setup

Figure 25-2 shows an example security configuration with Regions 0-3.

Figure 25-2. Region mapping example on a 32-bit address map for a filter unit

In this example Region 3 is disabled, and Region 1 and Region 2 are enabled.

Access A is mapped to Region 0 because its address is not in the regions that Regions 1-3 define.

Access B is mapped to Region 1.

Access C is mapped to Region 0 because Region 3 is not enabled.

25.3.3.1.1.3 Region setup

The following table describes the programmable parameters for Region 0 and Regions 1-8, respectively.

Table 25-3. Programmable parameters of regions

Parameter	Description	Value for Region 0	Value for Region 1 - Region 8
Region enables.	Enables and disables access checking for filter unit and each region.	Always enabled.	Programmable.
Secure access permissions.	Defines the Secure read and write permissions for the region.	Programmable.	Programmable.
Base address.	Defines the start address of the region, and is aligned to a 4KB boundary.	0[1].	Programmable.
Top address.	Defines the top address of the region, and is aligned to the top byte address of a 4KB region.	All bits are set to 1[1].	Programmable.

1. This means that Region 0 occupies the entire memory space.

25.3.3.1.1.4 Access permissions to regions

You must program access permissions separately for TZ Secure access and TZ Non-secure access.

Each region contains the following configuration registers:

- REGION_ATTRIBUTES
- REGION_ID_ACCESS

The table below shows the region settings that you can define for TZ Secure read and TZ Secure write enables and relates this to the outcome that a TZ Secure access can achieve. <*n*> is the filter unit number, which is always zero (single filter unit) in the chip.

Table 25-4. Permissions for TZ Secure accesses

Region parameters		Resulting permissions that grant access	
REGION_ATTRIBUTES_<n>.s_wr_en	REGION_ATTRIBUTES_<n>.s_rd_en	TZ Secure write	TZ Secure read
0	0	No	No
0	1	No	Yes
1	0	Yes	No
1	1	Yes	Yes

You must explicitly program a region to be granted TZ Secure access if it is required, regardless of the settings for TZ Non-secure reads and TZ Non-secure writes. For example, when permissions are set so that a TZ Non-secure read or TZ Non-secure write access is permitted, it does not mean that the corresponding TZ Secure read or TZ Secure write access is also automatically granted.

For each type of access, the filter unit checks whether the Non-secure access address resides in any region between 1 and 8 that is enabled for that filter. If the filter finds a match, then it uses that region. Otherwise, the filter selects the default region 0.

When the filter unit detects that an access falls into an overlapping address area that two enabled regions define, it sets the overlap status bit in the Interrupt Status Register. The only exception to this is if the overlap only occurs with Region 0. In this case, the filter unit does not change the overlap status bit in the Interrupt Status Register. See INT_STATUS.

25.3.3.1.2 Speculative accesses

The default behavior of the TZC-400 is to perform speculative accesses. You can disable speculative accesses if they have detrimental side effects, or might affect system security. See the Speculation Control Register (SPECULATION_CTRL).

When speculative access is enabled, the TZC-400 continues to allow accesses that fail security checks to progress through to the external slave interface. If this happens, the TZC-400 ensures that any read data, write data, and write enables are set to zero, so that no data is visible and no write transaction occurs.

25.3.3.1.3 Fast path selection

By default, all reads are speculative and take the fast path. This chip supports 16 outstanding fast path reads. If read speculation is disabled, reads will take the normal path.

Writes are also speculative by default, however writes always take the normal path. The chip supports up to 256 outstanding normal path.

25.3.3.1.4 Gate keeper

The TZC-400 implements gate keeper functionality on the ACE-Lite slave interface of each filter unit to permit or block accesses progressing through the filter unit. The two states are also called open and closed, respectively. You can use the gate keeper to suspend processing transactions immediately before changing region settings. See Changing TZC-400 region settings on an active system.

Each gate keeper provides an independent control bit in the GATE_KEEPER register, called open_request[<*x*>], where x is the filter unit number. This bit enables software to request the gate to open or close. An independent status bit, open_status[<*x*>], is also provided for each filter unit. This bit indicates whether the gate is open or closed. See Gate Keeper Register (GATE_KEEPER).

All open_request control bits default to zero at reset. To request access through a filter unit, software must set the corresponding bit of open_request to 1. When set, the corresponding open_status bit transitions to 1 and then the gate keeper starts to permit accesses to pass through.

To request gate closure of a filter unit, software must set the corresponding bit of open_request to 0. When this occurs, the filter unit no longer permits any new access arriving on the AW and AR channels to progress to the filter unit. It also waits for all outstanding accesses that have already entered the filter unit to complete. This includes both those currently progressing through the filter unit and those already issued by the filter unit. When this has occurred, the corresponding open_status bit goes to 0.

25.3.3.1.5 Barriers

Transactions that arrive after a barrier is received do not proceed until all transactions preceding the barrier are sent, and then the barrier is sent.Then the transactions that arrived after the barrier are sent. This process applies irrespective of:

- The IDs of all transactions.
- The path each transaction takes through the TZC-400.

Barriers are handled in a different manner to other transactions in the TZC-400, with the following differences:

- Security checks are not performed on barrier transactions. This is because they do not contain an address and they do not transfer any data.
- Barriers are never sent along the fast path even if *FPID*<*x*> is set HIGH.

25.3.3.1.6 Exclusive accesses

If a master performs exclusive accesses to an address region, you must program the TZC-400 to permit read and write accesses to that address region for the expected security access settings. Otherwise, the read or write transaction might fail.

When a read exclusive access fails its security check, if the filter unit currently supports speculative reads, then the read exclusive access is sent as a read access with the *ARLOCKM<x>[0]* bit set to 0. However, a failed write exclusive access, when speculation is active, does not have its *AWLOCKM<x>[0]* bit cleared when the transaction is issued from the TZC-400. All exclusive read accesses use the normal path regardless of the associated *FPID<x>* value.

25.3.3.1.7 Denied ACE-Lite transactions

If an ACE-Lite transaction has insufficient security privileges, then for:

Reads

The TZC-400 responds to the master by setting all bits of the read data bus, *RDATAS<x>*, to zero. If you program the TZC-400 to perform speculative accesses, the TZC-400 still issues the read access, but its corresponding read data, *RDATAM<x>*, is discarded.

NOTE

Writes

The TZC-400 prevents the transfer of data from the master to the slave by discarding the data that *WDATAM<x>* contains. If you program the TZC-400 to perform speculative accesses, it modifies the transfer to the slave by setting all bits of the:

- Write data bus, *WDATAM<x>*, to zero.
- Write data strobe, *WSTRBM<x>*, to zero.

The action configuration register controls how the TZC-400 signals the region permission failure of an access. This does not include region overlap conditions. See Action Register (ACTION). The available response strategies of TZC-400 are:

OK and raise no interrupts

This enables the access to fail without permitting the read data to be returned or the write data to be committed. From the perspective of the master that issued the access, the access was permitted.

OK and raise an interrupt

This generates a bus response, in the same way as the OK, and raises no interrupts. However, TZC-400 generates an interrupt that must be routed to a Secure OS to notify it that a security violation has occurred.

Decode error and raise no interrupts

This enables the issuing master to behave as though the memory location is absent. The TZC-400 presents a DECERR response to indicate that the transaction has failed.

Decode error and raise an interrupt

This enables the issuing master to behave as though the memory location is absent. The TZC-400 presents a DECERR response and generates an interrupt that must be routed to a Secure OS to notify it that a security violation has occurred.

For region overlap failure conditions, the bus response of the filter unit is unpredictable, but the overlap and status bits of the filter unit are set, and an interrupt is raised depending on the settings in the Action Register. See Action Register (ACTION). The FAIL status registers of the associated filter unit are set using values from the failed access.

25.3.3.1.8 Reset

The TZC-400 maintains a master copy of all region configurations in the configuration registers in the control unit. These are automatically copied in the filter unit as soon as the filter exits reset. The filter unit blocks all accesses to the ACE-Lite slave interface while copying these settings over. When copying is complete, the TZC-400 starts permitting access through to the filter unit for permission checks.

25.3.3.2 Constraints of use

This section describes:

- Changing TZC-400 region settings on an active system.

25.3.3.2.1 Changing TZC-400 region settings on an active system

No outstanding ACE-Lite transactions may use the region when a change of setting is in progress. You can achieve this as follows:

- Ensure that no access is sent to the region during the period when the region settings are being changed. However, the TZC-400 has no control over this and therefore the other masters in the system are responsible for this.
- Software can use the gate keeper to request that all gates are closed. When all gate keeper status bits go to 0, there are no outstanding accesses and therefore changes to the regions can be made before opening the gates again.

  However, when the gates are closed, all accesses on all incoming ACE-Lite ports are blocked. This can potentially result in a system deadlock. This occurs if an access to the APB port of TZC-400 is blocked indirectly.

When there are no outstanding ACE-Lite transactions, the settings can be changed by writing to the TZC-400 APB interface. To ensure that all updates have been committed before starting transactions again, you must perform a read operation from the APB register in the TZC-400.

25.4 TrustZone Protection Controller (TZPC)

25.4.1 About the TrustZone Protection Controller

The TrustZone Protection Controller (TZPC), is an AMBA-compliant, SoC peripheral that is developed, tested, and licensed by Arm Limited.

The TZPC acts as a programming interface for configuring the TrustZone access permissions for chip memory regions. The TZPC holds the configuration registers while the TZMA performs the actual enforcement.

The TZPC has the following features:

- it has protection bits to enable you to program up to 24 areas of memory as secure or non-secure
- it has secure region bits to enable you to split an area of internal RAM into both secure and non-secure regions

The figure below shows a simplified block diagram of the TZPC.

Figure 25-3. TrustZone Protection Controller

In this chip, the TZPC and TZMA are only used to configure and execute TZ access control over the On-Chip SRAM (OCRAM).

25.4.2 TZPC Functional description

The TZPC provides a software interface to set up memory areas as TZ secure or TZ non-secure. It does this in two ways:

- Programmable protection bits that can be allocated to areas of memory as determined by an external decoder.
- Programmable region size value for use by an AXI TrustZone Memory Adapter (TZMA).

This programmable flexibility enables you to enables you to allow internal on-chip RAM to be partitioned between TZ secure and TZ non-secure access, and make other memory-mapped peripherals TZ non-secure accessible, based on the specific application's requirements.

The TZ secure and TZ non-secure memory partitioning is not expected to change dynamically during normal software operation. It is fixed at compile time and is only configured once during system boot-up. Ensure that this boot-up is always made in secure-state to guarantee full security protection.

The following figure shows a TZPC configured in a typical TrustZone-enabled design.

Figure 25-4. Typical configuration

25.4.3 TZPC register descriptions

25.4.3.1 TZPC Peripheral Identification Registers 0-3

The read-only TZPCPERIPHID0-3 Registers are four 8-bit registers, that span address locations 0xFE0-0xFEC. The figure below shows how you can treat the registers conceptually as a single 32-bit register. The registers provide the following options for the peripheral:

Part number [11:0]

This identifies the peripheral. The three digit product code 0x870 is used for the TZPC.

Designer [19:12]

This is the identification of the designer. Arm Limited is 0x41, ASCII A.

Revision number [23:20]

This is the revision number of the peripheral. The revision number starts from 0 and the value is revision-dependent.

Configuration [31:24]

This is the configuration option of the peripheral. The configuration value is 0.

Figure 25-5. Peripheral identification Register bit assignments

See TZPC Peripheral Identification Register 0 (TZPCPERIPHID0), TZPC Peripheral Identification Register 1 (TZPCPERIPHID1), TZPC Peripheral Identification Register 2 (TZPCPERIPHID2), and TZPC Peripheral Identification Register 3 (TZPCPERIPHID3) for details.

25.4.3.2 Identification Registers 0-3

The read-only TZPCPCELLID0-3 Registers are four 8-bit registers that span address locations 0xFF0-0xFFC. You can treat the register conceptually as a single 32-bit register. The register is used for a standard cross-peripheral identification system.

Figure 25-6. TZPC Identification Register bit assignments

See TZPC Identification Register 0 (TZPCPCELLID0), TZPC Identification Register 1 (TZPCPCELLID1), TZPC Identification Register 2 (TZPCPCELLID2), and TZPC Identification Register 3 (TZPCPCELLID3) for details.

25.4.3.3 TZPC memory map

The following table lists the TZPC registers in base offset order.

TZPC base address: 220_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Secure RAM Region Size Register (TZPCR0SIZE)	32	RW	0000_0200h
800h	TZDECPROT Register 0 Stat (TZDECPROT0_Stat)	32	RW	0000_0000h
804h	TZDECPROT Register 0 Set (TZDECPROT0_Set)	32	RW	0000_0000h
808h	TZDECPROT Register 0 Clr (TZDECPROT0_Clr)	32	RW	0000_0000h
80Ch	TZDECPROT Register 1 Stat (TZDECPROT1_Stat)	32	RW	0000_0000h
810h	TZDECPROT Register 1 Set (TZDECPROT1_Set)	32	RW	0000_0000h
814h	TZDECPROT Register 1 Clr (TZDECPROT1_Clr)	32	RW	0000_0000h
FE0h	TZPC Peripheral Identification Register 0 (TZPCPERIPHID0)	32	RO	0000_0070h
FE4h	TZPC Peripheral Identification Register 1 (TZPCPERIPHID1)	32	RO	0000_0018h
FE8h	TZPC Peripheral Identification Register 2 (TZPCPERIPHID2)	32	RO	0000_0004h
FECh	TZPC Peripheral Identification Register 3 (TZPCPERIPHID3)	32	RO	0000_0000h
FF0h	TZPC Identification Register 0 (TZPCPCELLID0)	32	RO	0000_000Dh
FF4h	TZPC Identification Register 1 (TZPCPCELLID1)	32	RO	0000_00F0h
FF8h	TZPC Identification Register 2 (TZPCPCELLID2)	32	RO	0000_0005h
FFCh	TZPC Identification Register 3 (TZPCPCELLID3)	32	RO	0000_00B1h

25.4.3.4 Secure RAM Region Size Register (TZPCR0SIZE)

25.4.3.4.1 Offset

Register	Offset
TZPCR0SIZE	0h

25.4.3.4.2 Function

The read and write TZPCR0SIZE Register has a reset value of 0x00000200. It sets the size of the secure region in the internal RAM. This reset value means the entire OCRAM is TZ secure world only when the chip comes out of reset. Trusted software can later change R0Size to allow TZ non-secure world to access some or all of the OCRAM.

25.4.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						R0SIZE									
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

25.4.3.4.4 Fields

Field	Function
31-10 —	- Read undefined. Write as zero.
9-0 R0SIZE	R0SIZE Secure RAM region size in 4KB steps: 0x00000000 = no secure region 0x00000001 = 4KB secure region 0x00000002 = 8KB secure region … 0x000001FF = 2044KB secure region. 0x00000200 or above sets the entire RAM to secure regardless of size

25.4.3.5 TZDECPROT Register 0 Stat (TZDECPROT0_Stat)

25.4.3.5.1 Offset

Register	Offset
TZDECPROT0_Stat	800h

25.4.3.5.2 Function

The read-only TZDECPROT register 0 stat register has a reset value of 0x00000000. It provides the TrustZone protection status of the memory-mapped registers associated with each module.

This register is only accessible by TZ secure world software.

25.4.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DDRC	Reserved			DCFG_RESET	SPI_1	I2C_1	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.5.4 Fields

Field	Function
31-8 —	- Reserved
7 DDRC	DDRC Indicates the write protection status of DDR controller configuration register space (0x108_0000 - 0x108_FFFF)

Table continues on the next page...

Field	Function
	0b - locked (not writable) by any transactions. 1b - unlocked (writable) by all transactions, both secure and non-secure.
6-4 —	- Reserved
3 DCFG_RESET	DCFG_RESET Indicates the TZ protection status of DCFG (0x1E0_0000 - 0x1E0_FFFF) and Reset (0x1E6_0000 - 0x1E6_FFFF) memory spaces. 0b - DCFG and Reset address space is accessible by TZ secure world only 1b - DCFG and Reset address space is accessible by TZ non-secure world
2 SPI_1	SPI_1 Indicates the TZ protection status of SPI_1 (0x210_0000 - 0x210_FFFF) memory space. 0b - SPI address space is accessible by TZ secure world only 1b - SPI address space is accessible by TZ non-secure world
1 I2C_1	I2C_1 Indicates the TZ protection status of I2C_1 (0x200_0000 - 0x200_FFFF) memory space. 0b - I2C1 address space is accessible by TZ secure world only 1b - I2C1 address space is accessible by TZ non-secure world
0 —	- Reserved

25.4.3.6 TZDECPROT Register 0 Set (TZDECPROT0_Set)

25.4.3.6.1 Offset

Register	Offset
TZDECPROT0_Set	804h

25.4.3.6.2 Function

This register is only accessible by trusted software.

The write-only TZDECPROT register 0 set register has a reset value of 0x00000000. It is used to change the TrustZone protection status of the memory-mapped registers associated with each module.

This register is only accessible by TZ secure world software.

25.4.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DDRC	Reserved			DCFG_RESET	SPI_1	I2C_1	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.6.4 Fields

Field	Function
31-8 —	- Reserved
7 DDRC	DDRC Sets the write protection status of DDR controller configuration register space configuration register space (0x108_0000 - 0x108_FFFF) 0b - No effect, TZ secure world only 1b - Allows writes by all transactions, both TZ secure world and TZ non-secure world
6-4 —	- Reserved
3 DCFG_RESET	DCFG_RESET Sets the TZ protection status of DCFG (0x1E0_0000 - 0x1E0_FFFF) and Reset (0x1E6_0000 - 0x1E6_FFFF) memory spaces. 0b - No effect, TZ secure world only 1b - Allows writes by all transactions, both TZ secure world and TZ non-secure world
2 SPI_1	SPI_1 Sets the TZ protection status of SPI_1 (0x210_0000 - 0x210_FFFF) memory space. 0b - No effect, TZ secure world only 1b - Allows writes by all transactions, both TZ secure world and TZ non-secure world
1 I2C_1	I2C_1 Sets the TZ protection status of I2C_1 (0x200_0000 - 0x200_FFFF) memory space. 0b - No effect, TZ secure world only 1b - Allows writes by all transactions, both TZ secure world and TZ non-secure world
0 —	- Reserved

25.4.3.7 TZDECPROT Register 0 Clr (TZDECPROT0_Clr)

25.4.3.7.1 Offset

Register	Offset
TZDECPROT0_Clr	808h

25.4.3.7.2 Function

The write-only TZDECPROT Register 0 Clear register has a reset value of 0x00000000. It is used to restore the TrustZone protection status of the memory-mapped registers associated with each module.

This register is only accessible by TZ secure world software.

25.4.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DDRC	Reserved			DCFG_RESET	SPI_1	I2C_1	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.7.4 Fields

Field	Function
31-8 —	- Reserved
7 DDRC	DDRC Restores the default write protection status of DDR controller configuration register space configuration register space (0x108_0000 - 0x108_FFFF).

Table continues on the next page...

Field	Function
	0b - No effect 1b - Blocks writes by all transactions, both TZ secure world and TZ non-secure world
6-4 —	- Reserved
3 DCFG_RESET	DCFG_RESET Restores the default TZ protection status of DCFG (0x1E0_0000 - 0x1E0_FFFF) and Reset (0x1E6_0000 - 0x1E6_FFFF) memory spaces. 0b - No effect 1b - Restores the protection to TZ secure world only
2 SPI_1	SPI_1 Restores the default TZ protection status of SPI_1 (0x210_0000 - 0x210_FFFF) memory space. 0b - No effect 1b - Restores the protection to TZ secure world only
1 I2C_1	I2C_1 Restores the default TZ protection status of I2C_1 (0x200_0000 - 0x200_FFFF) memory space. 0b - No effect 1b - Restores the protection to TZ secure world only
0 —	- Reserved

25.4.3.8 TZDECPROT Register 1 Stat (TZDECPROT1_Stat)

25.4.3.8.1 Offset

Register	Offset
TZDECPROT1_Stat	80Ch

25.4.3.8.2 Function

The read-only TZDECPROT Register 1 Stat register has a reset value of 0x00000000. It provides the TrustZone protection status of the control signals listed for each associated bit.

This register is only accessible by TZ secure world software.

25.4.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TZ_WDT_CTL	Reserved			SPIDEN	SPNIDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.8.4 Fields

Field	Function
31-6 —	- Reserved
5 TZ_WDT_CTL	TZ_WDT_CTL TrustZone Watchdog Control 0b - TZ WDOG timer operates independently of the debug halted state of the GPP cores (0 = default), AND RSTCNTL[RST_REQ_MSK] IS NOT allowed to block any device reset request 1b - TZ WDOG timer is halted when all GPP cores are in a debug halted state, AND RSTCNTL[RST_REQ_MSK] IS allowed to block any device reset request
4-2 —	- Reserved
1 SPIDEN	SPIDEN Global secure privileged invasive debug 0b - not enabled 1b - enabled
0 SPNIDEN	SPNIDEN Global secure privileged non-invasive debug 0b - not enabled 1b - enabled

25.4.3.9 TZDECPROT Register 1 Set (TZDECPROT1_Set)

25.4.3.9.1 Offset

Register	Offset
TZDECPROT1_Set	810h

25.4.3.9.2 Function

The write-only TZDECPROT Register 1 Set register has a reset value of 0x00000000. It is used to change the TrustZone protection status of the control signals listed for each associated bit.

This register is only accessible by TZ secure world software.

25.4.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TZ_WDT_CTL	Reserved			SPIDEN	SPNIDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.9.4 Fields

Field	Function
31-6 —	- Reserved
5 TZ_WDT_CTL	TZ_WDT_CTL TrustZone Watchdog Control 0b - TZ WDOG timer operates independently of the debug halted state of the GPP cores (0 = default), AND RSTCNTL[RST_REQ_MSK] IS NOT allowed to block any device reset request 1b - TZ WDOG timer is halted when all GPP cores are in a debug halted state, AND RSTCNTL[RST_REQ_MSK] IS allowed to block any device reset request
4-2 —	- Reserved

Table continues on the next page...

Field	Function
1 SPIDEN	SPIDEN Global secure privileged invasive debug 0b - not enabled 1b - enabled
0 SPNIDEN	SPNIDEN Global secure privileged non-invasive debug 0b - not enabled 1b - enabled

25.4.3.10 TZDECPROT Register 1 Clr (TZDECPROT1_Clr)

25.4.3.10.1 Offset

Register	Offset
TZDECPROT1_Clr	814h

25.4.3.10.2 Function

The write-only TZDECPROT Register 1 Clear register has a reset value of 0x00000000. It is used to restore the TrustZone protection status of the control signals listed for each associated bit.

This register is only accessible by TZ secure world software.

25.4.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TZ_WDT_CTL	Reserved			SPIDEN	SPNIDEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.10.4 Fields

Field	Function
31-6 —	- Reserved
5 TZ_WDT_CTL	TZ_WDT_CTL TrustZone Watchdog Control 0b - TZ WDOG timer operates independently of the debug halted state of the GPP cores (0 = default), AND RSTCNTL[RST_REQ_MSK] IS NOT allowed to block any device reset request 1b - TZ WDOG timer is halted when all GPP cores are in a debug halted state, AND RSTCNTL[RST_REQ_MSK] IS allowed to block any device reset request
4-2 —	- Reserved
1 SPIDEN	SPIDEN Global secure privileged invasive debug 0b - not enabled 1b - enabled
0 SPNIDEN	SPNIDEN Global secure privileged non-invasive debug 0b - not enabled 1b - enabled

25.4.3.11 TZPC Peripheral Identification Register 0 (TZPCPERIPHID0)

25.4.3.11.1 Offset

Register	Offset
TZPCPERIPHID0	FE0h

25.4.3.11.2 Function

Peripheral Identification (PID) registers contain TZPC IP block identity information. They are not operational registers.

25.4.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Partnumber0							
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

25.4.3.11.4 Fields

Field	Function
31-8 —	- Read undefined
7-0 Partnumber0	Partnumber0 These bits read back as 0x70

25.4.3.12 TZPC Peripheral Identification Register 1 (TZPCPERIPHID1)

25.4.3.12.1 Offset

Register	Offset
TZPCPERIPHID1	FE4h

25.4.3.12.2 Function

Peripheral Identification (PID) registers contain TZPC IP block identity information. They are not operational registers.

25.4.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Designer0				Partnumber1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

25.4.3.12.4 Fields

Field	Function
31-8 —	- Read undefined.
7-4 Designer0	Designer0 These bits read back as 0x1
3-0 Partnumber1	Partnumber1 These bits read back as 0x8

25.4.3.13 TZPC Peripheral Identification Register 2 (TZPCPERIPHID2)

25.4.3.13.1 Offset

Register	Offset
TZPCPERIPHID2	FE8h

25.4.3.13.2 Function

Peripheral Identification (PID) registers contain TZPC IP block identity information. They are not operational registers.

25.4.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Revision				Designer1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

25.4.3.13.4 Fields

Field	Function
31-8 —	- Read undefined
7-4 Revision	Revision These bits read back as the revision number which can be 0-15
3-0 Designer1	Designer1 These bits read back as 0x4

25.4.3.14 TZPC Peripheral Identification Register 3 (TZPCPERIPHID3)

25.4.3.14.1 Offset

Register	Offset
TZPCPERIPHID3	FECh

25.4.3.14.2 Function

Peripheral Identification (PID) registers contain TZPC IP block identity information. They are not operational registers.

25.4.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Configuration							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

25.4.3.14.4 Fields

Field	Function
31-8 —	- Read undefined
7-0 Configuration	Configuration These bits read back as 0x00

25.4.3.15 TZPC Identification Register 0 (TZPCPCELLID0)

25.4.3.15.1 Offset

Register	Offset
TZPCPCELLID0	FF0h

25.4.3.15.2 Function

The read-only TZPCPCELLID0 Register is hard-coded and the fields in the register determine the reset value.

25.4.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TZPCPCELLID0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

25.4.3.15.4 Fields

Field	Function
31-8 —	- Read undefined
7-0 TZPCPCELLID0	TZPCPCELLID0 These bits read back as 0x0D

25.4.3.16 TZPC Identification Register 1 (TZPCPCELLID1)

25.4.3.16.1 Offset

Register	Offset
TZPCPCELLID1	FF4h

25.4.3.16.2 Function

The read-only TZPCPCELLID1 Register is hard-coded and the fields in the register determine the reset value.

25.4.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TZPCPCELLID1							
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

25.4.3.16.4 Fields

Field	Function
31-8 —	- Read undefined
7-0 TZPCPCELLID1	TZPCPCELLID1 These bits read back as 0xF0

25.4.3.17 TZPC Identification Register 2 (TZPCPCELLID2)

25.4.3.17.1 Offset

Register	Offset
TZPCPCELLID2	FF8h

25.4.3.17.2 Function

The read-only TZPCPCELLID2 Register is hard-coded and the fields in the register determine the reset value.

25.4.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TZPCPCELLID2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

25.4.3.17.4 Fields

Field	Function
31-8 —	- Read undefined
7-0 TZPCPCELLID2	TZPCPCELLID2 These bits read back as 0x05

25.4.3.18 TZPC Identification Register 3 (TZPCPCELLID3)

25.4.3.18.1 Offset

Register	Offset
TZPCPCELLID3	FFCh

25.4.3.18.2 Function

The read-only TZPCPCELLID3 Register is hard-coded and the fields in the register determine the reset value.

25.4.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TZPCPCELLID3							
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

25.4.3.18.4 Fields

Field	Function
31-8 —	- Read undefined
7-0 TZPCPCELLID3	TZPCPCELLID3 These bits read back as 0x00

25.5 TrustZone Memory Adapter (TZMA)

25.5.1 About the AXI TrustZone memory adapter

The TZMA is an AMBA-compliant, SoC peripheral that is developed, tested, and licensed by Arm Limited.

The TZC-400 provides TZ secure world logical separate enforcement for main memory, the TZPC and TZMA provide this function for on-chip SRAM and other on-chip memory mapped registers. The TZMA sits on the bus in front of the access controlled targets and blocks TZ non-secure world accesses to TZ secure world addresses. TZMAs have no configuration registers of their own, and are directly connected to a TZPC which provides the configuration space and control.

The TZMA has the following features:

- it is compatible with the AXI internal memory interface (BP140)
- it has configuration inputs that can be driven from the TrustZone Protection Controller (TZPC) or tied off as required.

- it supports both a single active read and a single active write transaction
- it does not have an AXI low-power interface because it does not initiate transfers or have a power-down sequence
- it does not provide exclusive access monitoring
- TZMA parameters are configured as below:
  - data width of 64 bits
  - ID width is four bits
  - addressed memory size of up to 2MB

The following figure shows a TZMA configured in a typical TrustZone-enabled design.

Figure 25-7. Typical configuration

25.5.2 Functional description

The TZMA is described in:

- Functional blocks
- Interface attributes

25.5.2.1 Functional blocks

The figure below shows a functional block diagram of the TZMA. It contains the sub-blocks described in:

- Write channel router
- Read channel router
- Address comparator

Figure 25-8. AXI TrustZone memory adapter

25.5.2.1.1 Write channel router

Transactions presented at the AXI slave interface are checked against the current security configuration. If a transaction passes this security check then it is routed through to the AXI master interface. If it fails then it is rejected by routing to an internal default slave.

One write transaction is handled at a time. Further transactions are held off until outstanding transactions are completed. If there are no other transactions in progress then routing is set up as soon as AWVALID is asserted so that no unnecessary latency is introduced.

25.5.2.1.2 Read channel router

This operates in a similar manner to the write channel router. If an illegal transaction is detected then an internal default slave signals and generates a response.

25.5.2.1.3 Address comparator

This compares the input address and transaction protection information against the secure region size from the configuration inputs and the addressed memory size. It uses this information to determine if the address is legal.

25.5.2.2 Interface attributes

The master and slave interface attributes for the TZMA are described in Table 25-5 and Table 25-6.

Table 25-5. Master interface attributes

Attribute	Description	Value
Write issuing capability	The maximum number of active write transactions that a master can generate	1
Read issuing capability	The maximum number of active read transactions that a master can generate	1
Write ID capability	The maximum number of different AWID values that a master can generate for all active write transactions at any one time	1
Write ID width	The number of bits in the AWID and WID buses	Set by the ID_WIDTH parameter
Read ID capability	The maximum number of different ARID values that a master can generate for all active read transactions at any one time	1
Read ID width	The number of bits in the ARID bus	Set by the ID_WIDTH parameter

Table 25-6. Slave interface attributes

Attribute	Description	Value
Write acceptance capability	The maximum number of active write transactions that a slave can accept.	1
Read acceptance capability	The maximum number of active read transactions that a slave can accept.	1
Write interleave depth	The number of active write transactions for which the slave can receive data. It is counted from the earliest transaction.	1
Read data reorder depth	The number of active read transactions for which a slave may transmit data. It is counted from the earliest transaction.	1

Chapter 26
Trust Architecture 3.0

26.1 Trust architecture objectives

The LS1080A supports the QorIQ Trust Architecture 3.0.

The objective of the trust architecture is to allow OEMs to prevent or strongly mitigate an attacker's ability to achieve the following attacks:

- theft of functionality,
- theft of third-party data,
- and theft of uniqueness.

These attacks are against the OEM's complete system, including both hardware and software. These systems operate in a wide variety of contexts. In addition, attackers have a broad range of motivations and skill sets. It is beyond the scope of this document to provide an exhaustive list of threats, mitigations, and contexts in which attacks might occur.

Consequently, this section focuses on trust architecture claims. This approach allows the OEM to assess how to best apply the trust architecture to mitigate attacks in a given context.

26.2 Characteristics and claims

1. Trust architecture is "opt in."
    - OEMs do not need to do anything to disable it. All trust features are enabled as the result of conscious decisions to set fuses, program registers, and perform code signing.
    - TrustZone is not entirely opt-in, as Arm cores come out of reset in execution level (EL) 3—TZ Secure World Supervisor level.

    - Many SoC registers are only writeable when the CPU is executing in TZ Secure World.
    - Beyond basic configuration firmware which would be challenging to run outside of TZ Secure World, running security services in TZ Secure World is an OEM 'opt in' decision.
2. OEMs own all of the system's secrets.
    - OEMs are not dependent on NXP to provision devices, and NXP is not required to participate in the chain of trust for system manufacturing, deployment, or field servicing.
    - NXP is developing new software tools and services to support the chain of trust.
3. Trust architecture allows OEMs to define system security policies & configurations, which cannot be changed or bypassed by attackers.
    - Always perform secure boot
    - Access control debug interfaces
    - Detect hardware security violations
    - Soft or Hard Fail reactions to security violations
4. Trust architecture protects against unauthorized modifications to developer software and system configuration information (for example, device trees, certificates, and so on).
    - Protection consists of both prevention (secure boot) and after-the-fact detection mechanisms (for example, runtime integrity checking).
        - Secure boot detects unauthorized modifications and, when detected, prevents the unauthorized code from executing on the device.
        - Runtime integrity checking scans regions of memory for unauthorized modifications to the contents.
5. Trust architecture allows OEMs to provision system permanent secrets, which are protected against extraction or exposure.
    - Permanent secrets continue across resets of the system, and are locked out in response to security violations.
    - Trust architecture 3.0 permanent secrets include:
        - Debug Response Value (DRV)
        - One-Time Programmable Master Key (OTPMK)
6. Trust architecture allows OEMs to provision system persistent secrets which survive device resets under normal circumstances, but are capable of being cleared or rendered unusable.
    - Trust architecture 3.0 persistent secrets include:
        - Battery backed Zeroizable Master Key (ZMK)
        - Any cryptographic blobs created with the OTPMK and stored to NVRAM prior to system reset
    - Cryptographic blobs can be used to make factory installed keys, certificates, and proprietary code into persistent secrets.

7. Trust architecture allows provisioning of system ephemeral secrets, which are protected against extraction or exposure.
   - Ephemeral secrets are cleared by a security violation or system reset.
   - Trust architecture 3.0 ephemeral secrets include:
     - Job Descriptor Key Encryption Keys (JDKEKs)
     - Session keys negotiated during runtime, which are protected with the JDKEKs
     - Secrets owned by TZ Secure World trusted applications, and stored in private on-chip or off-chip memory are also considered ephemeral.
8. Trust architecture provides high levels of separation between independent software domains.
   - The private resources of one domain must not be accessible by another domain.
   - These independent software partitions can be treated as independent security domains.
   - Arm TrustZone adds "Secure World," another level of separation not found in Power Architecture-based Trust Architecture devices.
9. Trust architecture 3.0 supports anti-rollback features
   - Super Root Key revocation
   - Monotonic Counter

26.3 Non-claims

It is also important for OEMs to understand the types of attacks that NXP does not claim to prevent or strongly mitigate, so that mitigation can be taken at the system level if necessary.

1. Stronger than the underlying crypto algorithms.
   - Trust architecture's foundation rests on the strength of SHA-256, AES-256, and RSA digital signatures. If these algorithms are found to be fundamentally broken, most trust architecture claims are broken as well.
2. Preventing advanced physical attacks
   - As noted earlier, NXP recognizes that the contents of the Security Fuse Processor can be read by careful de-processing of the device. This attack destroys the QorIQ Layerscape device, however cryptographic blobs protected with the OTPMK could be recovered from system memory and decrypted.
   - Memories and registers locked out or zeroized upon detection of security violations may be subject to memory remanence attacks.
   - The SEC's PKHA implements timing equalization, and the AESA implements differential power analysis resistance features; however, the effectiveness of these features has not been independently evaluated. Other side channel attack

methods, against the SEC and CPUs, may be possible given sufficient motivation.
- Operating the device outside of specified voltage, temperature, or frequency ranges can lead to unexpected failure modes. NXP has attempted to make all failure modes "fail safe," but the full range of possible glitches has not been evaluated.
- Trust architecture incorporates tamper detection inputs, which OEMs can use to alert the device to physical attacks. Without using the Zeroizable Master Key, the highest consequence of a security violation is that the OTPMK may be locked out. Assume that attackers are capable of bypassing external tamper detection logic and rebooting the system with no hardware security violations detected.

3. Providing absolute isolation between independent software domains/security domains
   - Trust architecture's support for strong partitioning is based on access control mechanisms. If a resource is private to domain 1, domain 2 must not be able to access that resource, directly or indirectly. This is different from absolute isolation, in which domain 2 is unable to interfere with the operation of domain 1. From internal and external bus bandwidth consumption to unfiltered buffer releases, NXP knows of many scenarios in which domain 2 can interfere with domain 1.
4. Operating as a single-edged sword
   - Trust architecture is designed to constrain the conditions under which the system can operate. Attackers can cause security violations for the purpose of shutting down the system. Trust architecture offers configuration options allowing OEMs to make trade-offs in security violation sensitivity and consequences, but it is beyond the scope of a trust architecture device to "know" when trust is being exploited to disrupt the system. Attackers in a position to trigger security violations could be capable of causing other types of system failures.

26.4 Related resources

QorIQ Trust Architecture is not a mandatory feature; it is opt in. To minimize the information available to potential attackers, access to detailed Trust Architecture reference material is restricted to vetted customers with active programs using Trust Architecture. Contact your NXP Sales or Field Application Engineer to initiate the admission process to the NXP QorIQ Trust Architecture User’s Group site.

The following table shows related resources that may be helpful.

Table 26-1. Related resources

Resource	Purpose
An Introduction to the QorIQ Platform's Trust Architecture (White Paper)	Discusses the objectives of the trust architecture, and how it works. This white paper also includes logistical considerations.
QorIQ Trust Architecture 3.0 User Guide	Primary technical reference for Trust Architecture hardware blocks, including register level detail, as well as procedural steps to enable secure boot and security violation detection.
User Enablement for Secure Boot	These documents are periodically updated as new features are added to the Code Signing Tool and reference chain of trust. They can be found in the QorIQ device SDK, and at www.nxp.com/infocenter: Software and Tools Information Center > QorIQ SDK Documentation > Secure Boot. **NOTE:** Some of these documents are for PBL-based products, some are for non-PBL-based products. This chip is a PBL-based product.

Chapter 27
Thermal Monitoring Unit (TMU)

27.1 The TMU module as implemented on the chip

This section provides details about how the TMU module is implemented on the chip.

NOTE

- The maximum operating temperature of the chip should not exceed that specified in the data sheet. For more information, refer the data sheet.
- Accuracy within ±5°C.

27.1.1 Local temperature sensor placement

The table below shows the placement of the temperature sensors.

Table 27-1. Local Temperature Sensor Placement in Chip

Temperature Sensor ID	Placement
0	A53 Core Clusters
1	SoC Platform
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved

Table continues on the next page...

Table 27-1. Local Temperature Sensor Placement in Chip (continued)

Temperature Sensor ID	Placement
13	Reserved
14	Reserved
15	Reserved

27.1.2 Initialization Information

The TMU calculates temperature by reading one of the temperature sensor sites on the chip. The temperature is calculated from a calibration table, which is programmed either by the pre-boot loader or initialization of calibration table through software. Failure to properly initialize the calibration table may lead to an undefined behavior. Calibration determines specific sensor reading translated in degrees Celsius as shown in the following tables.

Register	Value	Comment
TTR0CR	***0x000B_0000***	***12 points @ 0C***
TTCFGR	0x0000_0000	Point 0 @ 0C
TSCFGR	0x0000_0023	
TTCFGR	0x0000_0001	Point 1 @ 4C
TSCFGR	0x0000_002A	
TTCFGR	0x0000_0002	Point 2 @ 8C
TSCFGR	0x0000_0030	
TTCFGR	0x0000_0003	Point 3 @ 12C
TSCFGR	0x0000_0037	
TTCFGR	0x0000_0004	Point 4 @ 16C
TSCFGR	0x0000_003D	
TTCFGR	0x0000_0005	Point 5 @ 20C
TSCFGR	0x0000_0044	
TTCFGR	0x0000_0006	Point 6 @ 24C
TSCFGR	0x0000_004A	
TTCFGR	0x0000_0007	Point 7 @ 28C
TSCFGR	0x0000_0051	
TTCFGR	0x0000_0008	Point 8 @ 32C
TSCFGR	0x0000_0057	
TTCFGR	0x0000_0009	Point 9 @ 36C
TSCFGR	0x0000_005E	
TTCFGR	0x0000_000A	Point 10 @ 40C
TSCFGR	0x0000_0064	

Table continues on the next page...

TTCFGR	0x0000_000B	Point 11 @ 44C
TSCFGR	0x0000_006B	
TTR1CR	***0x0009_002A***	***10 points @ 42C***
TTCFGR	0x0001_0000	Point 0 @ 42C
TSCFGR	0x0000_0022	
TTCFGR	0x0001_0001	Point 1 @ 46C
TSCFGR	0x0000_002A	
TTCFGR	0x0001_0002	Point 2 @ 50C
TSCFGR	0x0000_0032	
TTCFGR	0x0001_0003	Point 3 @ 54C
TSCFGR	0x0000_003A	
TTCFGR	0x0001_0004	Point 4 @ 58C
TSCFGR	0x0000_0042	
TTCFGR	0x0001_0005	Point 5 @ 62C
TSCFGR	0x0000_004A	
TTCFGR	0x0001_0006	Point 6 @ 66C
TSCFGR	0x0000_0052	
TTCFGR	0x0001_0007	Point 7 @ 70C
TSCFGR	0x0000_005A	
TTCFGR	0x0001_0008	Point 8 @ 74C
TSCFGR	0x0000_0062	
TTCFGR	0x0001_0009	Point 9 @ 78C
TSCFGR	0x0000_006A	
TTR2CR	***0x0006_004C***	***7 points @ 76C***
TTCFGR	0x0002_0000	Point 0 @ 76C
TSCFGR	0x0000_0021	
TTCFGR	0x0002_0001	Point 1 @ 80C
TSCFGR	0x0000_002B	
TTCFGR	0x0002_0002	Point 2 @ 84C
TSCFGR	0x0000_0035	
TTCFGR	0x0002_0003	Point 3 @ 88C
TSCFGR	0x0000_0040	
TTCFGR	0x0002_0004	Point 4 @ 92C
TSCFGR	0x0000_004A	
TTCFGR	0x0002_0005	Point 5 @ 96C
TSCFGR	0x0000_0054	
TTCFGR	0x0002_0006	Point 6 @ 100C
TSCFGR	0x0000_005E	
TTR3CR	***0x0007_0062***	***4 points @ 98C***
TTCFGR	0x0003_0000	Point 0 @ 98C
TSCFGR	0x0000_0010	
TTCFGR	0x0003_0001	Point 1 @ 102C

Table continues on the next page...

TSCFGR	0x0000_001C	
TTCFGR	0x0003_0002	Point 2 @ 106C
TSCFGR	0x0000_0027	
TTCFGR	0x0003_0003	Point 3 @ 110C
TSCFGR	0x0000_0032	

The process for programming the calibration table is described below:

1. Disable the monitoring mode, TMR[ME]=0 (default).
2. Program TMU_TTR0CR=0x000B0000, TMU_TTR1CR=0x0009002A, TMU_TTR2CR=0x0006004C and TMU_TTR3CR=0x00070062.
3. Write the temperature configuration register (TTCFGR).
4. Write the sensor configuration register (TSCFGR).
5. Repeat steps 2-3 for all temperatures as defined in the above table.

27.2 Thermal Monitoring Unit Introduction

The Thermal Monitoring Unit (TMU) monitors and reports the temperature from one or more remote temperature measurement sites located on chip.

27.2.1 TMU Overview

The TMU has access to multiple temperature measurement sites strategically located on the chip. It monitors these sites and can signal an alarm if a programmed threshold is ever exceeded. The upper and lower temperature range is continuously captured. A set of reporting registers allow for reading the current temperature at monitored sites.

Figure 27-1. Thermal Monitoring Unit Block Diagram

27.2.2 Features

The temperature management unit features:

- Temperature measurement range 0-110°C.
- Calibration
  - Calibration table loaded from boot code ROM using pre-boot loader *or* initialization of calibration table through software
- Monitoring
  - Single-, or multi-site monitoring
  - Programmable monitoring interval
  - Out-of-range indication
  - High/low temperature range monitoring
  - Immediate and average temperature monitoring

  - Average temperature monitoring programmable low-pass filtering
  - Programmable monitoring thresholds for normal and critical alarm
- Reporting
  - Immediate and average temperature reporting for all monitor sites

27.2.3 Modes of Operation

The TMU has one mode of operation:

- Monitoring

The mode register monitoring enable bit, TMR[ME], determines if the unit is in active monitoring mode or in power saving mode.

The table below describes bit settings required for each TMU mode of operation.

Table 27-2. TMU Mode Bit Setting

Modes with Features	TMR[ME]
Monitoring mode disabled	0
Monitoring mode enabled	1

27.3 TMU register descriptions

The table shows the memory map for management of the TMU resources.

27.3.1 TMU memory map

TMU.TMU1 base address: 1F8_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	TMU mode register (TMR)	32	RW	0000_0000h
4h	TMU status register (TSR)	32	RO	0000_0000h
8h	TMU monitor temperature measurement interval register (TMTMIR)	32	RW	0000_0000h
20h	TMU interrupt enable register (TIER)	32	RW	0000_0000h
24h	TMU interrupt detect register (TIDR)	32	W1C	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
28h	TMU interrupt site capture register (TISCR)	32	RW	0000_0000h
2Ch	TMU interrupt critical site capture register (TICSCR)	32	RW	0000_0000h
40h	TMU monitor high temperature capture register (TMHTCR)	32	RO	0000_0000h
44h	TMU monitor low temperature capture register (TMLTCR)	32	RO	0000_0000h
50h	TMU monitor high temperature immediate threshold register (TMHTITR)	32	RW	0000_0000h
54h	TMU monitor high temperature average threshold register (TMHTATR)	32	RW	0000_0000h
58h	TMU monitor high temperature average critical threshold register (TMHTACTR)	32	RW	0000_0000h
80h	TMU temperature configuration register (TTCFGR)	32	RW	0000_0000h
84h	TMU sensor configuration register (TSCFGR)	32	RW	0000_0000h
100h	TMU report immediate temperature site register 0 (TRITSR0)	32	RO	0000_0000h
104h	TMU report average temperature site register 0 (TRATSR0)	32	RO	0000_0000h
110h	TMU report immediate temperature site register 1 (TRITSR1)	32	RO	0000_0000h
114h	TMU report average temperature site register 1 (TRATSR1)	32	RO	0000_0000h
F10h	TMU temperature range 0 control register (TTR0CR)	32	RW	000B_0000h
F14h	TMU temperature range 1 control register (TTR1CR)	32	RW	000A_0026h
F18h	TMU temperature range 2 control register (TTR2CR)	32	RW	0008_0048h
F1Ch	TMU temperature range 3 control register (TTR3CR)	32	RW	0007_0061h

27.3.2 TMU mode register (TMR)

27.3.2.1 Offset

Register	Offset
TMR	0h

27.3.2.2 Function

The TMU mode register allows software to control the operation of the thermal monitoring.

27.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ME	Reserved			ALPF		Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSITE		Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.2.4 Fields

Field	Function
31 ME	Monitoring mode enable. 0 No monitoring. 1 Monitoring of sites as defined by field MSITE. Before enabling the TMU for monitoring, the TMU must be configured, see section Initialization Information. Failure to properly initialize the configuration table may result in boundedly undefined behavior.
30-28 —	Reserved
27-26 ALPF	Average low pass filter setting. 00 1.0 01 0.5 10 0.25 11 0.125 The average temperature is calculated as: ALPF x Current_Temp + (1 - ALPF) x Average_Temp. If no previous (average) temperature is valid, current temperature is used. For proper operation, this field should only change when monitoring is disabled.
25-16 —	Reserved
15-14 MSITE	Monitoring site select 0 - 1. By setting the select bit for a temperature sensor site, it is enabled and included in all monitoring functions. For proper operation, this field should only change when monitoring is disabled. If no site is selected, site 0 is monitored by default.
13-0 —	Reserved

27.3.3 TMU status register (TSR)

27.3.3.1 Offset

Register	Offset
TSR	4h

27.3.3.2 Function

The TMU status register reports the monitoring and calibration status during operation.

27.3.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	MIE	ORL	ORH	Reserved											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.3.4 Fields

Field	Function
31 —	Reserved
30 MIE	Monitoring interval exceeded. 0 Monitoring interval not exceeded. 1 Monitoring interval exceeded. The time required to perform measurement of all monitored sites has exceeded the monitoring interval as defined by TMTMIR. This bit will clear automatically when TMU monitoring is (re-)enabled or the monitoring interval register, TMTMIR, is written.

Table continues on the next page...

Field	Function
29 ORL	Out-of-range low temperature measurement detected. A temperature sensor detected a temperature reading below the lowest measurable temperature of 0 degrees Celsius. This bit will clear automatically when TMU monitoring is (re-)enabled.
28 ORH	Out-of-range high temperature measurement detected. A temperature sensor detected a temperature reading above the highest measurable temperature of 110 °C. This bit will clear automatically when TMU monitoring is (re-)enabled.
27-0 —	Reserved

27.3.4 TMU monitor temperature measurement interval register (TMTMIR)

27.3.4.1 Offset

Register	Offset
TMTMIR	8h

27.3.4.2 Function

The TMU monitor temperature measurement interval register determines at what frequency temperature sensors are read. All enabled monitored sites are read once in the duration of the time interval. The status bit TSR[MIE] will be set if the temperature measument takes longer than the set interval. Software should consider increasing the interval or reducing the number of active sites if the interval is exceeded. Disabling the interval allows for continuous monitoring.

NOTE

The time it takes for one temperature measurement is dependent on the temperatures measured and mode settings, thus there is no fixed time advertised for a single temperature measurements.

27.3.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												TMI			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.4.4 Fields

Field	Function
31-4 —	Reserved
3-0 TMI	Temperature monitoring interval in seconds. For proper operation, this field should only change when monitoring is disabled, TMR[ME]=0. For lower platform speeds, the time increases proportionally, while the opposite is true for higher platform speeds.

Setting	Platform Clock Frequency			
	333 MHz	400 MHz	667 MHz	800 MHz
0000	0.03 s	0.02 s	0.015 s	0.01 s
0001	0.06 s	0.04 s	0.03 s	0.02 s
0010	0.10 s	0.08 s	0.05 s	0.04 s
0011	0.20 s	0.17 s	0.10 s	0.08 s
0100	0.40 s	0.34 s	0.20 s	0.17 s
0101	0.80 s	0.67 s	0.40 s	0.34 s
0110	1.60 s	1.34 s	0.80 s	0.67 s
0111	3.2 s	2.7 s	1.6 s	1.34 s
1000	6.4 s	5.4 s	3.2 s	2.7 s
1001	12.8 s	10.7 s	6.4 s	5.4 s
1010	25.8 s	21.5 s	12.9 s	10.7 s
1011	51.6 s	42.9 s	25.8 s	21.5 s
1100	103 s	85.9 s	51.5 s	42.9 s
1101	206.0 s	171.8 s	103.0 s	85.9 s
1110	412.2 s	343.6 s	206.1 s	171.8 s
1111	Disabled			

27.3.5 TMU interrupt enable register (TIER)

27.3.5.1 Offset

Register	Offset
TIER	20h

27.3.5.2 Function

The TMU interrupt enable register determines if a detected status condition should cause a system interrupt. A system interrupt occurs if a bit in this register is set and the corresponding bit in the interrupt detect register is also set. To clear the interrupt, write a 1 to the interrupt detect register.

27.3.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ITTEIE	ATTEIE	ATCTEIE	Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.5.4 Fields

Field	Function
31 ITTEIE	Immediate temperature threshold exceeded interrupt enable. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ITTE] is set.

Table continues on the next page...

Field	Function
30 ATTEIE	Average temperature threshold exceeded interrupt enable. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATTE] is set.
29 ATCTEIE	Average temperature critical threshold exceeded interrupt enable. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATCTE] is set.
28-0 —	Reserved

27.3.6 TMU interrupt detect register (TIDR)

27.3.6.1 Offset

Register	Offset
TIDR	24h

27.3.6.2 Function

The TMU interrupt detect register indicates if an status condition was detected that could generate an interrupt. Write 1 to clear the detected condition and the interrupt, if enabled.

27.3.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ITTE	ATTE	ATCTE	Reserved												
W	W1C	W1C	W1C													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.6.4 Fields

Field	Function
31 ITTE	Immediate temperature threshold exceeded. Write 1 to clear. 0 No threshold exceeded. 1 Immediate temperature threshold, as defined by TMHTITR, has been exceeded by one or more monitored sites. This includes an out-of-range measured temperature above 110 °C. The sites which has exceeded the threshold are captured in TISCR[ISITE].
30 ATTE	Average temperature threshold exceeded. Write 1 to clear. 0 No threshold exceeded. 1 Average temperature threshold, as defined by TMHTATR, has been exceeded by one or more monitored sites. The sites which has exceeded the threshold are captured in TISCR[ASITE].
29 ATCTE	Average temperature critical threshold exceeded. Write 1 to clear. 0 No threshold exceeded. 1 Average temperature critical threshold, as defined by TMHTACTR, has been exceeded by one or more monitored sites. The sites which has exceeded the threshold are captured in TICSCR[CASITE].
28-0 —	Reserved

27.3.7 TMU interrupt site capture register (TISCR)

27.3.7.1 Offset

Register	Offset
TISCR	28h

27.3.7.2 Function

The TMU interrupt site capture register holds information about the temperature sensor site associated with a detected interrupt event.

27.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ISITE		Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ASITE		Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.7.4 Fields

Field	Function
31-30 ISITE	Temperature sensor site associated with the setting of TIDR[ITTE]. This field has the same bit representation as TMR[MSITE]. Software should clear this field after handling the detected interrupt events ITTE.
29-16 —	Reserved
15-14 ASITE	Temperature sensor site associated with the setting of TIDR[ATTE] . This field has the same bit representation as TMR[MSITE]. Software should clear this field after handling the detected interrupt event ATTE.
13-0 —	Reserved

27.3.8 TMU interrupt critical site capture register (TICSCR)

27.3.8.1 Offset

Register	Offset
TICSCR	2Ch

27.3.8.2 Function

The TMU interrupt critical site capture register holds information about the temperature sensor site associated with a detected critical interrupt event.

27.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CASITE		Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.8.4 Fields

Field	Function
31-16 —	Reserved
15-14 CASITE	Temperature sensor site associated with the setting of TIDR[ATCTE] . This field has the same bit representation as TMR[MSITE]. Software should clear this field after handling the detected critical interrupt event ATCTE.
13-0 —	Reserved

27.3.9 TMU monitor high temperature capture register (TMHTCR)

27.3.9.1 Offset

Register	Offset
TMHTCR	40h

27.3.9.2 Function

This TMU monitor register captures and record the highest temperature reached for any one enabled monitored site within temperature sensor range.

27.3.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TEMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.9.4 Fields

Field	Function
31 V	Valid reading. 0 Temperature reading is not valid due to no measured temperature within the sensor range of 0-110 °C for an enabled monitored site. 1 Temperature reading is valid. (Re-)enabling the TMU will automatically clear this bit and start a new search.
30-8 —	Reserved
7-0 TEMP	Highest temperature recorded in degrees Celcius by any enabled monitored site. Valid when V=1. 0-110 °C Sensor range 111-255 °C Reserved

27.3.10 TMU monitor low temperature capture register (TMLTCR)

27.3.10.1 Offset

Register	Offset
TMLTCR	44h

27.3.10.2 Function

This TMU monitor register captures and record the lowest temperature reached for any one enabled monitored site within temperature sensor range.

27.3.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TEMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.10.4 Fields

Field	Function
31 V	Valid reading. 0 Temperature reading is not valid due to no measured temperature within the sensor range of 0-110 °C for an enabled monitored site. 1 Temperature reading is valid. (Re-)enabling the TMU will automatically clear this bit and start a new search.
30-8 —	Reserved
7-0 TEMP	Lowest temperature recorded in degrees Celcius by any enabled monitored site. Valid when V=1. 0-110 °C Sensor range 111-255 °C Reserved

27.3.11 TMU monitor high temperature immediate threshold register (TMHTITR)

27.3.11.1 Offset

Register	Offset
TMHTITR	50h

27.3.11.2 Function

This TMU monitor register determines the high current temperature threshold for generating the TIDR[ITTE] event.

27.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.11.4 Fields

Field	Function
31 EN	Enable threshold. 0 Disabled. 1 Threshold enabled.
30-8 —	Reserved

Table continues on the next page...

Field	Function
7-0 TEMP	High temperature immediate threshold value. Determines the current upper temperature threshold, for any enabled monitored site, that if exceeded will cause TIDR[ITTE] to be set when EN=1. 0-110 °C Sensor range 111-255 °C Reserved

27.3.12 TMU monitor high temperature average threshold register (TMHTATR)

27.3.12.1 Offset

Register	Offset
TMHTATR	54h

27.3.12.2 Function

This TMU monitor register determines the high average temperature threshold for generating the TIDR[ATTE] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating average temperature.

27.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.12.4 Fields

Field	Function
31 EN	Enable threshold. 0 Disabled. 1 Threshold enabled.
30-8 —	Reserved
7-0 TEMP	High temperature average threshold value. Determines the average upper temperature threshold, for any enabled monitor site, that if exceeded will cause TIDR[ATTE] to be set when EN=1. 0-110 °C Sensor range 111-255 °C Reserved

27.3.13 TMU monitor high temperature average critical threshold register (TMHTACTR)

27.3.13.1 Offset

Register	Offset
TMHTACTR	58h

27.3.13.2 Function

This TMU monitor register determines the high average critical temperature threshold for generating the TIDR[ATCTE] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating average temperature.

27.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.13.4 Fields

Field	Function
31 EN	Enable threshold. 0 Disabled. 1 Threshold enabled.
30-8 —	Reserved
7-0 TEMP	High temperature average critical threshold value. Determines the average upper critical temperature threshold, for any enabled monitored site, that if exceeded will cause TIDR[ATCTE] to be set when EN=1. 0-110 °C Sensor range 111-255 °C Reserved

27.3.14 TMU temperature configuration register (TTCFGR)

27.3.14.1 Offset

Register	Offset
TTCFGR	80h

27.3.14.2 Function

The TMU temperature configuration register, in conjunction with the sensor configuration register, is used to initialize the internal sensor translation table used during monitoring. This register pair defines indirect access to the table. See the section Initialization Information.

27.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.14.4 Fields

Field	Function
31-0 DATA	Sensor data.

27.3.15 TMU sensor configuration register (TSCFGR)

27.3.15.1 Offset

Register	Offset
TSCFGR	84h

27.3.15.2 Function

The TMU sensor configuration register, in conjunction with the temperature configuration register, is used to initialize the internal sensor translation table used during monitoring. This register pair defines indirect access to the table. Reading this register will return the data from the translation table as defined by TTCFGR. See the section Initialization Information.

27.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.15.4 Fields

Field	Function
31-0 DATA	Sensor data.

27.3.16 TMU report immediate temperature site register a (TRITSR0 - TRITSR1)

27.3.16.1 Offset

Register	Offset
TRITSR0	100h
TRITSR1	110h

27.3.16.2 Function

This TMU report register returns the last measured temperature at site. The site must be part of the list of enabled monitored sites as defined by TMR[MSITE].

27.3.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TEMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.16.4 Fields

Field	Function
31 V	Valid measured temperature. 0 Not valid. Temperature out of sensor range or first measurement still pending. 1 Valid.
30-8 —	Reserved
7-0 TEMP	Last temperature reading at site when V=1.

27.3.17 TMU report average temperature site register a (TRATSR0 - TRATSR1)

27.3.17.1 Offset

Register	Offset
TRATSR0	104h
TRATSR1	114h

27.3.17.2 Function

This TMU report register returns the average measured temperature at site. The site must be part of the list of enabled monitored sites as defined by TMR[MSITE].

27.3.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	V	Reserved														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TEMP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.17.4 Fields

Field	Function
31 V	Valid measured temperature. 0 Not valid. Temperature out of sensor range or first measurement still pending. 1 Valid.
30-8 —	Reserved
7-0 TEMP	Average temperature reading at site when V=1.

27.3.18 TMU temperature range 0 control register (TTR0CR)

27.3.18.1 Offset

Register	Offset
TTR0CR	F10h

27.3.18.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 27-3. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

27.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

27.3.18.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8 —	Reserved
7-0 TEMP	Starting temperature in Celsius for range.

27.3.19 TMU temperature range 1 control register (TTR1CR)

27.3.19.1 Offset

Register	Offset
TTR1CR	F14h

27.3.19.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 27-4. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12

Table continues on the next page...

Table 27-4. Default Temperature Configuration Points (continued)

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

27.3.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0

27.3.19.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8 —	Reserved
7-0 TEMP	Starting temperature in Celsius for range.

27.3.20 TMU temperature range 2 control register (TTR2CR)

27.3.20.1 Offset

Register	Offset
TTR2CR	F18h

27.3.20.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 27-5. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

27.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

27.3.20.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8 —	Reserved
7-0 TEMP	Starting temperature in Celsius for range.

27.3.21 TMU temperature range 3 control register (TTR3CR)

27.3.21.1 Offset

Register	Offset
TTR3CR	F1Ch

27.3.21.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 27-6. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12

Table continues on the next page...

Table 27-6. Default Temperature Configuration Points (continued)

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

27.3.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

27.3.21.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8 —	Reserved
7-0 TEMP	Starting temperature in Celsius for range.

27.4 Functional Description

The following sections describe the functionality of the TMU in details.

27.4.1 Monitoring

Monitoring is the process of reading enabled temperature sensor sites on-chip at regular intervals and taking appropriate actions, such as alarming the user when the temperature exceeds a programmed temperature threshold.

During monitoring, all enabled temperature sensor sites are periodically measured for temperature starting with site 0 and ending with site 1. The interval at which the sensors are read is set in TMTMIR and should reflect the maximum interval time required to accurately capture temperature changes. If the measurement interval has not expired after the last active site has been read, the sensor logic enters low-power mode. If the interval has expired when the last active site is read, the measurement interval exceeded bit, TSR[MIE], is set and the next active site is read immediately. If the interval has been exceeded, user may opt to reduce number of sites monitored or increase the interval, if possible.

For each site the current and average temperature is logged. The average temperature is calculated based on the low-pass filter function in the mode register. If any of the set temperature threshold registers are exceeded, the corresponding interrupt detect bit is set in TIDR. Interrupts are enabled through the interrupt enable register, TIER.

Process for enabling monitoring mode:

1. Clear the interrupt detect register, TIDR.
2. Clear interrupt site capture register (TISCR) and interrupt critical site capture register (TICSCR).
3. Enable interrupt handling by setting the appropriate bits in TIER.
4. Set the temperature threshold registers TMHTITR, TMHTATR and TMHTACTR.
5. Set the monitoring interval register, TMTMIR.
6. Enable monitor mode by setting TMR[ME]=1. Sites to monitor are controlled by setting TMR[MSITE]. There should be at least one active site enabled. Set other mode control bits as needed.
7. If the monitoring interval is too short as indicated by TSR[MIE], the interval may need to be increased or number of sites reduced.

27.4.2 Reporting

The TMU can directly report the current and average temperature for a particular temperature sensor site during monitoring mode by reading one of the report registers per site. The report uses the last measurement done by the monitoring process and requires a site to be actively monitored for accurate reading. Reading a site which has last measured an invalid temperature outside the sensor range of 0-110 degrees Celsius, will have the valid bit cleared.

If monitoring is disabled, the last temperature measurement remains for the site(s) previously monitored and can still be read using the report registers knowing that the temperature reported is no longer accurate. This method can be used to capture the temperature at multiple sites in time, but does not allow for continuous monitoring.

Chapter 28
Universal Serial Bus Interface 3.0

28.1 Overview

The USB module is a USB 3.0-compliant serial interface engine for implementing a USB interface. This module may be connected to an external port. Collectively the module and external port are called the USB 3.0 interface. USB 3.0 supports super-speed (SS), high-speed (HS), full-speed (FS), and low-speed (LS) operations.

- The upper layer is common for USB 2.0 and USB 3.0 operation. This has the bus interface, buffer management block, list processor for scheduling, and control and status register (CSR) functions
- USB 2.0 PHY and MAC layers
- USB 3.0 PHY, LINK, and MAC layers

Figure 28-1. USB interface block diagram

28.1.1 Features

The USB 3.0 module includes the following features:

- Complies with USB specification rev 3.0 (xHCI compatible)
- Supports operation as a standalone USB host controller
- USB dual-role operation and can be configured as host or device
- Super-speed (5 Gbit/s), high-speed (480 Mbit/s), full-speed (12 Mbit/s), and low-speed (1.5 Mbit/s) operations.
- Supports operation as a standalone single port USB
- Supports four programmable, bidirectional USB endpoints
- OTG (on-the-go) 2.0 compliant, which includes both device and host capability. Super-speed operation is not supported when OTG is enabled.
- Supports system memory interface with 40-bit addressing capability

28.1.2 Modes of Operation

The USB 3.0 module operates in following modes.

- Host Mode: SS/HS/FS/LS
- Device Mode: SS/HS/FS
- OTG: HS/FS/LS

28.1.3 External Signals

This section contains detailed descriptions of all the USB 3.0 controller signals. Many of the signals for the PHY interfaces are muxed onto the same pins in order to reduce pin count. The following table shows the USB signals, indicating which interface supports each signal.

Table 28-1. USB 3.0 External Signals

Signal	I/O	Description
USB1_D_P	IO	USB PHY Data Plus
USB1_D_M	IO	USB PHY Data Minus
USB1_VBUS	I	USB PHY VBUS
USB1_ID	IO	USB PHY ID Detect
USB1_TX_P	O	USB PHY 3.0 Transmit Data (positive)
USB1_TX_M	O	USB PHY 3.0 Transmit Data (negative)
USB1_RX_P	I	USB PHY 3.0 Receive Data (positive)
USB1_RX_M	I	USB PHY 3.0 Receive Data (negative)

Table continues on the next page...

Table 28-1. USB 3.0 External Signals (continued)

Signal	I/O	Description
USB1_RESREF	IO	USB PHY Impedance Calibration
USB1_DRVVBUS	O	VBus power enable.
USB1_PWRFAULT	I	Indicates that a Vbus fault has occurred.

28.2 USB3.0 register descriptions

USB3.0 registers are 32-bit wide, and the addresses are 32-bit block aligned. To avoid hardware or software incompatibility, the driver must access the registers as 32-bit units. The driver must not access these registers as 8 bit or 16 bit. CSRs are classified as follows:

Table 28-2. Registers

Control and status segisters	Reference
Global registers	Capability registers length and HC interface version number (CAPLENGTH) to Global frame length adjustment register (GFLADJ) **NOTE:** The default value of ref_clk is 125 MHz. Refer to Reset, Clocking, and Initialization chapter for chip-specific value of ref_clk.
Device register	Device configuration register (DCFG) to Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_0 - DEPCMDPAR0_7)
xHCI host register	See xHCI Spec for details. Refer Table 28-3 for list of registers.

The following table lists the supported xHCI registers. For register definitions, refer to the xHCI specification.

Table 28-3. Host registers

Offset	Register	Offset	Register
Register Set's Base- USB1: 0x310_0000 USB2: 0x311_0000		Register Set's Base= DBOFF	
0x0000	CAPLENGTH	0x0000	DB
0x0004	HCSPARAMS1	Register Set's Base= xECP * 4	
0x0008	HCSPARAMS2	0x0000	USBLEGSUP
0x000C	HCSPARAMS3	0x0004	USBLEGCTLSTS
0x0010	HCCPARAMS	Addr1	
0x0014	DBOFF	0x0000	SUPTPRT2_DW0

Table continues on the next page...

Table 28-3. Host registers (continued)

Offset	Register	Offset	Register
0x0018	RTSOFF	0x0004	SUPTPRT2_DW1
0x001C	Reserved	0x0008	SUPTPRT2_DW2
0x0020	USBCMD	0x000C	SUPTPRT2_DW3
0x0024	USBSTS	Addr2	
0x0028	PAGESIZE	0x0000	SUPTPRT3_DW0
0x002C-0x0033	Reserved	0x0004	SUPTPRT3_DW1
0x0034	DNCTRL	0x0008	SUPTPRT3_DW2
0x0038	CRCR	0x000C	SUPTPRT3_DW3
0x003C	CRCR	Addr2 + 10h	
0x0040-0x004F	Reserved	0x0000	DCID
0x0050	DCBAAP	0x0004	DCDB
0x0054	DCBAAP	0x0008	DCERSTSZ
0x0058	CONFIG	0x000C	Reserved
0x005C-0x041F	Reserved	0x0010	DCERSTBA
0x0420	PORTSC PORTPMSC_SS1	0x0014	DCERSTBA
0x0424	PORTPMSC_201	0x0018	DCERDP
0x0428	PORTLI	0x001C	DCERDP
0x042C	PORTHLPMC_SS2	0x0020	DCCTRL
0x042C	PORTHLPMC_202	0x0024	DCST
Register Set's Base= RTSOFF		0x0028	DCPORTSC
0x0000	MFINDEX	0x002C	Reserved
0x0004	Reserved	0x0030	DCCP
Register Set's Base= RTSOFF + 20h		0x0034	DCCP
0x0000	IMAN	0x0038	DCDDI1
0x0004	IMOD	0x003C	DCDDI2
0x0008	ERSTSZ		
0x000C	Reserved		
0x0010	ERSTBA		
0x0014	ERSTBA		
0x0018	ERDP		
0x001C	ERDP		

28.2.1 USB3 memory map

USB1 base address: 310_0000h

USB2 base address: 311_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Capability registers length and HC interface version number (CAPLENGTH)	32	RO	0100_0020h
4h	Host controller structural parameters 1 (HCSPARAMS1)	32	RO	0200_017Fh
8h	Host controller structural parameters 2 (HCSPARAMS2)	32	RO	1400_00F1h
Ch	Host controller structural parameters 3 (HCSPARAMS3)	32	RO	07FF_000Ah
10h	Host controller capability parameters 1 (HCCPARAMS1)	32	RO	0220_F66Dh
14h	Doorbell offset (DBOFF)	32	RO	0000_0480h
18h	Runtime register space offset (RTSOFF)	32	RO	0000_0440h
1Ch	Host controller capability parameters 2 (HCCPARAMS2)	32	RO	0000_000Bh
C100h	Global SoC bus configuration register 0 (GSBUSCFG0)	32	RW	0010_0080h
C104h	Global SoC bus configuration register 1 (GSBUSCFG1)	32	RW	0000_0700h
C108h	Global Tx threshold control register (GTXTHRCFG)	32	RW	0000_0000h
C10Ch	Global Rx threshold control register (GRXTHRCFG)	32	RW	0000_0000h
C110h	Global core control register (GCTL)	32	RW	30C1_3004h
C118h	Global status register (GSTS)	32	RW	3E80_0000h
C11Ch	Global user control register 1 (GUCTL1)	32	RW	0004_018Ah
C128h	Global user ID register (GUID)	32	RW	0130_290Ah
C12Ch	Global user control register (GUCTL)	32	RW	0200_8010h
C130h	Global SoC bus error address register low (GBUSERRADDRLO)	32	RO	0000_0000h
C134h	Global SoC bus error address register high (GBUSERRADDRHI)	32	RO	0000_0000h
C138h	Global SS port to bus instance mapping register - low (GPRTBIMAPLO)	32	RW	0000_0000h
C13Ch	Global SS port to bus instance mapping register - high (GPRTBIMAPHI)	32	RW	0000_0000h
C140h	Global hardware parameters register 0 (GHWPARAMS0)	32	RO	4020_400Ah
C144h	Global hardware parameters register 1 (GHWPARAMS1)	32	RO	81E0_C93Bh
C148h	Global hardware parameters register 2 (GHWPARAMS2)	32	RO	0130_290Ah
C14Ch	Global hardware parameters register 3 (GHWPARAMS3)	32	RO	0410_8485h
C150h	Global hardware parameters register 4 (GHWPARAMS4)	32	RO	4782_0004h
C154h	Global hardware parameters register 5 (GHWPARAMS5)	32	RO	0420_4108h
C158h	Global hardware parameters register 6 (GHWPARAMS6)	32	RO	0904_9820h
C15Ch	Global hardware parameters register 7 (GHWPARAMS7)	32	RO	0308_044Dh
C180h	Global high-speed port to bus instance mapping register - low (GPRTBIMAP_HSLO)	32	RW	0000_0000h
C184h	Global high-speed port to bus instance mapping register - high (GPRTBIMAP_HSHI)	32	RW	0000_0000h
C200h	Global USB2 PHY configuration register (GUSB2PHYCFG)	32	RW	4010_2400h
C2C0h	Global USB 3.0 PIPE control register (GUSB3PIPECTL)	32	RW	010C_0002h
C300h	Global transmit FIFO size register (GTXFIFOSIZ_0)	32	RW	0000_0042h
C310h	Global transmit FIFO size register (GTXFIFOSIZ_1)	32	RW	0000_0042h
C320h	Global transmit FIFO size register (GTXFIFOSIZ_2)	32	RW	0000_0042h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C330h	Global transmit FIFO size register (GTXFIFOSIZ_3)	32	RW	0000_0042h
C380h	Global receive FIFO size register (GRXFIFOSIZ_0)	32	RW	0000_0305h
C390h	Global receive FIFO size register (GRXFIFOSIZ_1)	32	RW	0000_0305h
C3A0h	Global receive FIFO size register (GRXFIFOSIZ_2)	32	RW	0000_0305h
C400h	Global event buffer address (low) register (GEVNTADRLO)	32	RW	0000_0000h
C404h	Global event buffer address (high) register (GEVNTADRHI)	32	RW	0000_0000h
C408h	Global event buffer size register (GEVNTSIZ)	32	RW	0000_0000h
C40Ch	Global event buffer count register (GEVNTCOUNT)	32	RW	0000_0000h
C600h	Global hardware parameters register 8 (GHWPARAMS8)	32	RO	0000_0904h
C610h	Global device TXFIFO DMA priority register (GTXFIFOPRIDEV)	32	RW	0000_0000h
C618h	Global host TXFIFO DMA priority register (GTXFIFOPRIHST)	32	RW	0000_0000h
C61Ch	Global host RXFIFO DMA priority register (GRXFIFOPRIHST)	32	RW	0000_0000h
C624h	Global host FIFO DMA high-low priority ratio register (GDMAHLRATIO)	32	RW	0000_0808h
C630h	Global frame length adjustment register (GFLADJ)	32	RW	0000_0000h
C700h	Device configuration register (DCFG)	32	RW	0008_0804h
C704h	Device control register (DCTL)	32	RW	00F0_0000h
C708h	Device event enable register (DEVTEN)	32	RW	0000_0000h
C70Ch	Device status register (DSTS)	32	RO	0052_0004h
C710h	Device generic command parameter register (DGCMDPAR)	32	RW	0000_0000h
C714h	Device generic command register (DGCMD)	32	RW	0000_0000h
C720h	Device active USB endpoint enable register (DALEPENA)	32	RW	0000_0000h
C800h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_0)	32	RW	0000_0000h
C804h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_0)	32	RW	0000_0000h
C808h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_0)	32	RW	0000_0000h
C80Ch	Device physical endpoint-n command register (DEPCMD_0)	32	RW	0000_0000h
C810h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_1)	32	RW	0000_0000h
C814h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_1)	32	RW	0000_0000h
C818h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_1)	32	RW	0000_0000h
C81Ch	Device physical endpoint-n command register (DEPCMD_1)	32	RW	0000_0000h
C820h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_2)	32	RW	0000_0000h
C824h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_2)	32	RW	0000_0000h
C828h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C82Ch	Device physical endpoint-n command register (DEPCMD_2)	32	RW	0000_0000h
C830h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_3)	32	RW	0000_0000h
C834h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_3)	32	RW	0000_0000h
C838h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_3)	32	RW	0000_0000h
C83Ch	Device physical endpoint-n command register (DEPCMD_3)	32	RW	0000_0000h
C840h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_4)	32	RW	0000_0000h
C844h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_4)	32	RW	0000_0000h
C848h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_4)	32	RW	0000_0000h
C84Ch	Device physical endpoint-n command register (DEPCMD_4)	32	RW	0000_0000h
C850h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_5)	32	RW	0000_0000h
C854h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_5)	32	RW	0000_0000h
C858h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_5)	32	RW	0000_0000h
C85Ch	Device physical endpoint-n command register (DEPCMD_5)	32	RW	0000_0000h
C860h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_6)	32	RW	0000_0000h
C864h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_6)	32	RW	0000_0000h
C868h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_6)	32	RW	0000_0000h
C86Ch	Device physical endpoint-n command register (DEPCMD_6)	32	RW	0000_0000h
C870h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_7)	32	RW	0000_0000h
C874h	Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_7)	32	RW	0000_0000h
C878h	Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_7)	32	RW	0000_0000h
C87Ch	Device physical endpoint-n command register (DEPCMD_7)	32	RW	0000_0000h
CC20h	ADP configuration register (ADPCFG)	32	RW	0000_0001h
CC24h	ADP control register (ADPCTL)	32	RW	0000_0001h
CC28h	ADP event register (ADPEVT)	32	RW	0000_0000h
CC2Ch	ADP event enable register (ADPEVTEN)	32	RW	0200_0001h

28.2.2 Capability registers length and HC interface version number (CAPLENGTH)

28.2.2.1 Offset

Register	Offset
CAPLENGTH	0h

28.2.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HCIVERSION															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								CAPLENGTH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

28.2.2.3 Fields

Field	Function
31-16 HCIVERSION	HC interface version number The value is set as 32'h100.
15-8 —	Reserved
7-0 CAPLENGTH	Capability registers length The value is 32'h20.

28.2.3 Host controller structural parameters 1 (HCSPARAMS1)

28.2.3.1 Offset

Register	Offset
HCSPARAMS1	4h

28.2.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAXPORTS								Reserved					MAXINTRS		
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAXINTRS								MAXSLOTS							
W																
Reset	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1

28.2.3.3 Fields

Field	Function
31-24 MAXPORTS	Maximum number of ports set as 2.
23-19 —	Reserved
18-8 MAXINTRS	Number of interrupters set as 1.
7-0 MAXSLOTS	Number of device slots set as 127.

28.2.4 Host controller structural parameters 2 (HCSPARAMS2)

28.2.4.1 Offset

Register	Offset
HCSPARAMS2	8h

28.2.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAXSCRATCHPADBUFS					SP R	MAXSCRATCHPADBUFS_HI					Reserved				
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								ERSTMAX				IST			
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1

28.2.4.3 Fields

Field	Function
31-27 MAXSCRATCH PADBUFS	Max scratchpad buffers low set as 2
26 SPR	Scratchpad restore
25-21 MAXSCRATCH PADBUFS_HI	Max scratchpad buffers high
20-8 —	Reserved

Table continues on the next page...

Field	Function
7-4 ERSTMAX	Event ring segment table max set as 15
3-0 IST	Isochronous scheduling threshold set as 1.

28.2.5 Host controller structural parameters 3 (HCSPARAMS3)

28.2.5.1 Offset

Register	Offset
HCSPARAMS3	Ch

28.2.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	U2_DEVICE_EXIT_LAT															
W																
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								U1_DEVICE_EXIT_LAT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

28.2.5.3 Fields

Field	Function
31-16 U2_DEVICE_EXIT_LAT	U2 device exit latency set as 32'h7FF
15-8 —	Reserved
7-0	U1 device exit latency set as 32'hA

Field	Function
U1_DEVICE_EXIT_LAT	

28.2.6 Host controller capability parameters 1 (HCCPARAMS1)

28.2.6.1 Offset

Register	Offset
HCCPARAMS1	10h

28.2.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	xECP															
W																
Reset	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAXPSASIZE				CFC	SEC	SPC	PAE	NSS	LTC	LHRC	PIND	PPC	CSZ	BNC	AC64
W																
Reset	1	1	1	1	0	1	1	0	0	1	1	0	1	1	0	1

28.2.6.3 Fields

Field	Function
31-16 xECP	xHCI extended capabilities pointer set as 544
15-12 MAXPSASIZE	Maximum primary stream array size set as 15
11	Contiguous frame ID capability

Table continues on the next page...

Field	Function
CFC	
10 SEC	Stopped EDLTA capability
9 SPC	Short packet capability
8 PAE	Parse all event data
7 NSS	No secondary SID support
6 LTC	Latency tolerance messaging capability
5 LHRC	Light HC reset capability
4 PIND	Port indicators
3 PPC	Port power control
2 CSZ	Context size
1 BNC	BW negotiation capability
0 AC64	64-bit addressing capability

28.2.7 Doorbell offset (DBOFF)

28.2.7.1 Offset

Register	Offset
DBOFF	14h

28.2.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DOORBELL_ARRAY_OFFSET															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOORBELL_ARRAY_OFFSET														Reserved	
W																
Reset	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0

28.2.7.3 Fields

Field	Function
31-2 DOORBELL_ARRAY_OFFSET	Doorbell array offset set as 1152
1-0 —	Reserved

28.2.8 Runtime register space offset (RTSOFF)

28.2.8.1 Offset

Register	Offset
RTSOFF	18h

28.2.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RUNTIME_REG_SPACE_OFFSET															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RUNTIME_REG_SPACE_OFFSET											Reserved				
W																
Reset	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0

28.2.8.3 Fields

Field	Function
31-5 RUNTIME_REG_SPACE_OFFSET	Runtime register space offset set as 1088
4-0 —	Reserved

28.2.9 Host controller capability parameters 2 (HCCPARAMS2)

28.2.9.1 Offset

Register	Offset
HCCPARAMS2	1Ch

28.2.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										CIC	LEC	CTC	FSC	CMC	U3C
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

28.2.9.3 Fields

Field	Function
31-6 —	Reserved
5 CIC	Configuration information capability
4 LEC	Large ESIT payload capability
3 CTC	Compliance transition capability
2 FSC	Force save context capability
1 CMC	Configure endpoint command max exit latency too large capability
0 U3C	U3 entry capability

28.2.10 Global SoC bus configuration register 0 (GSBUSCFG0)

28.2.10.1 Offset

Register	Offset
GSBUSCFG0	C100h

28.2.10.2 Function

This register cconfigures system bus DMA options for the AXI master bus. Options include burst length and cache type (bufferable/posted, cacheable/snoop, and so on). The application can program this register upon power-on, or a change in mode of operation after the DMA engine is halted.

xHCI register power-on value

While using a standard xHCI host driver, make sure to set the register's power-on value (current values are 32'h100080 and 32'h700) because the standard xHCI driver does not access this register.

Burst length

For a given DMA transfer, the burst length is set according to the largest enabled burst length. Note that the undefined length INCR burst, if enabled, has priority over all other burst lengths.

Cache type

For a given DMA transfer, the cache type is set according to the 4-bit cache type register bit corresponding to the transfer type:

- Data read
- Descriptor read
- Data write
- Descriptor write

The definition of the 4-bit cache type register bit corresponds to the cache type definition of the configured master bus type (AXI3) as defined below:

Table 28-4. Cache type bit assignments

MBUS_TYPE	Cache type output	[3]	[2]	[1]	[0]
AXI3	ar/awcache[3:0]	Write allocate	Read allocate	Cacheable	Bufferable

Cache type in AXI

AXI3: CSR cache type refers to AXI cache type (awcache[3:0] and arcache[3:0]); or in ACE (AXI coherency extensions) to AXI memory type.

Posted requests

The native interface DMA read request gmr_mcmd[2:0] indicates posted vs non-posted requests depending on the bufferable/posted bit of the cache type.

Table 28-5. Cache type bit definitions and usage

Signal name	Definition	Usage
Bufferable, Posted	The transaction response can be returned quickly, but the transaction itself can take an arbitrary number of clock cycles to reach the final destination.	• Improves bus utilization of DMA writes with a high response latency, such as off-chip memory or bridges. • Allows the core to start a new DMA write transaction while a DMA write is in progress. • For descriptor write transactions, "Bufferable/ Posted" must be 0 to avoid a DMA write race condition with a software interrupt or subsequent DMA read. When the core writes back a descriptor or event, it waits for the DMA response to indicate that it can safely re-fetch that descriptor or set the core's interrupt. If the descriptor DMA write is posted, the data integrity is not ensured for future transactions.
Cacheable, Modifiable, Snoop (negation of No Snoop)	The characteristics of the transaction at the destination may not match the original.	• For DMA writes, multiple write transactions may be merged. • For DMA reads, a location can be pre-fetched or fetched just once for multiple read transactions. • For AXI, this bit should be used in conjunction with the read allocate / write allocate to indicate upon a cache miss whether the transaction should be cached, or other allocate bits to indicate how to handle cache misses.

Connections from GSBUSCFG0 to master cache type

In device mode, the GSBUSCFG0 cache type bits are connected to the two (write and read) system bus master cache type ports through a pair of 4-bit wide 2x1 multiplexers; the MUX select signal is the descriptor access indication (as opposed to data).

In host mode, the GSBUSCFG0 cache type bits are connected as in device mode with these exceptions:

- For data accesses, the core sets cacheable to the inverse of the xHCI TRB's (transfer request block) no snoop flag; the cacheable bit in GSBUSCFG0 is disregarded.
- For descriptor accesses, the core sets cacheable to the inverse of the internal DMA command's no snoop flag. When the LSP performs a Scratchpad DMA write, it sets the internal DMA command's No Snoop flag thereby ensuring cacheable=0, as required by the xHCI specification.

The master bus type is AXI, the GSBUSCFG0 to cache type connections are reflected by following diagrams.

Figure 28-2. GSBUSCFG0 cache type connections for AXI

NOTE

For master interface DMA access, program the GSBUSCFG0 register to 0x2222000F for better performance.

28.2.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATRDREQINFO				DESRDREQINFO				DATWRREQINFO				DESWRREQINFO			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								INCR256BRSTENA	INCR128BRSTENA	INCR64BRSTENA	INCR32BRSTENA	INCR16BRSTENA	INCR8BRSTENA	INCR4BRSTENA	INCRBRSTENA
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

28.2.10.4 Fields

Field	Function
31-28 DATRDREQINFO	AXI-cache for data read Input to BUS-GM.
27-24 DESRDREQINFO	AXI-cache for descriptor read Input to BUS-GM.
23-20 DATWRREQINFO	AXI-cache for data write Input to BUS-GM.
19-16 DESWRREQINFO	AXI-cache for descriptor write Input to BUS-GM.
15-8 —	Reserved
7 INCR256BRSTENA	NCR256 burst type enable Input to BUS-GM. For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 256-beat burst. **NOTE:** The bit is not supported for now.
6 INCR128BRSTENA	NCR128 burst type enable Input to BUS-GM. For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 128-beat burst. **NOTE:** The bit is not supported for now.

Table continues on the next page...

Field	Function
5 INCR64BRSTENA	INCR64 burst type enable Input to BUS-GM. For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 64-beat burst. **NOTE:** The bit is not supported for now.
4 INCR32BRSTENA	INCR32 burst type enable Input to BUS-GM. For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 32-beat burst. **NOTE:** The bit is not supported for now.
3 INCR16BRSTENA	INCR16 burst type enable Input to BUS-GM. For the AXI configuration, if software sets this bit to 1, the AXI master uses INCR to do the 16-beat burst.
2 INCR8BRSTENA	INCR8 burst type enable Input to BUS-GM. For the AXI configuration, if software sets this bit to 1, the AXI master uses INCR to do the 8-beat burst.
1 INCR4BRSTENA	INCR4 burst type enable Input to BUS-GM. For the AXI configuration, when this bit is enabled, the controller is allowed to do bursts of beat length 1, 2, 3, and 4. It is highly recommended that this bit is enabled to prevent descriptor reads and writes from being broken up into separate transfers.
0 INCRBRSTENA	Undefined length INCR burst type enable Input to BUS-GM This bit determines the set of burst lengths the master interface uses. It works in conjunction with the GSBUSCFG0[7-1] enables (INCR256/128/64/32/16/8/4). 0b - INCRX burst mode ARLEN/AWLEN (for AXI configurations) use only the following burst lengths: • 1 (if GSBUSCFG0[1:7] = 0 • 4 (if GSBUSCFG0[INCR4BRSTENA] = 1) • 8 (if GSBUSCFG0[INCR8BRSTENA] = 1) • 16 (if GSBUSCFG0[INCR16BRSTENA] = 1) • 32 (if GSBUSCFG0[INCR32BRSTENA] = 1) • 64 (if GSBUSCFG0[INCR64BRSTENA] = 1) • 128 (if GSBUSCFG0[INCR128BRSTENA] = 1) • 256 (if GSBUSCFG0[INCR256BRSTENA] = 1) They do not use INCR. 1b - INCR (undefined length) burst mode AXI configurations- ARLEN/AWLEN uses any length less than or equal to the largest-enabled burst length of INCR4/8/16/32/64/128/256. For cache line-aligned applications, this bit is typically set to 0 to ensure that the master interface uses only power-of-2 burst lengths (as enabled through GSBUSCFG0[7-0]).

28.2.11 Global SoC bus configuration register 1 (GSBUSCFG1)

28.2.11.1 Offset

Register	Offset
GSBUSCFG1	C104h

28.2.11.2 Function

NOTE

For master interface DMA access, program the GSBUSCFG1 register to 0x00000F00 for better performance.

28.2.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			EN1KPAGE	PIPETRANSLIMIT				Reserved							
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

28.2.11.4 Fields

Field	Function
31-13 —	Reserved
12 EN1KPAGE	1K page boundary enable By default (this bit is disabled), the AXI breaks transfers at the 4K page boundary. When this bit is enabled, the AXI master (DMA data) breaks transfers at the 1K page boundary.
11-8	AXI pipelined transfers burst request limit

Table continues on the next page...

Field	Function
PIPETRANSLIMIT	The bit controls the number of outstanding pipelined transfers requests the AXI master pushes to the AXI slave. Once the AXI master reaches this limit, it does not make more requests on the AXI ARADDR and AWADDR buses until the associated data phases complete. 0000b - 1 request 0001b - 2 requests 0010b - 3 requests 0011b - 4 requests... 1111b - 16 requests
7-0 —	Reserved

28.2.12 Global Tx threshold control register (GTXTHRCFG)

28.2.12.1 Offset

Register	Offset
GTXTHRCFG	C108h

28.2.12.2 Function

All bits in the GTXTHRCFG register are valid only in host mode. This register is not applicable for debug target and USB 2.0-only mode.

28.2.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		USBTXPKTCNTSEL	Reserved	USBTXPKTCNT				USBMAXTXBURSTSIZE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.12.4 Fields

Field	Function
31-30 —	Reserved
29 USBTXPKTCNTSEL	USB transmit packet count enable This bit enables/disables the USB transmission multi-packet thresholding. 0b - USB transmission multi-packet thresholding is disabled, the core can only start transmission on the USB after the entire packet has been fetched into the corresponding TXFIFO. 1b - USB transmission multi-packet thresholding is enabled. The core can only start transmission on the USB after USB transmit packet count amount of packets for the USB transaction (burst) are already in the corresponding TXFIFO. This mode is only valid in the host mode. It is only used for SuperSpeed.
28 —	Reserved
27-24 USBTXPKTCNT	USB transmit packet count This bit specifies the number of packets that must be in the TXFIFO before the core can start transmission for the corresponding USB transaction (burst). This bit is only valid when the USBTXPKTCNTSEL bit is set to 1. The valid values are from 1 to 15. **NOTE:** This bit must be less than or equal to the USBMAXTXBURSTSIZE bit.
23-16 USBMAXTXBURSTSIZE	USB maximum Tx burst size When USBTXPKTCNTSEL is set to 1, this bit specifies the maximum bulk OUT burst, the core can do. When the system bus is slower than the USB, TXFIFO can underrun during a long burst. User can program a smaller value to this bit to limit the TX burst size that the core can do. It only applies to SS Bulk, isochronous, and interrupt OUT endpoints in the host mode. Valid values are from 1 to 16.

Table continues on the next page...

Field	Function
15-0 —	Reserved

28.2.13 Global Rx threshold control register (GRXTHRCFG)

28.2.13.1 Offset

Register	Offset
GRXTHRCFG	C10Ch

28.2.13.2 Function

This register is not applicable for debug target and USB 2.0-only mode.

28.2.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		USBRXPKTCNTSEL	Reserved	USBRXPKTCNT				USBMAXRXBURSTSIZE					Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.13.4 Fields

Field	Function
31-30 —	Reserved
29 USBRXPKTCNTSEL	USB receive packet count enable This bit enables/disables the USB reception multi-packet thresholding. In device mode, Setting this bit to 1 also enables the functionality of reporting NUMP in the ACK TP based on the RXFIFO space instead of reporting a fixed NUMP derived from DCFG[NUMP]. While using external buffer control (EBC) feature, disable this mode by setting USBRXPKTCNTSEL to 0. 0b - The core can only start reception on the USB when the RXFIFO has space for at least one packet. 1b - The core can only start reception on the USB when the RXFIFO has space for at least USBRXPKTCNT amount of packets. This mode is valid in both host and device mode. It is only used for SuperSpeed.
28 —	Reserved
27-24 USBRXPKTCNT	USB receive packet count In host mode, this bit specifies the space (in terms of the number of packets) that must be available in the RXFIFO before the core can start the corresponding USB RX transaction (burst). In device mode, this bit specifies the space (in terms of the number of packets) that must be available in the RXFIFO before the core can send ERDY for a flow-controlled endpoint. This bit is valid only when the USBRXPKTCNTSE bit is set to 1. The valid values for this bit are from 1 to 15. **NOTE:** This bit must be less than or equal to the USB maximum receive burst size (USBMAXRXBURSTSIZE) bit.
23-19 USBMAXRXBURSTSIZE	USB maximum receive burst size In host mode, this bit specifies the maximum bulk IN burst, the USB 3.0 core can perform. When the system bus is slower than the USB, RXFIFO can overrun during a long burst. User can program a smaller value to this bit to limit the RX burst size that the core can perform. It only applies to SS bulk, isochronous, and interrupt IN endpoints in the host mode. In device mode, this bit specifies the NUMP value that is sent in ERDY for an OUT endpoint. This bit is valid only when USBRXPKTCNTSEL is set to 1. The valid values for this bit are from 1 to 16.
18-0 —	Reserved

28.2.14 Global core control register (GCTL)

28.2.14.1 Offset

Register	Offset
GCTL	C110h

28.2.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	PWRDNSCALE													MASTERFILTBYPASS	BYPSSETADDR	U2RSTECN
Reset	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	FRMSCLDWN		PRTCAPDIR		CORESOFTRESET	SOFITPSYNC	Reserved	Reserved	RAMCLKSEL		Reserved		DISSCRAMBLE	U2EXIT_LFPS	Reserved	DSBLCLKGTNG
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0

28.2.14.3 Fields

Field	Function
31-19 PWRDNSCALE	Power down scale The USB3 suspend_clk input replaces pipe3_rx_pclk as a clock source to a small part of the USB3 core that operates when the SS PHY is in its lowest power (P3) state, and therefore does not provide a clock. The power down scale bit specifies how many suspend_clk periods fit into a 16 KHz clock period. When performing the division, round up the remainder. For example, when using an 8-/16-/32-bit PHY and 25 MHz suspend clock, Power down scale = 25000 KHz/16 KHz = 13'd1563 (rounder up) **NOTE:** • Minimum suspend clock frequency is 32 KHz • Maximum suspend clock frequency is 125 MHz The LTSSM uses suspend clock for 12 ms and 100 ms timers during suspend mode. According to the USB 3.0 specification, the accuracy on these timers is 0% to +50%.

Table continues on the next page...

Field	Function
	• 12 ms + 0~+50% accuracy = 18 ms (Range is 12 ms - 18 ms) • 100 ms + 0~+50% accuracy = 150 ms (Range is 100 ms - 150 ms) The suspend clock accuracy requirement is: • (12,000/62.5) * (GCTL[PWRDNSCALE]) * actual suspend_clk_period should be between 12,000 and 18,000 • (100,0000/62.5) * (GCTL[PWRDNSCALE]) * actual suspend_clk_period should be between 100,000 and 150,000 For example, if the suspend_clk frequency varies from 7.5 MHz to 10.5 MHz, then the value needs to programmed is: Power down scale = 10500/16 = 657 (rounded up; and fastest frequency used)
18 MASTERFILTBYPASS	Master filter bypass 0b - Enable all the filter modules 1b - Bypass all the filter modules
17 BYPSSETADDR	Bypass SetAddress in device mode When this bit is set, the device core uses the value in DCFG[DEVADDR] bits directly for comparing the device address in the tokens. **NOTE:** This bit must be set to 0.
16 U2RSTECN	If the SuperSpeed connection fails during POLL or LMP exchange, the device connects at non-SS mode. If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode. **NOTE:** This bit is applicable only in device mode.
15-14 FRMSCLDWN	This bit scales down device view of a SoF/USOF/ITP duration. For SS/HS mode: 11 Implements interval to be 15.625 µs 10 Implements interval to be 31.25 µs 01 Implements interval to be 62.5 µs 00 Implements interval to be 125 µs For FS mode, the scale-down value is multiplied by 8. 00 1024 bytes 01 512 bytes 10 256 bytes 11 128 bytes
13-12 PRTCAPDIR	Port capability direction The sequence for switching modes in DRD configuration is as follows: Switching from device to host: 1. Reset the controller using GCTL[CORESOFTRESET]. 2. Set GCTL[PRTCAPDIR] to 2'b01 (Host mode). 3. Reset the host using USBCMD[HCRESET]. 4. Follow the steps in "Initializing Host Registers". Switching from host to device: 1. Reset the controller using GCTL[CORESOFTRESET]. 2. Set GCTL[PRTCAPDIR] to 2'b10 (Device mode). 3. Reset the device by setting DCTL[CSFTRST]. 4. Follow the steps in "Register Initialization".

Table continues on the next page...

Field	Function
	01b - For host configurations 10b - For device configurations 11b - Reserved
11 CORESOFTRESET	Core Soft Reset 0 No soft reset 1 Soft reset to core Clears the interrupts and all the CSRs except the following registers: GCTL GUCTL GSTS GGPIO GUID GUSB2PHYCFGn registers GUSB3PIPECTLn registers DCFG DCTL DEVTEN DSTS While resetting PHYs (using GUBS3PHYCFG or GUSB3PIPECTL registers), the core must be in reset state until PHY clocks are stable. This controls the bus, ram, and mac domain resets. **NOTE:** This bit is for debug purpose only. Use USBCMD.HCRESET in xHCI Mode and DCTL.SoftReset in device mode for soft reset.
10 SOFITPSYNC	The bit is set to 0. The core keeps the UTMI PHY on the first port in a non-suspended state whenever there is a SuperSpeed port that is not in Rx.Detect, SS.Disable and U3.
9 —	Reserved
8 —	Reserved
7-6 RAMCLKSEL	RAM clock select On USB-reset, hardware clears these bits to 2'b00. **NOTE:** In host mode, this bit must be set to 2'b00, that is, the ram_clk must be assigned to bus_clk only. The reason is if the SS port0 goes to P3, the pipe_clk is shutdown, and the USB 2.0 ports cannot operate. 00b - Bus clock 01b - Pipe clock 10b - Pipe/2 clock 11b - Reserved
5-4 —	Reserved
3 DISSCRAMBLE	Disable scrambling Transmit request to link partner on next transition to recovery or polling.

Table continues on the next page...

Field	Function
2 U2EXIT_LFPS	This bit is added to improve interoperability with a third party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4 µs duration. This causes the device to exit from U2 state because the LFPS filter value is 248 ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller. 0b - The link treats 248 ns LFPS as a valid U2 exit. 1b - The link waits for 8 µs of LFPS before it detects a valid U2 exit.
1 —	Reserved
0 DSBLCLKGTNG	Disable clock gating When this bit is set to 1 and the core is in low power mode, internal clock gating is disabled. Set this bit to 1 after power-on-reset.

28.2.15 Global status register (GSTS)

28.2.15.1 Offset

Register	Offset
GSTS	C118h

28.2.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												Reserved			
Reset	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				Reserved	Reserved	Reserved	Reserved	HOST_IP	DEVICE_IP	CSRTIMEOUT	BUSERRADDRVLD	Reserved		CURMOD	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.15.3 Fields

Field	Function
31-20 —	Reserved
19-12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 HOST_IP	Host interrupt pending This bit indicates that there is a pending interrupt pertaining to xHC in the host event queue.
6 DEVICE_IP	Device interrupt pending This bit indicates that there is a pending interrupt pertaining to peripheral (device) operation in the device event queue.
5 CSRTIMEOUT	CSR timeout When this bit is set to 1, it indicates that software performed a write or read to a core register that could not be completed within 17'h1FFFF bus clock cycles.
4 BUSERRADDR VLD	Bus error address valid Indicates that the GBUSERRADDR register is valid and reports the first bus address that encounters a bus error.
3-2 —	Reserved
1-0 CURMOD	Current mode of operation 00b - Device mode 01b - Host mode

28.2.16 Global user control register 1 (GUCTL1)

28.2.16.1 Offset

Register	Offset
GUCTL1	C11Ch

28.2.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved		FILTER_SE0_FSLS_EOP	TX_IPGAP_LINECHECK_DIS	DEV_TRB_OUT_SPR_IND	DEV_FORCE_20_CLK_FOR_30_CLK	P3_IN_U2	DEV_L1_EXIT_BY_HW	IP_GAP_ADD_ON			DEV_LSP_TAIL_LOCK_DIS	NAK_PER_ENH_FS	NAK_PER_ENH_HS	Reserved	PARKMODE_DISABLE_HS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	Reserved						L1_SUSP_THRLD_EN_FOR_HOST	L1_SUSP_THRLD_FOR_HOST				HC_ERRATA_ENABLE	HC_PARCHK_DISABLE	OVRLD_L1_SUSP_COM	LOA_FILTER_EN
Reset	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0

28.2.16.3 Fields

Field	Function
31-30 —	Reserved
29 FILTER_SE0_FSLS_EOP	This bit is applicable for FS/LS operation. If this feature is enabled, then SE0 on the linestate is validated for 2 consecutive utmi clock edges for EOP detection. This feature is applicable only in FS in device mode and FS/LS mode of operation in host mode.

Table continues on the next page...

Field	Function
	• Device mode: FS - If GUCTL1[FILTER_SE0_FSLS_EOP] is set, then for device LPM hanshake, the core ignores single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS. • Host mode: FS/LS - If GUCTL1[FILTER_SE0_FSLS_EOP] is set, then the core ignores single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS/LS port. Enable this feature if the LineState has SE0 glitches during transmission. This bit is quasi-static, i.e., should not be changed during device operation. 0b - No change in linestate check for SE0 detection in FS/LS (default) 1b - Feature enabled, FS/LS SE0 is filtered for 2 clocks for detecting EOP
28 TX_IPGAP_LINECHECK_DIS	This bit is applicable for HS operation of u2mac. If this feature is enabled, then the 2.0 mac operating in HS ignores the UTMI Linestate during the transmit of a token (during token-to-token and token-to-data IPGAP). When enabled, the controller implements a fixed 40-bit TxEndDelay after the packet is given on UTMI and ignores the linestate during this time. This fetaure is applicable only in HS mode of operation. Device mode: If GUCTL1[TX_IPGAP_LINECHECK_DIS] is set, then for device LPM handshake, the core ignores the linestate after TX and wait for a fixed clocks (40 bit times equivalent) after transmiting ACK on utmi. Host mode: If GUCTL1[TX_IPGAP_LINECHECK_DIS] is set, then the ipgap between (tkn to tkn/data) is added by 40 bit times of TXENDDELAY, and linestate is ignored during this 40 bit times delay. Enable this bit if the linestate does not reflect the expected line state (J) during transmission. This bit is quasi-static, i.e., should not be changed during device operation. 0b - No change in linestate check (default) 1b - Feature enabled, 2.0 MAC disables linestate check during HS transmit
27 DEV_TRB_OUT_SPR_IND	This bit is applicable for device mode only (and ignored in host mode). If the device application (SW/HW) wants to know if a short packet was received for an OUT in the TRB status itself, then this feature can be enabled, so that a bit is set in the TRB writeback in the buf_size DWORD. Bit[26] - SPR of the {trbstatus, RSVD, SPR, PCM1, bufsize} DWORD is set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. This bit is quasi-static, i.e., should not be changed during device operation. 0b - No change in TRB status DWORD (default) 1b - Feature enabled, OUT TRB status indicates short packet
26 DEV_FORCE_20_CLK_FOR_30_CLK	This bit is applicable (and to be set) for device mode (DCFG[SPEED] != SS) only. In the 3.0 device core, if the core is programmed to operate in 2.0 only (i.e., device speed is programmed to 2.0 speeds in DCFG[SPEED]), then setting this bit makes the internal 2.0 (utmi) clock to be routed as the 3.0 (pipe) clock. Enabling this feature allows the pipe3 clock to be not-running when forcibily operating in 2.0 device mode. **NOTE:** When using this feature, all pipe3 inputs must be in inactive mode, esp. pipe3 clocks not running and pipe3_phystatus_async must be tied to 0. This bit should not be set if the core is programmed to operate in SuperSpeed mode (even when it falls back to 2.0). This bit is quasi-static, i.e., should not be changed during operation. 0b - Uses 3.0 clock when operating in 2.0 mode (default) 1b - Feature enabled
25 P3_IN_U2	Setting this bit enables P3 power state when the SuperSpeed link is in U2. Another power saving option. Check with the PHY vendor before enabling this option. When setting this bit to 1 to enable P3 in P2, GUSB3PIPECTL[27] should be set to 0 to make sure that the U2 exit is attempted in P0. This bit should be set only when GCTL[SOFITPSYNC]=1 or GFLADJ[GFLADJ_REFCLK_LPM_SEL]=1. 0b - When SuperSpeed link is in U2 , PowerState P2 is attempted on the PIPE interface (default) 1b - When SuperSpeed link is in U2, PowerState P3 is attempted if GUSB3PIPECTL[17] is set
24	This bit is applicable for device mode (2.0) only. This bit enables device controller sending remote wakeup for L1 if the device becomes ready for sending/accepting data when in L1 state. If the host expects the device to send remote wakeup signalling to resume after going into L1 in flow controlled

Table continues on the next page...

Field	Function
DEV_L1_EXIT_BY_HW	state, then this bit can be set to send the remote wake signal automatically when the device controller becomes ready. This hardware remote wake feature is applicable only to bulk and interrupt transfers, and not for Isoch/Control. When control transfers are in progress, the LPM is rejected (NYET response). Only after control transfers are completed (either with ACK/STALL), LPM is accepted. For Isoch transfers, the host needs to do the wake-up and start the transfer. The device controller does not do remote-wakeup when Isoch endpoints get ready. The device software needs to keep the GUSB2PHYCFG[ENBLSLPM] reset in order to keep the PHY clock to be running for keeping track of SOF intervals. This bit is quasi-static, i.e., should not be changed during device operation. 0b - Disables device L1 hardware exit logic (default) 1b - Feature enabled
23-21 IP_GAP_ADD_ON	This bit is used to add on to the default inter packet gap setting in the USB 2.0 MAC.
20 DEV_LSP_TAIL_LOCK_DIS	This is a bug fix for STAR 9000716195 that affects the CSP mode for OUT endpoints in device mode. The issue is that tail TRB index is not synchronized with the cache scratchpad bytecount update. If the fast-forward request comes in-between the bytecount update ona newly fetched TRB and the tail-index write update in TPF, the RDP works on an incorrect tail index and misses the byte count decrement for the newly fetched TRB in the fast-forwarding process. This fix needs to be present all the times. 0b - Enables device lsp lock logic for tail TRB update (default) 1b - Fix disabled
19 NAK_PER_ENH_FS	If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other full-speed bulk endpoint which is ACked by the device. Setting this bit to 1 enables the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence improves the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for the full-speed application. Setting this bit only controls, and is only required for full-speed transfers. 0b - Enhancement not applied 1b - Enables performance enhancement for FS async endpoints in the presence of NAKs
18 NAK_PER_ENH_HS	If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other high-speed bulk endpoint which is ACKed by the device. Setting this bit to 1, enables the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence improves the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for the high-speed application. Setting this bit only controls, and is only required for high-speed transfers. 0b - Enhancement not applied 1b - Enables performance enhancement for HS async endpoints in the presence of NAKs
17 —	Reserved
16 PARKMODE_DISABLE_HS	This bit is used only in host mode. When this bit is set to 1, all HS bus instances park mode are disabled. To improve performance in park mode, the xHCI scheduler queues in three requests of 4 packets each for high-speed asynchronous endpoints in a micro-frame. But if a device is slow and if it NAKs more than 3 times, then it is rescheduled only in the next micro-frame. This could decrease the performance of a slow device even further. In a few high-speed devices (such as Sandisk Cruzer Blade 4GB VID:1921, PID:21863 and Flex Drive VID:3744, PID:8552) when an IN request is sent within 900 ns of the ACK of the previous packet, these devices send a NAK. When connected to these devices, if required, the software can disable the park

Table continues on the next page...

Field	Function
	mode if performance drop is found in your system. When park mode is disabled, pipelining of multiple packet is disabled and instead one packet at a time is requested by the scheduler. This allows up to 12 NAKs in a micro-frame and improves performance of these slow devices.
15 —	Reserved
14-9 —	Reserved
8 L1_SUSP_THRLD_EN_FOR_HOST	This bit is used only in host mode. The host controller asserts the utmi_l1_suspend_n and utmi_sleep_n output signals to the PHY in the L1 state. 0b - Disable 1b - Enable
7-4 L1_SUSP_THRLD_FOR_HOST	This bit is effective only when the L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1. 1111 State is normal working 0110 State is L2 Suspend 0101 State is L1 Suspend 1011 State is L1 Sleep
3 HC_ERRATA_ENABLE	Host ELD enable When this bit is set to 1, it enables the exit latency delta (ELD) support defined in the xHCI 1.0 errata. This bit is used only in the host mode.
2 HC_PARCHK_DISABLE	Host parameter check disable 0b - The xHC checks that the input slot/EP context bits comply to the xHCI Specification. Upon detection of a parameter error during command execution, the xHC generates an event TRB with completion code indicating 'PARAMETER ERROR'. (default) 1b - The xHC does not perform parameter checks and does not generate 'PARAMETER ERROR' completion code.
1 OVRLD_L1_SUSP_COM	If this bit is set, the utmi_l1_suspend_com_n is overloaded with the utmi_sleep_n signal. This bit is usually set if the PHY stops the port clock during L1 sleep condition. **NOTE:** The recommended connection for the SUSPENDM/SLEEPM signals to the PHY with respect to this bit is as follows. For non-0 ports, connect • utmi_sleep_n[n] to SLEEPM[n] • (utmi_suspend_n[n] & utmi_l1_suspend_n[n]) to SUSPENDM[n] • USB2 PHYCLK[n] to utmi_clk[n] GUCTL1[OVRLD_L1_SUSP_COM] impacts only port 0. For port 0: • For PHY, GUSB2PHYCFGn[U2_FREECLK_EXISTS]=1: With this connection, the PHY keeps PLL active so that FREECLK is always available irrespective of suspend/sleep. • Connect USB2 PHY COMMONONN to 0. • Connect utmi_sleep_n[0] to SLEEPM[0]. • Connect (utmi_suspend_n[0] & utmi_l1_suspend_n[0]) to SUSPENDM[0]. • Connect USB2 PHY FREECLK to utmi_clk[0].

Table continues on the next page...

Field	Function
	• Leave utmi_suspend_com_n, utmi_l1_suspend_com_n unconnected. • GUCTL1[OVRLD_L1_SUSP_COM] can be set to any value. • For third-party PHY, GUSB2PHYCFGn[U2_FREECLK_EXISTS]=0: With this connection the PHY can shut off all the clocks when the required conditions are met (like, GUSB2PHYCFGn[8,6], GUCTL1[1], GFLADJ[23], GCTL[10], Suspend condition, HW LPM enable etc). • Connect ~utmi_suspend_com_n to SUSPENDM[0] (or equivalent). • Connect ~utmi_l1_suspend_com_n to SLEEPM[0] (or equivalent). • Connect PHYCLK0 (first port clock) to utmi_clk[0]. • Leave utmi_suspend_n[0], utmi_l1_suspend_n[0], utmi_sleep_n[0] unconnected. • Set GUCTL1[OVRLD_L1_SUSP_COM] to 1'b1.
0 LOA_FILTER_EN	If this bit is set, the USB 2.0 port babble is checked at least three consecutive times before the port is disabled. This prevents false triggering of the babble condition when using low quality cables. **NOTE:** This bit is valid only in host mode.

28.2.17 Global user ID register (GUID)

28.2.17.1 Offset

Register	Offset
GUID	C128h

28.2.17.2 Function

This register contains the user ID.

28.2.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USERID															
Reset	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	USERID															
Reset	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0

28.2.17.4 Fields

Field	Function
31-0 USERID	User ID

28.2.18 Global user control register (GUCTL)

28.2.18.1 Offset

Register	Offset
GUCTL	C12Ch

28.2.18.2 Function

This register provides a few options for the software to control the core behavior in the host mode. Most of the options are used to improve host inter-operability with different devices.

28.2.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REFCLKPER										NOEXTRDI	Reserved			SPRSCTRLTRANSEN	RESBWHSEPS
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDEVADDR	USBHSTINAUTORETRYEN	ENOVERLAPCHK	EXTCAPSUPTEN	INSRTEXTRFSBODI	DTCT		DTFT								
Reset	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

28.2.18.4 Fields

Field	Function
31-22 REFCLKPER	This bit indicates in terms of nano seconds the period of ref_clk. The default value of this register is set to 10'h8 (8 ns/125 MHz). This bit needs to be updated during power-on-initialization, if GCTL[SOFITPSYNC] or GFLADJ[GFLADJ_REFCLK_LPM_SEL] is set to 1. The programmable maximum value is 62 ns, and the minimum value is 8 ns. Use the reference clock with a period that is an integer multiple, so that ITP can meet the jitter margin of 32 ns. The allowable ref_clk frequencies whose period is not integer multiples are 16/17/19.2/24/39.7 MHz. This bit should not be set to 0 at any time. If this feature is not planned for future use, then set this bit to the default value (10'h8).
21 NOEXTRDI	No extra delay between SOF and the first packet Some HS devices misbehave when the host sends a packet immediately after a SOF. However, adding an extra delay between a SOF and the first packet can reduce the USB data rate and performance. This bit is used to control whether the host should wait for 2 µs before it sends the first packet after a SOF, or not. User can set this bit to 1 to improve the performance if those problematic devices are not a concern in the user's host environment. 0b - Host waits for 2 µs after a SOF before it sends the first USB packet 1b - Host doesn't wait after a SOF before it sends the first USB packet

Table continues on the next page...

Field	Function
20-18 —	Reserved
17 SPRSCTRLTRANSEN	Sparse control transaction enable Some devices are slow in responding to control transfers. Scheduling multiple transactions in one microframe/frame can cause these devices to misbehave. If this bit is set to 1'b1, the host controller schedules transactions for a Control transfer in different microframes/frames.
16 RESBWHSEPS	Reserving 85% bandwidth for HS periodic EPs By default, HC reserves 80% of the bandwidth for periodic EPs. If this bit is set, the bandwidth is relaxed to 85% to accommodate two high-speed, high bandwidth ISOC EPs. USB 2.0 required 80% bandwidth allocated for ISOC traffic. If two high-bandwidth ISOC devices (HD Webcams) are connected, and if each requires 1024-bytes X 3 packets per micro-frame, then the bandwidth required is around 82%. If this bit is set, then it is possible to connect two Webcams of 1024 bytes X 3 paylod per micro-frame each. Otherwise, the resolution of the Webcams needs to be reduced. This bit is valid in host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
15 CMDEVADDR	Compliance mode for device address When this bit is 1'b1, Slot ID may have different value than device address if max_slot_enabled < 128. The xHCI compliance requires this bit to be set to 1. The 0 mode is for debug purpose only. This allows to easily identify a device connected to a port in the Lecroy or Eliisys trace during hardware debug. This bit is valid in host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode. 0b - Device address is equal to Slot ID 1b - Increment device address on each address device command
14 USBHSTINAUTORETRYEN	Host IN auto retry When set, this bit enables the auto retry feature. For IN transfers (non-isochronous) that encounter data packets with CRC errors or internal overrun scenarios, the auto retry feature causes the host core to reply to the device with a non-terminating retry ACK (that is, an ACK transaction packet with RETRY = 1 and NUMP != 0). If the auto retry feature is disabled, the core responds with a terminating retry ACK (that is, an ACK transaction packet with RETRY = 1 and NUMP = 0). **NOTE:** This bit is also applicable to the device mode. 0b - Auto retry disabled (default) 1b - Auto retry enabled
13 ENOVERLAPCHK	Enable check for LFPS overlap during remote Ux exit 0b - When the link exists U1/U2/U3 because of a remote exit, it does not look for an LFPS overlap. 1b - The SuperSpeed link when exiting U1/U2/U3 waits for either the remote link LFPS or TS1/TS2 training symbols before it confirms that the LFPS handshake is complete. This is done to handle the case where the LFPS glitch causes the link to start exiting from the low power state. Looking for the LFPS overlap makes sure that the link partner also sees the LFPS.
12 EXTCAPSUPTEN	External extended capability support enable When set, this bit enables extended capabilities to be implemented outside the core. A read to the first DWORD of the last internal extended capability (the "xHCI supported protocol capability for USB 3.0 when the Debug Capability is not enabled, or the "Debug Capability" when it is enabled) returns a value of 4 in the Next Capability Pointer bit. This indicates to software that there is another capability for DWORDs after this capability (for example, at address N+16 where N is the address of this DWORD). If enabled, an external address decoder that snoops the xHC slave interface needs to be implemented. If it

Table continues on the next page...

Field	Function
	sees an access to N+16 or greater, the slave access is re-routed to a piece of hardware which returns the external capability pointer register of the new capability and also handles reads/writes to this new capability and the side effects. If disabled, a read to the first DWORD of the last internal extended capability returns 0 in the 'Next Capability Pointer' bit. This indicates there are no more capabilities.
11 INSRTEXTRFSBODI	Insert extra delay between FS bulk OUT transactions Some FS devices are slow to receive bulk OUT data and can get stuck when there are consecutive bulk OUT transactions with short inter-transaction delays. This bit is used to control whether the host inserts extra delay between consecutive bulk OUT transactions to a FS endpoint. 0b - Host doesn't insert extra delay between consecutive bulk OUT transactions to a FS Endpoint 1b - Host inserts about 12 µs extra delay between consecutive bulk OUT transactions to a FS Endpoint to work around the device issue. Setting this bit to 1 reduces the bulk OUT transfer performance for most of the FS devices.
10-9 DTCT	Device timeout coarse tuning This bit is a host mode parameter which determines how long the host waits for a response from device before considering a timeout. The core first checks the DTCT value. If it is 0, then the timeout value is defined by the DTFT. If it is non-zero, then it uses the following Timeout values: 00b - 0 µs -> use DTFT value instead 01b - 500 µs 10b - 1.5 µs 11b - 6.5 µs
8-0 DTFT	Device timeout fine tuning This bit is a host mode parameter which determines how long the host waits for a response from device before considering a timeout. For DTFT bit to take effect, DTCT must be set to 2'b00. The DTFT value is the number of 125 MHz clocks * 256 to count before considering a device timeout. For the 125 MHz clock (8 ns period), this is calculated as follows- (DTFT value) * 256 * (8 ns) Quick Reference: • if DTFT = 0x2, 2*256*8 = 4 µs timeout • if DTFT = 0x5, 5*256*8 = 10 µs timeout • if DTFT = 0xA, 10*256*8 = 20 µs timeout • if DTFT = 0x10, 16*256*8 = 32 µs timeout • if DTFT = 0x19, 25*256*8 = 51 µs timeout • if DTFT = 0x31, 49*256*8 = 100 µs timeout • if DTFT = 0x62, 98*256*8 = 200 µs timeout

28.2.19 Global SoC bus error address register low (GBUSERRADDRLO)

28.2.19.1 Offset

Register	Offset
GBUSERRADDRLO	C130h

28.2.19.2 Function

When the AXI master bus returns "Error" response, the "SoC Bus Error" is generated. In the host mode, the host_system_err port indicates this condition. In addition, it is also indicated in the USBSTS[HSE] bit. In the device mode, the GSTS[BUSERRADDRVLD] bit is the only indication of the SoC bus error.

Due to the nature of AXI, it is possible that multiple AXI transactions are active at a time. The USB 3.0 controller does not keep track of the start address of all outstanding transactions. Instead, it keeps track of the start address of the DMA transfer associated with all active transactions. It is this address that is reported in the GBUSERRADDR when a bus error occurs.

For example, if the USB 3.0 controller initiates a DMA transfer to write 1K of packet data starting at buffer address 0xABCD0000, and this DMA is broken up into multiple 256B bursts on the AXI, then if a bus error occurs on any of these associated AXI transfers, the GBUSERRADDR reflects the DMA start address of 0xABCD0000 regardless of which AXI transaction received the error.

28.2.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.19.4 Fields

Field	Function
31-0 BUSERRADDR	Bus address - low This 64-bit register contains the lower 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS[BUSERRADDRVLD] bit is 1. It can only be cleared by resetting the core.

28.2.20 Global SoC bus error address register high (GBUSERRADDRHI)

28.2.20.1 Offset

Register	Offset
GBUSERRADDRHI	C134h

28.2.20.2 Function

It represents the remanining bits of global SoC bus error address register.

28.2.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUSERRADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.20.4 Fields

Field	Function
31-0 BUSERRADDR	Bus address - high This 64-bit register contains the higher 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS[BUSERRADDRVLD] bit is 1. It can only be cleared by resetting the core.

28.2.21 Global SS port to bus instance mapping register - low (GPRTBIMAPLO)

28.2.21.1 Offset

Register	Offset
GPRTBIMAPLO	C138h

28.2.21.2 Function

This is an alternate register for the GPRTBIMAP register.

NOTE

For reset values, refer to the corresponding values in the GPRTBIMAP register.

28.2.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.21.4 Fields

Field	Function
31-4 —	Reserved
3-0 BINUM1	SS USB instance number for port. Value set as 0.

28.2.22 Global SS port to bus instance mapping register - high (GPRTBIMAPHI)

28.2.22.1 Offset

Register	Offset
GPRTBIMAPHI	C13Ch

28.2.22.2 Function

This is an alternate register for the GPRTBIMAP register.

NOTE

For reset values, refer to the corresponding values in the GPRTBIMAP register.

28.2.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												BINUM9			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.22.4 Fields

Field	Function
31-4 —	Reserved
3-0	SS USB instance number for port 9.

Field	Function
BINUM9	

28.2.23 Global hardware parameters register 0 (GHWPARAMS0)

28.2.23.1 Offset

Register	Offset
GHWPARAMS0	C140h

28.2.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_AWIDTH								DWC_USB3_SDWIDTH							
W																
Reset	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_MDWIDTH								DWC_USB3_SBUS_TYPE		DWC_USB3_MBUS_TYPE			DWC_USB3_MODE		
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0

28.2.23.3 Fields

Field	Function
31-24	Master/Slave address bus width: 64 bit

Table continues on the next page...

Field	Function
DWC_USB3_AWIDTH	
23-16 DWC_USB3_SDWIDTH	Slave bus (Register access bus) data bus width: 32 bit
15-8 DWC_USB3_MDWIDTH	Master bus (DMA bus) data bus width It selects the data bus width of the master bus interface. 33-bit option is used only for Hub configuration. The possible values are: • 32 32-bits • 33 33-bits • 64 64-bits • 128 128-bits
7-6 DWC_USB3_SBUS_TYPE	Slave bus (Register access bus) interface type It selects the chip slave bus interface type. The slave bus is used for register programming. The settings not shown are reserved. 00b - AHB
5-3 DWC_USB3_MBUS_TYPE	Master bus (DMA bus) interface type It selects the chip master bus interface type. The master bus is used for DMA. The settings not shown are reserved. 001b - AXI
2-0 DWC_USB3_MODE	Mode of operation It selects the controller mode for USB 3.0. **NOTE:** It is configurable based on license(s) purchased. The settings not shown are resereved. 010b - DRD

28.2.24 Global hardware parameters register 1 (GHWPARAMS1)

28.2.24.1 Offset

Register	Offset
GHWPARAMS1	C144h

28.2.24.2 Function

This register contains the hardware configuration options.

28.2.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_EN_DBC	DWC_USB3_RM_OPT_FEATURES	Reserved	DWC_USB3_RAM_BUS_CLKS_SYNC	DWC_USB3_MAC_RAM_CLKS_SYNC	DWC_USB3_MAC_PHY_CLKS_SYNC	DWC_USB3_EN_PWROPT		DWC_USB3_SPRAM_TYP	DWC_USB3_NUM_RAMS		DWC_USB3_DEVICE_NUM_INT				
W																
Reset	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_DEVICE_NUM_INT	DWC_USB3_ASPACEWIDTH			DWC_USB3_REQINFOWIDTH			DWC_USB3_DATAINFOWIDTH			DWC_USB3_BURSTWIDTH			DWC_USB3_IDWIDTH1		
W																
Reset	1	1	0	0	1	0	0	1	0	0	1	1	1	0	1	1

28.2.24.4 Fields

Field	Function
31 DWC_USB3_EN_DBC	Enables xHCI debug capability 0b - No 1b - Yes
30	It specifies whether to remove optional features.

Table continues on the next page...

Field	Function
DWC_USB3_RM_OPT_FEATURES	When this parameter is enabled, the user ID register, General Purpose Input/Output ports, and SOF toggle and counter ports are removed. 0b - No 1b - Yes
29 —	Reserved
28 DWC_USB3_RAM_BUS_CLKS_SYNC	It specifies whether the RAM clock and the Bus clock are synchronous to each other. 0b - No 1b - Yes
27 DWC_USB3_MAC_RAM_CLKS_SYNC	It specifies whether the MAC clock and the RAM clock are synchronous to each other. 0b - No 1b - Yes
26 DWC_USB3_MAC_PHY_CLKS_SYNC	It specifies whether the MAC clock and the PHY clock are synchronous to each other. 0b - No 1b - Yes
25-24 DWC_USB3_EN_PWROPT	Power optimization mode It specifies the power optimization mode. If clock gating only is selected, RAM and PHY clocks are gated when the core is inactive during U1, U2, or U3 states. 00b - No power optimization 01b - Clock gating only 10b - Reserved 11b - Reserved
23 DWC_USB3_SPRAM_TYP	Synchronous static RAM type It selects the FIFO synchronous static RAM type. 0b - 2-port RAM (2Port-RAM) 1b - Single-port RAM (SPRAM)
22-21 DWC_USB3_NUM_RAMS	Number of RAMs It selects the number of RAMs. The possible values are 1, 2 and 3.
20-15 DWC_USB3_DEVICE_NUM_INT	Number of device mode event buffers It selects the number of event buffers in device mode. The possible values are 1, 2,..., and 32.
14-12 DWC_USB3_ASPACEWIDTH	It selects the address space port width of the master and slave bus interfaces. The possible values are 1, 2, 3, 4, 5 and 6.
11-9 DWC_USB3_REQINFOWIDTH	It selects the request/response info port width of the master and slave bus interfaces. The possible values are 4, 5 and 6.
8-6	It selects the data info port width of the master and slave bus interfaces. The possible values are 1, 2, 3, 4, 5 and 6.

Table continues on the next page...

Field	Function
DWC_USB3_DATAINFOWIDTH	
5-3 DWC_USB3_BURSTWIDTH	DWC_USB3_BURSTWIDTH 1 It selects the burst port width of the master and slave bus interfaces. The possible values are 1,2, 3, 4, 5, 6, 7 and 8.
2-0 DWC_USB3_IDWIDTH1	Master ID port width It selects the ID port width of the master bus interface. This parameter limits the number of pipelined AXI transfers. The GSBUSCFG1[PIPETRANSLIMIT] bit can only be programmed to a value less than or equal to two to the power of DWC_USB3_IDWIDTH. The possible values are 4, 5, 6, 7 and 8.

28.2.25 Global hardware parameters register 2 (GHWPARAMS2)

28.2.25.1 Offset

Register	Offset
GHWPARAMS2	C148h

28.2.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_USERID															
W																
Reset	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_USERID															
W																
Reset	0	0	1	0	1	0	0	1	0	0	0	0	1	0	1	0

28.2.25.3 Fields

Field	Function
31-0 DWC_USB3_USERID	Global user ID (GUID) register's power-on-initialization value It specifies the global user ID (GUID) register's power-on-initialization value. The value is set as 32'h130290A.

28.2.26 Global hardware parameters register 3 (GHWPARAMS3)

28.2.26.1 Offset

Register	Offset
GHWPARAMS3	C14Ch

28.2.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	DWC_USB3_CACHE_TOTAL_XFER_RESOURCES								DWC_USB3_NUM_IN_EPS					DWC_USB3_NUM_EPS	
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_NUM_EPS				Reserved	DWC_USB3_VENDOR_CTL_INTERFACE	Reserved		DWC_USB3_HSPHY_DWIDTH		Reserved		DWC_USB3_HSPHY_INTERFACE		DWC_USB3_SSPHY_INTERFACE	
W																
Reset	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1

28.2.26.3 Fields

Field	Function
31	Reserved

Table continues on the next page...

Field	Function
—	
30-23 DWC_USB3_CACHE_TOTAL_XFER_RESOURCES	It selects the maximum number of transfer resources in the core. The value is set as 8.
22-18 DWC_USB3_NUM_IN_EPS	Number of device mode active IN endpoints It specifies the maximum number of device mode IN endpoints active at any time, including control endpoint 0, which is always present. The value is set as 4.
17-12 DWC_USB3_NUM_EPS	Number of device mode endpoints It specifies the number of device mode single directional endpoints, including OUT and IN endpoint. The value is set as 8.
11 —	Reserved
10 DWC_USB3_VENDOR_CTL_INTERFACE	The bit enables the UTMI+ PHY vendor control interface. The value is enabled and value is set as 1.
9-8 —	Reserved
7-6 DWC_USB3_HSPHY_DWIDTH	It specifies the data width of the UTMI+ PHY interface. All other settings are reserved. 10b - 8-/16-bits
5-4 —	Reserved
3-2 DWC_USB3_HSPHY_INTERFACE	It specifies the high-speed PHY interface(s). The value is set as 1 for UTMI+.
1-0 DWC_USB3_SSPHY_INTERFACE	It specifies the superSpeed PHY interface. The value is set as 1 for PIPE3. In USB 2.0 only mode, set to 0 else select the PIPE3 interface.

28.2.27 Global hardware parameters register 4 (GHWPARAMS4)

28.2.27.1 Offset

Register	Offset
GHWPARAMS4	C150h

28.2.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_BMU_LSP_DEPTH				DWC_USB3_BMU_PTL_DEPTH				DWC_USB3_EN_ISOC_SUPT	Reserved	Reserved	DWC_USB3_NUM_SS_USB_INSTANCES				Reserved
W																
Reset	0	1	0	0	0	1	1	1	1	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			Reserved							DWC_USB3_CACHE_TRBS_PER_TRANSFER					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

28.2.27.3 Fields

Field	Function
31-28 DWC_USB3_BMU_LSP_DEPTH	It specifies the depth of the BMU-LSP status buffer. The value is set as 4.
27-24 DWC_USB3_BMU_PTL_DEPTH	It specifies the depth of the BMU-PTL source/sink buffers. The value is set as 8.
23 DWC_USB3_EN_ISOC_SUPT	It enables isochronous endpoint capability. By default, the value is set to 1 to enable this capability.
22 —	Reserved
21 —	Reserved
20-17 DWC_USB3_NUM_SS_USB_INSTANCES	Number of SuperSpeed USB bus instances It specifies the number of SuperSpeed USB bus instances. The value is set as 1.
16-13 —	Reserved
12-6 —	Reserved
5-0 DWC_USB3_CACHE_TRBS_PER_TRANSFER	Number of cached TRBs per transfer It selects the number of transfer request blocks (TRBs) per transfer that can be cached within the core. The value is set as 4.

28.2.28 Global hardware parameters register 5 (GHWPARAMS5)

28.2.28.1 Offset

Register	Offset
GHWPARAMS5	C154h

28.2.28.2 Function

The register is read-only.

28.2.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				DWC_USB3_DFQ_FIFO_DEPTH						DWC_USB3_DWQ_FIFO_DEPTH					
W																
Reset	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_TXQ_FIFO_DEPTH						DWC_USB3_RXQ_FIFO_DEPTH						DWC_USB3_BMU_BUSGM_DEPTH			
W																
Reset	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0

28.2.28.4 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27-22 DWC_USB3_DFQ_FIFO_DEPTH	It specifies the size of the BMU descriptor fetch request queue. The specified depth is allocated in the data FIFO RAM and defines the number of descriptor fetch commands the scheduler can queue to the BMU. The value is set as 16.
21-16 DWC_USB3_DWQ_FIFO_DEPTH	It specifies the size of the BMU descriptor write queue. The specified depth is allocated in the data FIFO RAM (32-/64-/128-bit wide) and defines the number of descriptor write commands the scheduler can queue to the BMU. value is set as 32.
15-10 DWC_USB3_TXQ_FIFO_DEPTH	It specifies the size of the BMU Tx request queue. The specified depth is allocated in the data FIFO RAM (32-/64-/128-bit wide) and defines the number of Tx commands the scheduler can queue to the BMU. The value is set as 16.
9-4 DWC_USB3_RXQ_FIFO_DEPTH	It specifies the size of the BMU Rx request queue. The specified depth is allocated in the data FIFO RAM (32-/64-/128-bits wide) and defines the number of Rx commands the scheduler can queue to the BMU. The value is set as 16.
3-0 DWC_USB3_BMU_BUSGM_DEPTH	It specifies the depth of the BMU-BUSGM source/sink buffers. The FIFOs are 32-/64-/128-bit wide, matching the bus master data width. The value is set as 8.

28.2.29 Global hardware parameters register 6 (GHWPARAMS6)

28.2.29.1 Offset

Register	Offset
GHWPARAMS6	C158h

28.2.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_RAM0_DEPTH															
W																
Reset	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUSFLTRSSUPPORT	Reserved	Reserved	ADPSUPPORT	HNPSUPPORT	Reserved	Reserved		DWC_USB3_EN_FPGA	DWC_USB3_EN_DBG_PORTS	DWC_USB3_PSQ_FIFO_DEPTH					
W																
Reset	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0

28.2.29.3 Fields

Field	Function
31-16 DWC_USB3_RAM0_DEPTH	Total RAM0 depth. The value is set as 2308. It specifies the depth of RAM0. In device, host, and DRD configuration, RAM0 contains: • 3-RAM configuration: Descriptor cache • 2-RAM configuration: Descriptor cache and RXFIFOs • 1-RAM configuration: Descriptor cache, TXFIFOs, and RXFIFOs
15 BUSFLTRSSUPPORT	It specifies whether to add a filter for VBUS and ID related control inputs from the PHY. • UTMI+ PHY: This signal is from the PHY. 0b - No 1b - Yes
14 —	Reserved
13 —	Reserved
12 ADPSUPPORT	It enables internal ADP capability of the USB 3.0 core. When it is enabled, the core incorporates ADP controller logic and provides ADP control signals.

Table continues on the next page...

Field	Function
11 HNPSUPPORT	HNP support enabled The application uses this bit to determine the USB 3.0 core's HNP support. 0b - HNP support is not enabled 1b - HNP support is enabled
10 —	Reserved
9-8 —	Reserved
7 DWC_USB3_EN_FPGA	Hardware validation/driver development with an FPGA platform 0b - No 1b - Yes
6 DWC_USB3_EN_DBG_PORTS	It is used for FPGA hardware validation of the core. 0b - No 1b - Yes
5-0 DWC_USB3_PSQ_FIFO_DEPTH	It specifies the size of the BMU protocol status queue. The value is set as 32. The specified depth is allocated in the data FIFO RAM 32-/64-/128-bit wide) and defines the number of header and status DWORDs the PTL can queue to the LSP.

28.2.30 Global hardware parameters register 7 (GHWPARAMS7)

28.2.30.1 Offset

Register	Offset
GHWPARAMS7	C15Ch

28.2.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_RAM2_DEPTH															
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_RAM1_DEPTH															
W																
Reset	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	1

28.2.30.3 Fields

Field	Function
31-16 DWC_USB3_RAM2_DEPTH	Total RAM2 depth. It specifies the depth of RAM2. The value is set as 776.
15-0 DWC_USB3_RAM1_DEPTH	Total RAM1 depth. It specifies the depth of RAM1. The value is set as 1101.

28.2.31 Global high-speed port to bus instance mapping register - low (GPRTBIMAP_HSLO)

28.2.31.1 Offset

Register	Offset
GPRTBIMAP_HSLO	C180h

28.2.31.2 Function

This is an alternate register for the GPRTBIMAP_HS register.

NOTE

For reset values, refer to the corresponding values in the GPRTBIMAP_HS register.

28.2.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.31.4 Fields

Field	Function
31-4 —	Reserved
3-0 BINUM1	HS USB instance number for port 1. The value is set as 0.

28.2.32 Global high-speed port to bus instance mapping register - high (GPRTBIMAP_HSHI)

28.2.32.1 Offset

Register	Offset
GPRTBIMAP_HSHI	C184h

28.2.32.2 Function

This is an alternate register for the GPRTBIMAP_HS register.

28.2.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												BINUM9			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.32.4 Fields

Field	Function
31-4 —	Reserved
3-0 BINUM9	HS USB instance number for port 9.

28.2.33 Global USB2 PHY configuration register (GUSB2PHYCFG)

28.2.33.1 Offset

Register	Offset
GUSB2PHYCFG	C200h

28.2.33.2 Function

The SoC must program this register before starting any transaction.

28.2.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PHYSOFTRST	U2_FREECLK_EXISTS	Reserved					LSTRD			LSIPD			Reserved		
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							ENBLSLPM	0	SUSPENDUSB20	0		PHYIF	Reserved		
W																
Reset	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

28.2.33.4 Fields

Field	Function
31 PHYSOFTRST	UTMI PHY soft reset. It causes the usb2phy_reset signal to be asserted to reset a UTMI PHY.
30 U2_FREECLK_EXISTS	Specifies USB 2.0 PHY free-running PHY clock exists. The value is set as 1.
29-25 —	Reserved
24-22 LSTRD	LS turnaround time. This bit indicates the value of the Rx-to-Tx packet gap for LS devices. **NOTE:** This bit is applicable only in host mode. For normal operation, to work with most LS devices, the default value is set as 0 (2-bit times).

Table continues on the next page...

Field	Function
	The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3-bit times of inter-packet gap to work correctly. Include the PHY delays when programming the LSIPD/LSTRDTIM values. For example, if the PHY's TxEndDelay in LS mode is 30 UTMI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement. 000b - 2-bit times 001b - 2.5-bit times 010b - 3-bit times 011b - 3.5-bit times 100b - 4-bit times 101b - 4.5-bit times 110b - 5-bit times 111b - 5.5-bit times
21-19 LSIPD	LS inter-packet time. This bit indicates the value of Tx-to-Tx packet gap for LS devices. **NOTE:** This bit is applicable only in host mode. For normal operation to work with most LS devices the bit is set to 2 (3-bit times). The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3-bit times of inter-packet gap to work correctly. Include the PHY delays when programming the LSIPD/LSTRDTIM values. For example, if the PHY's TxEndDelay in LS mode is 30 UTMI clocks, then subtract this delay (~1 LS bit time) from the device's delay requirement. 000b - 2-bit times 001b - 2.5-bit times 010b - 3-bit times 011b - 3.5-bit times 100b - 4-bit times 101b - 4.5-bit times 110b - 5-bit times 111b - 5.5-bit times
18-9 —	Reserved
8 ENBLSLPM	Enable utmi_sleep_n and utmi_l1_suspend_n. The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state. 0b - utmi_sleep_n and utmi_l1_suspend_n assertion from the core is not transferred to the external PHY 1b - utmi_sleep_n and utmi_l1_suspend_n assertion from the core is transferred to the external PHY
7 —	Reserved This bit should always read 0.
6 SUSPENDUSB20	Suspend USB2.0 HS/FS/LS PHY When set, USB2.0 PHY enters suspend mode if suspend conditions are valid. If it is set to 1, then the application should clear this bit after power-on-reset. Application needs to set it to 1 after the core initialization is completed. **NOTE:** In host mode, on reset, this bit is set to 1. Software can override this bit after reset.
5-4 —	Reserved This bit should always read 0.
3 PHYIF	PHY interface

Table continues on the next page...

Field	Function
	If UTMI+ is selected, the application uses this bit to configure the core to support a UTMI+ PHY with an 8-bit or 16-bit interface. 0b - 8-bit interface 1b - 16-bit interface
2-0 —	Reserved

28.2.34 Global USB 3.0 PIPE control register (GUSB3PIPECTL)

28.2.34.1 Offset

Register	Offset
GUSB3PIPECTL	C2C0h

28.2.34.2 Function

The application uses this register to configure the USB3 PHY and PIPE interface.

28.2.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PHYSOFTRST	Reserved	U2SSINACTP3OK	DISRXDETP3	0		U1U2EXITFAIL_TO_RECOV	Reserved	Reserved				Reserved		0	DATWIDTH
W								1	0				1			
Reset	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATWIDTH	0	Reserved		0		Reserved			TX_SWING	TX_MARGIN			TX_DE_EPPHASIS		ELASTIC_BUFFER_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

28.2.34.4 Fields

Field	Function
31 PHYSOFTRST	USB3 PHY soft reset. After setting this bit to 1, the software needs to clear this bit.
30 —	Reserved
29 U2SSINACTP3OK	P3 OK for U2/SSInactive 0b - During link state U2/SS.Inactive, put PHY in P2 (default) 1b - During link state U2/SS.Inactive, put PHY in P3
28 DISRXDETP3	Disabled receiver detection in P3 0b - If PHY is in P3 and core needs to perform receiver detection, the core performs receiver detection in P3 (default) 1b - If PHY is in P3 and core needs to perform receiver detection, the core changes PHY power state to P2 and then perform receiver detection. After receiver detection, the core changes PHY power state to P3
27-26	Reserved

Table continues on the next page...

Field	Function
—	The bit should always read 0.
25 U1U2EXITFAIL_TO_RECOV	U1U2exitfail to recovery When set, and U1/U2 LFPS handshake fails, the LTSSM transitions from U1/U2 to recovery instead of SS inactive. If recovery fails, then the LTSSM can enter SS.Inactive. This is an enhancement only. It prevents interoperability issue if the remote link does not do proper handshake.
24 —	Reserved This bit should be always set to 1.
23-20 —	Reserved
19-18 —	Reserved This bit is read-only and should be always set to 1.
17 SUSPENDENABLE	Suspend USB3.0 SS PHY When set, and if suspend conditions are valid, the USB 3.0 PHY enters suspend mode. The value is set as 0.
16-15 DATWIDTH	PIPE data width One clock after reset, these bits receive the value. 00b - 32 bits 01b - 16 bits 10b - 8 bits
14 —	Reserved This bit should always read 0.
13-12 —	Reserved
11-10 —	Reserved This bit should always read 0.
9-7 —	Reserved
6 TX_SWING	Tx swing. Refer to the PIPE3 specification.
5-3 TX_MARGIN	Tx margin[2:0]. Refer to Table 5-3 of the PIPE3 specification.
2-1 TX_DE_EPPHASIS	Tx deemphasis The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode. (Refer to Table 5-3 of the PIPE3 specification.)
0 ELASTIC_BUFFER_MODE	Elastic buffer mode

28.2.35 Global transmit FIFO size register (GTXFIFOSIZ_0 - GTXFIFOSIZ_3)

28.2.35.1 Offset

Register	Offset
GTXFIFOSIZ_0	C300h
GTXFIFOSIZ_1	C310h
GTXFIFOSIZ_2	C320h
GTXFIFOSIZ_3	C330h

28.2.35.2 Function

This register specifies the RAM start address and depth

Figure 28-3. GTxFIFOSIZn and GRxFIFOSIZn usage in DRD mode without debug capability

28.2.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TXFSTADDR_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXFDEP_N															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

28.2.35.4 Fields

Field	Function
31-16 TXFSTADDR_N	Transmit FIFOn RAM start address This bit contains the memory start address for TXFIFOn in MDWIDTH-bit words.
15-0 TXFDEP_N	TXFIFO depth This bit contains the depth of TXFIFOn in MDWIDTH-bit words. Minimum value: 32 Maximum value: 32,768

28.2.36 Global receive FIFO size register (GRXFIFOSIZ_0 - GRXFIFOSIZ_2)

28.2.36.1 Offset

Register	Offset
GRXFIFOSIZ_0	C380h
GRXFIFOSIZ_1	C390h
GRXFIFOSIZ_2	C3A0h

28.2.36.2 Function

This register specifies the RAM start address and depth.

28.2.36.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RXFSTADDR_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RXFDEP_N															
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1

28.2.36.4 Fields

Field	Function
31-16 RXFSTADDR_N	RXFIFOn RAM start address This bit contains the memory start address for RXFIFOn in MDWIDTH-bit words.
15-0 RXFDEP_N	RXFIFO depth This bits contains the depth of RXFIFOn in MDWIDTH-bit words. Minimum value: 32 Maximum value: 16,384

28.2.37 Global event buffer address (low) register (GEVNTADRLO)

28.2.37.1 Offset

Register	Offset
GEVNTADRLO	C400h

28.2.37.2 Function

This register holds the event buffer DMA address pointer. Software must initialize this address once during power-on-initialization. Software must not change the value of this register after it is initialized. Software must only use the GEVNTCOUNT register for event processing.

28.2.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTADRLO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTADRLO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.37.4 Fields

Field	Function
31-0 EVNTADRLO	Event buffer address Holds the lower 32 bits of start address of the external memory for the event buffer. During operation, hardware does not update this address.

28.2.38 Global event buffer address (high) register (GEVNTADRHI)

28.2.38.1 Offset

Register	Offset
GEVNTADRHI	C404h

28.2.38.2 Function

This register holds the event buffer DMA address pointer. Software must initialize this address once during power-on-initialization. Software must not change the value of this register after it is initialized. Software must only use the GEVNTCOUNT register for event processing.

28.2.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTADRHI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTADRHI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.38.4 Fields

Field	Function
31-0 EVNTADRHI	Event buffer address Holds the higher 32 bits of start address of the external memory for the event buffer. During operation, hardware does not update this address.

28.2.39 Global event buffer size register (GEVNTSIZ)

28.2.39.1 Offset

Register	Offset
GEVNTSIZ	C408h

28.2.39.2 Function

This register holds the event buffer size and the event interrupt mask bits. During power-on-initialization, software must initialize the size with the number of bytes allocated for the event buffer. The event interrupt mask bit masks the interrupt, but events are still queued. After configuration, software must preserve the event buffer size value when changing the event interrupt mask.

28.2.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTINTRPTMASK	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVENTSIZ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.39.4 Fields

Field	Function
31 EVNTINTRPTMASK	Event interrupt mask When set to 1, this prevents the interrupt from being generated. However, even when the mask is set, the events are queued.
30-16 —	Reserved
15-0 EVENTSIZ	Event buffer size in bytes Holds the size of the event buffer in bytes; must be a multiple of four. This is programmed by software once during initialization. The minimum size of the event buffer is 32 bytes.

28.2.40 Global event buffer count register (GEVNTCOUNT)

28.2.40.1 Offset

Register	Offset
GEVNTCOUNT	C40Ch

28.2.40.2 Function

This register holds the number of valid bytes in the event buffer. During initialization, software must initialize the count by writing 0 to the event count bit. Each time the hardware writes a new event to the event buffer, it increments this count. Most events are four bytes, but some events may span over multiple four byte entries. Whenever the count is greater than zero, the hardware raises the corresponding interrupt line (depending on the GEVNTSIZ[EVNTINTRPTMASK] bit). On an interrupt, software processes one or more events out of the event buffer. Afterwards, software must write the event count bit with the number of bytes it processed.

Clock crossing delays may result in the interrupt's continual assertion after software acknowledges the last event. Therefore, when the interrupt line is asserted, software must read the GEVNTCOUNT register and only process events if the GEVNTCOUNT is greater than 0.

28.2.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTCOUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.40.4 Fields

Field	Function
31-16 —	Reserved
15-0 EVNTCOUNT	Event count When read, returns the number of valid events in the event buffer (in bytes). When written, hardware decrements the count by the value written. The interrupt line remains high when count is not 0.

28.2.41 Global hardware parameters register 8 (GHWPARAMS8)

28.2.41.1 Offset

Register	Offset
GHWPARAMS8	C600h

28.2.41.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DWC_USB3_DCACHE_DEPTH_INFO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DWC_USB3_DCACHE_DEPTH_INFO															
W																
Reset	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0

28.2.41.3 Fields

Field	Function
31-0	The read-only value defines the minimum RAM0 requirement. The value of the bit is set as 2308.

Field	Function
DWC_USB3_DCACHE_DEPTH_INFO	

28.2.42 Global device TXFIFO DMA priority register (GTXFIFOPRIDEV)

28.2.42.1 Offset

Register	Offset
GTXFIFOPRIDEV	C610h

28.2.42.2 Function

This register specifies the relative DMA priority level among the Device TXFIFOs (one per IN endpoint). Each register bit[n] controls the priority (1: high, 0: low) of each TXFIFO[n]. When multiple TXFIFOs compete for DMA service at a given time (that is, multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner:

1. High-priority TXFIFOs are granted access using round-robin arbitration
2. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full).

For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed.

When configuring periodic IN endpoints, software must set register bit[n]=1, where n is the TXFIFO assignment. This ensures that the DMA for isochronous or interrupt IN endpoints are prioritized over bulk or control IN endpoints. The register size corresponds to the number of Device IN endpoints.

NOTE

Since the device mode uses only one RXFIFO, there is no device RXFIFO DMA priority register.

28.2.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												GTXFIFOPRIDEV			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.42.4 Fields

Field	Function
31-4 —	Reserved
3-0 GTXFIFOPRIDEV	Device TXFIFO priority 0000b - Low (default) 0001b - High

28.2.43 Global host TXFIFO DMA priority register (GTXFIFOPRIHST)

28.2.43.1 Offset

Register	Offset
GTXFIFOPRIHST	C618h

28.2.43.2 Function

This register specifies the relative DMA priority level among the host TXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FS/LS). Each register bit[n] controls the priority (1: high, 0: low) of TXFIFO[n] within a speed group. When multiple TXFIFOs compete for DMA service at a given time (that is, multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner:

1. Among the FIFOs in the same speed group (SS or HS/FS/LS):
   a. High-priority TXFIFOs are granted access using round-robin arbitration
   b. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high- priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full).
2. The TX DMA arbiter prioritizes the SS speed group or HS/FS/LS speed group according to the ratio programmed in the GDMAHLRATIO register.

For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 3 USB bus instances (1 SS, 1 HS, and 1 FS/LS).

28.2.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													GTXFIFOPRIHST		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.43.4 Fields

Field	Function
31-3 —	Reserved
2-0 GTXFIFOPRIHST	Host TXFIFO priority 000b - Low priority (default) 001b - High priority

28.2.44 Global host RXFIFO DMA priority register (GRXFIFOPRIHST)

28.2.44.1 Offset

Register	Offset
GRXFIFOPRIHST	C61Ch

28.2.44.2 Function

This register specifies the relative DMA priority level among the host RXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FS/LS). Each register bit[n] controls the priority (1: high, 0: low) of RXFIFO[n] within a speed group. When multiple RXFIFOs compete for DMA service at a given time (that is, multiple RXQs contain RX DMA requests and their corresponding RXFIFOs have data available), the RX DMA arbiter grants access on a packet-basis in the following manner:

1. Among the FIFOs in the same speed group (SS or HS/FS/LS):
   a. High-priority RXFIFOs are granted access using round-robin arbitration
   b. Low-priority RXFIFOs are granted access using round-robin arbitration only after high-priority RXFIFOs have no further processing to do (that is, either the RXQs are empty or the corresponding RXFIFOs do not have the required data).
2. The RX DMA arbiter prioritizes the SS speed group or HS/FS/LS speed group according to the ratio programmed in the GDMAHLRATIO register.

For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed. This register is present only when the core is configured to operate in the host mode (includes DRD mode). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 3 USB bus instances (1 SS, 1 HS, and 1 FS/LS).

28.2.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													GRXFIFOPRIHST		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.44.4 Fields

Field	Function
31-3 —	Reserved
2-0 GRXFIFOPRIHST	Host RXFIFO priority 000b - Low priority (default) 001b - High priority

28.2.45 Global host FIFO DMA high-low priority ratio register (GDMAHLRATIO)

28.2.45.1 Offset

Register	Offset
GDMAHLRATIO	C624h

28.2.45.2 Function

This register specifies the relative priority of the SS FIFOs with respect to the HS/FS/LS FIFOs. The DMA arbiter prioritizes the HS/FS/LS round-robin arbiter group every DMA high-low priority ratio grants as indicated in the register separately for TX and RX.

To illustrate, consider that all FIFOs are requesting access simultaneously, and the ratio is 4. SS gets priority for 4 packets, HS/FS/LS gets priority for 1 packet, SS gets priority for 4 packets, HS/FS/LS gets priority for 1 packet, and so on.

If FIFOs from both speed groups are not requesting access simultaneously then,

- If SS got grants 4 out of the last 4 times, then HS/FS/LS get the priority on any future request.
- If HS/FS/LS got the grant last time, SS gets the priority on the next request.
- If there is a valid request on either SS or HS/FS/LS, a grant is always awarded; there is no idle. This register is present if the core is configured to operate in host mode.

28.2.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			HSTRXFIFO					Reserved			HSTTXFIFO				
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

28.2.45.4 Fields

Field	Function
31-13 —	Reserved
12-8 HSTRXFIFO	Host RXFIFO DMA high-low priority ratio
7-5 —	Reserved
4-0 HSTTXFIFO	Host TXFIFO DMA high-low priority ratio

28.2.46 Global frame length adjustment register (GFLADJ)

28.2.46.1 Offset

Register	Offset
GFLADJ	C630h

28.2.46.2 Function

It provides an option to override the fladj_30mhz_reg sideband signal.

28.2.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									Reserved	GFLADJ_REFCLK_FLADJ					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GFLADJ_REFCLK_FLADJ								GFLADJ_30MHZ_REG_SEL	Reserved	GFLADJ_30MHZ					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.46.4 Fields

Field	Function
31-23 —	Reserved This bit should always read 0.
22 —	Reserved
21-8 GFLADJ_REFCLK_FLADJ	This bit indicates the frame length adjustment to be applied when SOF/ITP counter is running on the ref_clk.
7 GFLADJ_30MHZ_REG_SEL	This bit selects whether to use a hard-coded value of 20h (32 decimal) or the value in GFLADJ[GFLADJ_30MHZ] to adjust the frame length for the SOF/ITP. 0b - The controller uses the hard coded value 20h (32 decimal) . which gives a SOF cycle time of 60000. 1b - The controller uses the value in GFLADJ[GFLADJ_30MHZ].
6 —	Reserved

Table continues on the next page...

Field	Function
5-0 GFLADJ_30MHZ	This bit indicates the value that is used for frame length adjustment when GFLADJ_30MHZ_REG_SEL = 1. Each step of this field's value corresponds to 16 high-speed bit times. The SOF cycle time (the number of SOF counter clock periods to generate a SOF microframe length) is equal to 59488 + the value in this field. When GFLADJ_30MHZ_REG_SEL = 0, a hard-coded value of 20h (32 decimal) is used, which gives a SOF cycle time of 60000. For details on how to set this value, refer to section 5.2.4, "Frame Length Adjustment Register (FLADJ)," of the xHCI Specification. 000000b - Frame length is 59488 HS bit times 000001b - Frame length is 59504 HS bit times 000010b - Frame length is 59520 HS bit times... 011111b - Frame length is 59984 HS bit times 100000b - Frame length is 60000 HS bit times... 111110b - Frame length is 60480 HS bit times 111111b - Frame length is 60496 HS bit times

28.2.47 Device configuration register (DCFG)

28.2.47.1 Offset

Register	Offset
DCFG	C700h

28.2.47.2 Function

This register configures the core in device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

28.2.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								IGMSTRMPP	LPMCAP	NUMP					INTRNUM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	INTRNUM				Reserved		DEVADDR							DEVSPD		
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0

28.2.47.4 Fields

Field	Function
31-24 —	Reserved
23 IGMSTRMPP	Ignore stream PP This bit only affects stream-capable bulk endpoints. When this bit is set to 0 and the controller receives a data packet with the Packet Pending (PP) bit set to 0 for OUT endpoints, or it receives an ACK with the NUMP bit set to 0 and PP set to 0 for IN endpoints, the core attempts to search for another stream (CStream) to initiate to the host. However, there are two situations where this behavior is not optimal: • When the host is setting PP=0 even though it has not finished the stream, or • When the endpoint on the device is configured with one transfer resource and therefore does not have any other streams to initiate to the host. When this bit is set to 1, the core ignores the Packet Pending bit for the purposes of stream selection and does not search for another stream when it receives DP(PP=0) or ACK(NUMP=0, PP=0). This can enhance the performance when the device system bus bandwidth is low or the host responds to the core's ERDY transmission very quickly.
22 LPMCAP	LPM capable The application uses this bit to control the USB 3.0 core LPM capabilities. If the core operates as a non-LPM-capable device, it cannot respond to LPM transactions. 0b - LPM capability is not enabled 1b - LPM capability is enabled
21-17 NUMP	Number of receive buffers This bit indicates the number of receive buffers to be reported in the ACK TP.

Table continues on the next page...

Field	Function
	The USB 3.0 controller uses this bit if GRXTHRCFG[USBRXPKTCNTSEL] is set to 0. The application can program this value based on RXFIFO size, buffer sizes programmed in descriptors, and system latency. For an OUT endpoint, this bit controls the number of receive buffers reported in the NUMP bit of the ACK TP transmitted by the core. **NOTE:** This bit is used in host mode when debug capability is enabled.
16-12 INTRNUM	Interrupt number Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated.
11-10 —	Reserved
9-3 DEVADDR	Device address The application must perform the following: • Program this bit after every SetAddress request. • Reset this bit to 0 after USB reset.
2-0 DEVSPD	Device speed Indicates the speed at which the application requires the core to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. 000b - High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 001b - Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 100b - SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz)

28.2.48 Device control register (DCTL)

28.2.48.1 Offset

Register	Offset
DCTL	C704h

28.2.48.2 Function

The table below lists the minimum duration under various conditions for which the soft disconnect (SFTDISCON) bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration.

Table 28-6. Minimum duration for soft disconnect

Operating speed	Device state	Minimum duration
SuperSpeed	Suspended	30 ms
SuperSpeed	Idle	30 ms
SuperSpeed	Transmit or Receive	30 ms
High-speed	Suspended	10 ms
High-speed	Idle	10 ms
High-speed	Not Idle or Suspended (Performing transactions)	10 ms
Full-speed/Low-speed	Suspended	10 ms
Full-speed/Low-speed	Idle	10 ms
Full-speed/Low-speed	Not idle or suspended (performing transactions)	10 ms

28.2.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	RUN_STOP	CSFTRST	Reserved	HIRDTHRES					LPM_NYET_THRES				Reserved		CRS	CSS
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved			INITU2ENA	ACCEPTU2ENA	INITU1ENA	ACCEPTU1ENA	ULSTCHNGREQ				TSTCTL				Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.48.4 Fields

Field	Function
31	The software writes 1 to this bit to start the device controller operation.

Table continues on the next page...

Field	Function
RUN_STOP	To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DSTS[DEVCTRLHLT] bit when the core is idle and the lower layer finishes the disconnect process. The RUN_STOP bit must be used in following cases as specified: 1. After power-on-reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. 2. The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. 3. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DCTL[ULSTCHNGREQ] to 5 before reasserting the RUN_STOP bit. The minimum duration of keeping this bit cleared is specified in Table 28-6. 4. When the USB or Link is in a lower power state and the two power rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the core power rail. After the software turns on the core power rail again and re-initializes the device controller, it must set this bit to start the device controller. For more details, see Low power operation
30 CSFTRST	Core soft reset Resets all clock domains as follows: • Clears the interrupts and all the CSRs except the following registers: GCTL, GUCTL, GSTS, GUID, GUSB2PHYCFG, GUSB3PIPECTL, DCFG, DCTL, DEVTEN, DSTS • All module state machines (except the SoC bus slave unit) are reset to the IDLE state, and all the TXFIFOs and the RXFIFO are flushed. • Any transactions on the SoC bus master are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately. The application can write this bit at any time to reset the core. This is a self- clearing bit; the core clears this bit after all necessary logic is reset in the core, which may take several clocks depending on the core's current state. Once this bit is cleared, the software must wait at least 3 PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and also when the PHY selection bits are dynamically changed in the USB configuration registers listed above. When the PHY is changed, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation.
29 —	Reserved

28-24
 	
HIRDTHRES
	HIRD threshold
	
	The core asserts output signals utmi_l1_suspend_n and utmi_sleep_n on the basis of this signal.
	
	• The core asserts utmi_l1_suspend_n to put the PHY into deep low-power mode in L1 when both of the following are true:
	  • HIRD value is greater than or equal to the value in DCTL[HIRD_THRES][3-0]
	  • HIRD_THRES[4] is set to 1'b1.
	• The core asserts utmi_sleep_n on L1 when one of the following is true:
	  • If the HIRD value is less than HIRD_Thres[3-0]
	  • HIRD_Thres[4] is set to 1'b0.
	
	NOTE: This bit must be set to 0 during SuperSpeed mode of operation.

23-20 LPM_NYET_THRES	When LPM errata is enabled: LPM NYET response threshold handshake response to LPM token specified by device application. Response depends on DCFG[LPMCAP].

Table continues on the next page...

Field	Function

	• DCFG[LPMCAP] is 1'b0 : The core always responds with timeout (that is, no response).
	• DCFG[LPMCAP] is 1'b1 : The core responds with an ACK on successful LPM transaction, which requires that all of the following are satisfied:
	  • There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR)
	  • A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL) .
	  • No data is pending in the Transmit FIFO and OUT endpoints not in flow controlled state (else NYET).
	  • The BESL value in the LPM token is less than or equal to LPM_NYET_THRES[3-0]

19-18 —	Reserved **NOTE:** Software should program bit 19 to 0.
17 CRS	Controller restore state This command is similar to the USBCMD[CRS] bit in host mode and initiates the restore process. When software sets this bit to 1, the controller immediately sets DSTS[RSS] to 1. When the controller has finished the restore process, it sets DSTS[RSS] to 0. **NOTE:** When read, this bit always returns 0.
16 CSS	Controller save state This command is similar to the USBCMD[CSS] bit in host mode and initiates the save process. When software sets this bit to 1, the controller immediately sets DSTS[SSS] to 1. When the controller has finished the save process, it sets DSTS[SSS] to 0. **NOTE:** When read, this bit always returns 0.
15-13 —	Reserved
12 INITU2ENA	Initiate U2 enable On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U2_ENABLE), and clears this bit when ClearFeature(U2_ENABLE) is received. If DCTL[ACCEPTU2ENA] is 0, the link immediately exits U2 state. 0b - May not initiate U2 (default) 1b - May initiate U2
11 ACCEPTU2ENA	Accept U2 enable On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command. 0b - Reject U2 except when Force_LinkPM_Accept bit is set (default) 1b - Core accepts transition to U2 state if nothing is pending on the application side
10 INITU1ENA	Initiate U1 enable On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U1_ENABLE), and clears this bit when ClearFeature(U1_ENABLE) is received. If DCTL[ACCEPTU1ENA] is 0, the link immediately exits U1 state. 0b - May not initiate U1 (default) 1b - May initiate U1
9 ACCEPTU1ENA	Accept U1 enable On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command. 0b - Core rejects U1 except when Force_LinkPM_Accept bit is set (default) 1b - Core accepts transition to U1 state if nothing is pending on the application side
8-5	USB/Link state change request

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>ULSTCHNGREQ</td><td>Software writes this bit to issue a USB/Link state change request. A change in this bit indicates a new request to the core. If software wants to issue the same request back-to-back, it must write a 0 to this bit between the two requests. The result of the state change request is reflected in the USB/Link state in DSTS. These bits are self-cleared on the MAC Layer exiting suspended state.

If software is updating other bits of the DCTL register and not intending to force any link state change, then it must write a 0 to this bit.

SS compliance mode is normally entered and controlled by the remote link partner. Refer to the USB3 specification. Alternatively, the local link can be directly forced into compliance mode, by resetting the SS link with the RUN_STOP bit set to 0. If you then write 10 to the USB/Link state change bit and 1 to RUN_STOP, the Link goes to compliance. Once you are in compliance, you may alternately write 0 and 10 to this bit to advance the compliance pattern.

In SS mode:
<table>
<tr><th>Value</th><th>Requested link state transition/action</th></tr>
<tr><td>0</td><td>No action</td></tr>
<tr><td>4</td><td>SS.Disabled</td></tr>
<tr><td>5</td><td>Rx.Detect</td></tr>
<tr><td>6</td><td>SS.Inactive</td></tr>
<tr><td>8</td><td>Recovery</td></tr>
<tr><td>Others</td><td>Reserved</td></tr>
</table>
In HS/FS/LS mode:
<table>
<tr><th>Value</th><th>Requested USB state transition</th></tr>
<tr><td>8</td><td>Remote wakeup request</td></tr>
<tr><td>Others</td><td>Reserved</td></tr>
</table>
The Remote wakeup request should be issued 2 µs after the device goes into suspend state (DSTS[USBLNKST] is 3 (refer to Device status register (DSTS)).</td></tr>
<tr><td>4-1

TSTCTL</td><td>Test control

The settings not defined are reserved.

0000b - Test mode disabled
0001b - Test_J mode
0010b - Test_K mode
0011b - Test_SE0_NAK mode
0100b - Test_Packet mode
0101b - Test_Force_Enable</td></tr>
<tr><td>0

—</td><td>Reserved</td></tr>
</table>

28.2.49 Device event enable register (DEVTEN)

28.2.49.1 Offset

Register	Offset
DEVTEN	C708h

28.2.49.2 Function

This register controls the generation of device-specific events. If an enable bit is set to 0, the event is not generated.

28.2.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved	VENDEVTSTRCVDEN	Reserved		ERRTICERREVTEN	Reserved	SOFTEVTEN	U3L2L1SUSPEN	Reserved	WKUPEVTEN	ULSTCNGEN	CONNECTDONEEVTEN	USBRSTEVTEN	DISSCONNEVTEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.49.4 Fields

Field	Function
31-14 —	Reserved
13 —	Reserved
12	Vendor device test LMP received rvent

Table continues on the next page...

Field	Function
VENDEVTSTRCVDEN	
11-10 —	Reserved
9 ERRTICERREVTEN	Erratic error event enable
8 —	Reserved
7 SOFTEVTEN	Start of (micro)frame enable
6 U3L2L1SUSPEN	U3/L2-L1 suspend event enable
5 —	Reserved
4 WKUPEVTEN	Resume/Remote wakeup detected event enable
3 ULSTCNGEN	USB/Link state change event enable
2 CONNECTDONEEVTEN	Connection done enable
1 USBRSTEVTEN	USB reset enable
0 DISSCONNEVTEN	Disconnect detected event enable

28.2.50 Device status register (DSTS)

28.2.50.1 Offset

Register	Offset
DSTS	C70Ch

28.2.50.2 Function

This register indicates the status of the device controller with respect to USB-related events.

28.2.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		Reserved	Reserved	Reserved		RSS	SSS	COREIDLE	DEVCTRLHLT	USBLNKST				RXFIFOEMPTY	SOFFN
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SOFFN													CONNECTSPD		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

28.2.50.4 Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 —	Save/Restore error This bit is currently not supported.
27-26 —	Reserved
25 RSS	Restore state status This bit is similar to the USBSTS[RSS] in host mode.

Table continues on the next page...

Field	Function
	When the controller has finished the restore process, it completes the command by setting DSTS[RSS] to 0.
24 SSS	Save state status This bit is similar to the USBSTS[SSS] in host mode. When the controller has finished the save process, it completes the command by setting DSTS[SSS] to 0.
23 COREIDLE	Core idle The bit indicates that the core finished transferring all RXFIFO data to system memory, writing out all completed descriptors, and all event counts are zero. **NOTE:** While testing for reset values, mask out the read value. This bit represents the changing state of the core and does not hold a static value.
22 DEVCTRLHLT	Device controller halted This bit is set to 0 when the DCTL[RUN_STOP] bit is set to 1. The core sets this bit to 1 when, after software sets RUN_STOP to 0, the core is idle and the lower layer finishes the disconnect process. When Halted =1, the core does not generate device events.
21-18 USBLNKST	USB/Link state In SS mode: LTSSM State 4'h0 U0 4'h1 U1 4'h2 U2 4'h3 U3 4'h4 SS_DIS 4'h5 RX_DET 4'h6 SS_INACT 4'h7 POLL 4'h8 RECOV 4'h9 HRESET 4'hA CMPLY 4'hB LPBK 4'hF Resume/Reset In HS/FS/LS mode. 4'h0 On state 4'h2 Sleep (L1) state 4'h3 Suspend (L2) state 4'h4 Disconnected state (default) Software must write 8 (Recovery) to the DCTL[ULSTCHNGREQ] bit to acknowledge the resume/reset request.
17 RXFIFOEMPTY	RXFIFO empty
16-3 SOFFN	Frame/microframe number of the received SOF When the core is operating at high-speed,

Table continues on the next page...

Field	Function
	[16:6] indicates the frame number [5:3] indicates the microframe number When the core is operating at high-speed, [16:14] is not used. Software can ignore these 3 bits [13:3] indicates the frame number
2-0 CONNECTSPD	Connected speed Indicates the speed at which the USB 3.0 core has come up after speed detection through a chirp sequence. Low-speed is not supported for devices using a UTMI+ PHY. 000b - High-speed (PHY clock is running at 30 MHz or 60 MHz) 001b - Full-speed (PHY clock is running at 30 MHz or 60 MHz) 010b - Low-speed (PHY clock is running at 6 MHz) 011b - Full-speed (PHY clock is running at 48 MHz) 100b - SuperSpeed (PHY clock is running at 125 MHz or 250 MHz)

28.2.51 Device generic command parameter register (DGCMDPAR)

28.2.51.1 Offset

Register	Offset
DGCMDPAR	C710h

28.2.51.2 Function

This register indicates the device command parameter. This must be programmed before or along with the device command. The available device commands are listed in Device generic command register (DGCMD).

28.2.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.51.4 Fields

Field	Function
31-0 PARAMETER	Parameter for device command

28.2.52 Device generic command register (DGCMD)

28.2.52.1 Offset

Register	Offset
DGCMD	C714h

28.2.52.2 Function

This register enables software to program the core using a single generic command interface to send link management packets and notifications. This register contains command, control, and status bits relevant to the current generic command, while the DGCMDPAR register provides the command parameter.

Any commands that are not present in the following table are considered Reserved.

Table 28-7. Device generic command types

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>02h</td><td>Set periodic parameters

Parameter[9:0] (SystemExitLatency): Software should set this to the same value programmed by the host through the Set SEL device request, in microseconds.

The Set SEL control transfer has 6 bytes of data and contains 4 values.
<table>
<tr><th>Offset</th><th>Name</th><th>Meaning</th></tr>
<tr><td>0</td><td>U1SEL</td><td>Time in µs for U1 system exit latency</td></tr>
<tr><td>1</td><td>U1PEL</td><td>Time in µs for U1 device to host exit latency</td></tr>
<tr><td>2</td><td>U2SEL</td><td>Time in µs for U2 system exit latency</td></tr>
<tr><td>4</td><td>U2PEL</td><td>Time in µs for U2 device to host exit latency</td></tr>
</table>
If the device is enabled for U1 and U2, then the U2PEL should be programmed. If the device is enabled only for U1, then U1PEL should be programmed into this parameter.

If the value is greater than 125 µs, then the software must program a value of 0 into this register.</td></tr>
<tr><td>04h</td><td>Set scratchpad buffer array address low

This command sets bits [31:0] of the external address of the scratchpad buffer array used for save/restore.

If either this command or command 05h is issued while the controller is stopped (RUN_STOP = 0), the CMDIOC bit must be set to 0.

The device scratchpad buffer array has the same format as the xHCI scratchpad buffer array; it contains an array of 64-bit pointers to data buffers that is used to save the controller's state.</td></tr>
<tr><td>05h</td><td>Set scratchpad buffer array address high

This command sets bits [63:32] of the external address of the scratchpad array buffer used for save/restore.

If either this command or command 04h is issued while the controller is stopped (RUN_STOP = 0), the CMDIOC bit must be set to 0.

The device scratchpad buffer array has the same format as the xHCI scratchpad buffer array; it contains an array of 64-bit pointers to data buffers that is used to save the controller's state.</td></tr>
<tr><td>07h</td><td>Transmit device notification

This command allows any device notification to be transmitted, using the notification type and notification parameters specified in the DGCMDPAR register.

DGCMDPAR[3:0] = Notification type

DGCMDPAR[31:4] = Notification parameters, depends on the notification type

For example, to transmit a function wake, software sets DGCMDPAR[3:0] to 1, and DGCMDPAR[10:4] to the interface number.

This bit relates to the "Notification Type Specific" bit in a device notification transaction packet as described in Section 8.5.6 of the USB3 Specification. The following bits of the DGCMDPAR register have been put into the corresponding DWORD described in Section 8.5.6 of the USB3 Specification:

DGCMDPAR[3:0] into DWORD1[7:4] (Notification Type)

DGCMDPAR[27:4] into DWORD1[31:8] (Notification Type Specific)</td></tr>
</table>

Table continues on the next page...

Table 28-7. Device generic command types (continued)

Command	Description
	DGCMDPAR[31:28] into DWORD2[3:0] (Notification Type Specific) There is one exception for the bus interval adjustment device notification: DGCMDPAR[19:4] represents the bus interval adjustment bit; however, in the USB3 specification, the bus interval adjustment bit is actually at 31:16 of DWORD1.
09h	Selected FIFO flush Parameter[4:0] = FIFO number Parameter[5] = 1 for TXFIFO or 0 for RXFIFO
0Ah	All FIFO flush No parameter
0Ch	Set endpoint NRDY Issuing this command makes the core think that the given endpoint is in an NRDY state. If there are buffers available in that endpoint, the core immediately transmits an ERDY. Parameter[4:0] = Physical endpoint number

28.2.52.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDSTATUS				Reserved	CMDACT	Reserved	CMDIOC	CMDTYP							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.52.4 Fields

Field	Function
31-16 —	Reserved
15-12 CMDSTATUS	Command status 0000b - Indicates command success

Table continues on the next page...

Field	Function
	0001b - CmdErr, indicates that the device controller encountered an error while processing the command
11 —	Reserved
10 CMDACT	Command active The software sets this bit to 1 to enable the device controller to execute the generic command. The device controller sets this bit to 0 after executing the command.
9 —	Reserved
8 CMDIOC	Command interrupt on complete When this bit is set, the device controller issues a generic command completion event after executing the command. Note that this interrupt is mapped to DCFG[INTRNUM]. **NOTE:** This bit must not set to 1 if the DCTL[RUN_STOP] bit is 0.
7-0 CMDTYP	Command type Specifies the type of command the software driver is requesting the core to perform. 8'h0 Reserved 8'h1 Set endpoint configuration - 64-bit or 96-bit parameter 8'h2 Set endpoint transfer resource configuration - 32-bit parameter 8'h3 Get endpoint state - no parameter needed 8'h5 Clear stall (see set stall) - no parameter needed 8'h6 Start transfer - 64-bit parameter 8'h7 Update transfer - no parameter needed 8'h8 End transfer - no parameter needed 8'h9 Start new configuration - no parameter needed

28.2.53 Device active USB endpoint enable register (DALEPENA)

28.2.53.1 Offset

Register	Offset
DALEPENA	C720h

28.2.53.2 Function

This register indicates whether a USB endpoint is active in a given configuration or interface.

28.2.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								USBACTEP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.53.4 Fields

Field	Function
31-8 —	Reserved
7-0 USBACTEP	USB active endpoints This bit indicates if a USB endpoint is active in the current configuration and interface. Bit[0]: USB EP0-OUT Bit[1]: USB EP0-IN Bit[2]: USB EP1-OUT Bit[3]: USB EP1-IN Bit[4]: USB EP2-OUT Bit[5]: USB EP2-IN Bit[6]: USB EP3-OUT Bit[7]: USB EP3-IN The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USB reset. Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set these bits.

28.2.54 Device physical endpoint-n command parameter 2 register (DEPCMDPAR2_0 - DEPCMDPAR2_7)

28.2.54.1 Offset

Register	Offset
DEPCMDPAR2_0	C800h
DEPCMDPAR2_1	C810h
DEPCMDPAR2_2	C820h
DEPCMDPAR2_3	C830h
DEPCMDPAR2_4	C840h
DEPCMDPAR2_5	C850h
DEPCMDPAR2_6	C860h
DEPCMDPAR2_7	C870h

28.2.54.2 Function

This register indicates the physical endpoint command parameter 2. It must be programmed before issuing the command.

28.2.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.54.4 Fields

Field	Function
31-0 PARAMETER	Parameter 2

28.2.55 Device physical endpoint-n command parameter 1 register (DEPCMDPAR1_0 - DEPCMDPAR1_7)

28.2.55.1 Offset

Register	Offset
DEPCMDPAR1_0	C804h
DEPCMDPAR1_1	C814h
DEPCMDPAR1_2	C824h
DEPCMDPAR1_3	C834h
DEPCMDPAR1_4	C844h
DEPCMDPAR1_5	C854h
DEPCMDPAR1_6	C864h
DEPCMDPAR1_7	C874h

28.2.55.2 Function

This register indicates the physical endpoint command parameter 1. It must be programmed before issuing the command.

28.2.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.55.4 Fields

Field	Function
31-0 PARAMETER	Parameter 1

28.2.56 Device physical endpoint-n command parameter 0 register (DEPCMDPAR0_0 - DEPCMDPAR0_7)

28.2.56.1 Offset

Register	Offset
DEPCMDPAR0_0	C808h
DEPCMDPAR0_1	C818h
DEPCMDPAR0_2	C828h
DEPCMDPAR0_3	C838h
DEPCMDPAR0_4	C848h
DEPCMDPAR0_5	C858h
DEPCMDPAR0_6	C868h
DEPCMDPAR0_7	C878h

28.2.56.2 Function

This register indicates the physical endpoint command parameter 0. This must be programmed before or along with the command. For commands needing only one 32-bit parameter, this register must be programmed with the command register.

28.2.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.56.4 Fields

Field	Function
31-0 PARAMETER	Parameter 0

28.2.57 Device physical endpoint-n command register (DEPCMD_0 - DEPCMD_7)

28.2.57.1 Offset

Register	Offset
DEPCMD_0	C80Ch
DEPCMD_1	C81Ch
DEPCMD_2	C82Ch
DEPCMD_3	C83Ch

Table continues on the next page...

Register	Offset
DEPCMD_4	C84Ch
DEPCMD_5	C85Ch
DEPCMD_6	C86Ch
DEPCMD_7	C87Ch

28.2.57.2 Function

This register enables software to issue physical endpoint-specific commands. This register contains command, control, and status bits relevant to the current generic command, while the DEPCMDPAR[2:0]n registers provide command parameters and return status information.

Several bits (including command type) are write-only, so their read values are undefined. After power-on, prior to issuing the first endpoint command, the read value of this register is undefined. In particular, the DGCMD[CMDACT] bit may be set after power-on. In this case, it is safe to issue an endpoint command.

For more details about the commands, refer Device physical endpoint-specific commands.

28.2.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COMMANDPARAM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDSTATUS				HIPRI_FORCERM	CMDACT	Reserved	CMDIOC	Reserved				CMDTYP			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.57.4 Fields

Field	Function

31-16
COMMANDPARAM
	Command parameters. When this register is written for:
	• Start transfer command:
	  • [31-16] StreamID. The USB StreamID assigned to this transfer
	• Start transfer command applied to an isochronous endpoint:
	  • [31-16] StartMicroFramNum Indicates the (micro) frame number to which the first TRB applies
	• For update transfer, end transfer, and start new configuration commands:
	  • [22-16] Transfer resource index (XferRscIdx). The hardware- assigned transfer resource index for the transfer, which was returned in response to the start transfer command. The application software-assigned transfer resource index for a start new configuration command.
	
	Event parameters (EventParam). When this register is read, refer to bits 31:16 in Table 28-16

15-12 CMDSTATUS	Command completion status Additional information about the completion of this command is available in this bit. The information is in the same format as bits 15-12 of the endpoint command complete event. Refer Table 28-16.
11 HIPRI_FORCERM	HighPriority/ForceRM HighPriority: Only valid for start transfer command ForceRM: Only valid for end transfer command ClearPendIN: Only valid for clear stall command; software sets this bit to clear any pending IN transaction (on that endpoint) stuck at the lower layers when a clear stall command is issued.
10 CMDACT	Command active Software sets this bit to 1 to enable the device endpoint controller to execute the generic command. The device controller sets this bit to 0 when the CMDSTATUS bit is valid and the endpoint is ready to accept another command. This does not imply that all the effects of the previously-issued command have taken place.
9 —	Reserved
8 CMDIOC	Command interrupt on complete When this bit is set, the device controller issues a generic endpoint command complete event after executing the command. Note that this interrupt is mapped to DEPCFG[INTRNUM]. When the DEPCFG command is executed, the command interrupt on completion goes to the interrupt pointed by the DEPCFG[INTRNUM] in the current command. **NOTE:** This bit must not set to 1 if the DCTL[RUN_STOP] bit is 0.
7-4 —	Reserved
3-0 CMDTYP	Command type Specifies the type of command the software driver is requesting the core to perform. 0000b - Reserved 0001b - Set endpoint configuration: 64-bit or 96-bit parameter 0010b - Set endpoint transfer resource configuration: 32-bit parameter 0011b - Get endpoint state: no parameter needed 0100b - Set stall: no parameter needed 0101b - Clear stall (see set stall): no parameter needed 0110b - Start transfer: 64-bit parameter

Field	Function
	0111b - Update transfer: no parameter needed 1000b - End transfer: no parameter needed 1001b - Start new configuration: no parameter needed

28.2.58 ADP configuration register (ADPCFG)

28.2.58.1 Offset

Register	Offset
ADPCFG	CC20h

28.2.58.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRBPER		PRBDELTA		PRBDSCHG		Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

28.2.58.3 Fields

Field	Function
31-30 PRBPER	Probe period This bit sets the value of T_ADP_PRB. The scaledown values for PRBPER are: 00 12.5 ms 01 18.75 ms 10 25 ms

Table continues on the next page...

Field	Function
	11 31.25 ms 00b - 0.775 sec 01b - 1.55 sec 10b - 2.275 sec 11b - Reserved
29-28 PRBDELTA	Probe delta These bits set the resolution for RTIM value. They are defined in units of 32 KHz clock cycles. For example, if this value is chosen to be 2'b01, it means that RTIM increments for every two 32 KHz clock cycles. 00b - 1 cycle 01b - 2 cycles 10b - 3 cycles 11b - 4 cycles
27-26 PRBDSCHG	Probe discharge These bits set the time for TADP_DSCHG. The scaledown values for the PrbDschg are as follows: 00 62.5 µs 01 125 µs 10 250 µs 11 500 µs 00b - 4 ms 01b - 8 ms 10b - 16 ms 11b - 32 ms
25-0 —	Reserved

28.2.59 ADP control register (ADPCTL)

28.2.59.1 Offset

Register	Offset
ADPCTL	CC24h

28.2.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			ENAPRB	ENASNS	ADPEN	ADPRES	WB	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

28.2.59.3 Fields

Field	Function
31-29 —	Reserved
28 ENAPRB	Enable probe When set to 1 along with ADPEN, the core performs a probe operation.
27 ENASNS	Enable sense When set to 1 along with ADPEN, the core performs a sense operation.
26 ADPEN	ADP enable When set to 1, the core performs either ADP probing or sensing based on ENAPRB and ENASNS. Setting ADPEN to 0 gates the suspend clock for major portion of ADP related logic.
25 ADPRES	ADP reset When set to 1, the ADP controller is reset. This bit is auto-cleared after the reset procedure is complete in the ADP controller.
24 WB	Write busy The application can read or write ADPCFG and ADPCTL registers only if this bit is cleared. Hardware sets this bit when the write is in progress in the suspend clock domain. 0b - Write completed 1b - Write in progress
23-0 —	Reserved

28.2.60 ADP event register (ADPEVT)

28.2.60.1 Offset

Register	Offset
ADPEVT	CC28h

28.2.60.2 Function

Writing 1 to the information bit in this register clears the register bit and associated interrupt.

28.2.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			ADPPRBEVNT	ADPSNSEVNT	ADPTMOUTEVNT	ADPRSTCMPLTEVNT	Reserved								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RTIM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.2.60.4 Fields

Field	Function
31-29 —	Reserved
28 ADPPRBEVNT	ADPEVTINFO[4] ADP probe event (ADPPRBEVNT)

Table continues on the next page...

Field	Function
	When this event is set, it means that the VBUS voltage is greater than VADPPRB or VADPPRB is reached.
27 ADPSNSEVNT	ADPEVTINFO[3] ADP sense event (ADPSNSEVNT) When this event is set, it means that the VBUS voltage is greater than VADPSNS or VADPSNS is reached.
26 ADPTMOUTEVNT	ADP timeout event This event is relevant when ADP probe command is executed. When this event is set, it means that the ramp time is completed (GADPCTL[RTIM] has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle.
25 ADPRSTCMPLTEVNT	ADP reset complete event This event when set, indicates that the ADP reset command is successful.
24-16 —	Reserved
15-0 RTIM	Ramp time These bits capture the latest time it took for VBUS to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 KHz clock cycles. The maximum time of 65536 cycles corresponds to a time of 2.04 seconds. Note for scaledown ramp_timeout PRBDELTA = 2'b00 => 6250 µs PRBDELTA = 2'b01 => 3125 µs PRBDELTA = 2'b10 => 1562.5 µs PRBDELTA = 2'b11 => 781.25 µs 0000000000000000b - 1 cycle 0000000000000001b - 2 cycles 0000000000000010b - 3 cycles.... 1111111111111111b - 65536 cycles

28.2.61 ADP event enable register (ADPEVTEN)

28.2.61.1 Offset

Register	Offset
ADPEVTEN	CC2Ch

28.2.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			ADPPRBEVNTEN	ADPSNSEVNTEN	ADPTMOUTEVNTEN	ADPRSTCMPLTEVNTEN	Reserved								
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

28.2.61.3 Fields

Field	Function
31-29 —	Reserved
28 ADPPRBEVNTEN	ADP probe event enable When this bit is set, the ADPEVT[ADPPRBEVNT] bit is enabled.
27 ADPSNSEVNTEN	ADP sense event enable When this bit is set, the ADPEVT[ADPSNSEVNT] bit is enabled.
26 ADPTMOUTEVNTEN	ADP timeout event enable When this bit is set, the ADPEVT[ADPTMOUTEVNT] bit is enabled.
25 ADPRSTCMPLTEVNTEN	ADP reset complete event enable When this bit is set, the ADPEVT[ADPRSTCMPLTEVNT] bit is enabled.
24-0 —	Reserved

28.3 USB 3.0 Controller and PHY Configuration

The USB controllers are configurable via CCSR, while the USB PHYs are configurable via DCSR.

USB PHY registers should not be accessed during the PBI phase before the system is ready.

For example- To access the USB PHY1 SUP.IDCODE_LO register, the following physical address must be used.

($DCSR_BASE + $USB1_BASE + register_offset) = 0x07_0000_0000 + 0x0460_0000 + 2* 0x1000 = 0x07_0460_2000

28.4 USB PHY SuperSpeed register descriptions

28.4.1 USB_PHY_SS memory map

USB PHY registers should not be accessed during the PBI phase before the system is ready.

USBPHY1 base address: 460_0000h

USBPHY2 base address: 461_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	SUP_IDCODE_LO (IP_IDCODE_LO)	16	RO	64CDh
2h	SUP_IDCODE_HI (SUP_IDCODE_HI)	16	RO	A21Ch
50h	MPLL_LOOP_CTL (MPLL_LOOP_CTL)	16	RW	0040h
1FFCh	LANE0_RX_OVRD_IN_HI (LANE0_RX_OVRD_IN_HI)	16	RW	0000h

28.4.2 SUP_IDCODE_LO (IP_IDCODE_LO)

28.4.2.1 Offset

Register	Offset
IP_IDCODE_LO	0h

28.4.2.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	1	1	0	0	1	0	0	1	1	0	0	1	1	0	1

28.4.2.3 Fields

Field	Function
15-0 DATA	DATA These bits indicate the IP version.

28.4.3 SUP_IDCODE_HI (SUP_IDCODE_HI)

28.4.3.1 Offset

Register	Offset
SUP_IDCODE_HI	2h

28.4.3.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	1	0	1	0	0	0	1	0	0	0	0	1	1	1	0	0

28.4.3.3 Fields

Field	Function
15-0 DATA	DATA These bits indicate the IP version.

28.4.4 MPLL_LOOP_CTL (MPLL_LOOP_CTL)

28.4.4.1 Offset

Register	Offset
MPLL_LOOP_CTL	50h

28.4.4.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PROP_CNTRL				Reserved			
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

28.4.4.3 Fields

Field	Function
15-8 —	-
7-4 PROP_CNTRL	PROP_CNTRL Charge pump proportional current setting
3-0 —	-

28.4.5 LANE0_RX_OVRD_IN_HI (LANE0_RX_OVRD_IN_HI)

28.4.5.1 Offset

Register	Offset
LANE0_RX_OVRD_IN_HI	1FFCh

28.4.5.2 Function

For initialization details, refer Initialization/application information.

28.4.5.3 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				RX_EQ_OVRD	RX_EQ			RX_EQ_EN_OVRD	RX_EQ_EN	Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28.4.5.4 Fields

Field	Function
15-12 —	-
11 RX_EQ_OVRD	RX_EQ_OVRD Override value for rx_eq
10-8 RX_EQ	RX_EQ Override value for rx_eq
7 RX_EQ_EN_OVRD	RX_EQ_EN_OVRD Override enable for rx_eq_en
6	RX_EQ_EN

Table continues on the next page...

Field	Function
RX_EQ_EN	Override value for rx_eq_en
5-0 —	Reserved

28.5 Functional Description

28.5.1 System memory descriptor and data buffers

The software creates transfer request blocks (TRBs), and four DWORDs each, that point to the data buffers. Normally, the TRBs are allocated consecutively in system memory; only the data buffers can be scattered. In the case of a circular buffer, the link-TRB points to the next TRB. Once the TRBs and data buffers are set up in the system memory, the software driver issues a start transfer command that points to the location of the first TRB in the system memory to start the DMA operation. TRBs, though small (only four DWORDs), provide a rich set of features for the software to schedule transfers, isochronous, control, interrupt moderation, and so on.

Figure 28-4. System Memory Descriptor and Data Buffers

28.5.2 Device descriptor structures

Device mode transfer request blocks (TRBs) are small 4 DWORDs and at the same time provide a rich set of features so that the software can efficiently manage the USB core, memory buffers, and MIPS requirements.

The following is a list of device mode TRB characteristics:

- TRBs provide support for scattered data structures. The scattered data buffers can be zero bytes to 16 MB in length.
- TRBs must be placed in system memory aligned to a 16-byte boundary.
- TRBs are kept in linear memory to enhance descriptor caching performance. If the data buffers are small (as in an Ethernet application) the core can efficiently collect the scattered data to build USB packets without wasting bus efficiency that collecting scattered descriptors requires.

- Supports TRB linking. This allows software to set up a circular ring of TRBs with the last TRB linking to the first one.
- Supports system memory interface with 40-bit addressing capability
- Supports byte-aligned buffers for each TRB of a transfer. This feature prevents the need for buffer copying in cases where the USB device driver receives unaligned data buffers from other applications (for example, Ethernet). Whenever the application controls buffer allocation, it must allocate SoC bus width and burst-aligned buffers to facilitate efficient bus and memory utilization. For transmitting, the core supports byte-aligned buffers on all TRBs.

  For example: To use 16 bursts in a 64-bit system, you must try to allocate buffers that are 16 * 8 = 128 bytes aligned. This is the normal buffer structure because Linux-like OSs allocate 4 KB buffers. SDR/DDR memory controllers also provide better performance when requests are burst aligned.

- Supports software queueing of multiple USB transfers (LST bit for IN/OUT transfers and CSP bit for OUT transfers control this function).
- Provides interrupt moderation capability, allowing software to selectively enable events on TRB completion, as controlled by the IOC (Interrupt on completion) bit. The IOC event is also used by software to reallocate the released buffers, allowing software to re-use just a few buffers in a circular fashion. This helps when the memory is limited but have enough MIPS to process interrupts. Larger buffers can be allocated to reduce the number of interrupts.
  - In USB 3.0 larger transfers are recommended because the raw transfer rate is almost 10 times faster, unlike USB 2.0, where drivers set up only 64 KB or 128 KB transfers. For example, a 64 KB transfer that takes 1.3 ms in USB 2.0 requires only 164 s in USB 3.0. If you set up buffers of 64 KB and enable the transfer completion event, then you receive an interrupt every 164 sec.
- Supports streaming (Stream ID field used for this purpose).

28.5.2.1 Structures

This figure shows the control and status field of a transfer request block.

Figure 28-5. TRB control and status fields

Table 28-8. Device descriptor structure field definitions

Field	Description	Hardware access
DW 03-00		
31:0	Buffer pointer low (BPTRL) Data buffer pointer to low 32 bit address (BPTR[31:0]). Hardware may also update this field (implementation specific).	R/W
DW 07-04		
31:0	Buffer pointer high (BPTRH) Data buffer pointer to high 32-bit address (BPTR[63:32]).	R/W
DW 0B-08		
31:28	TRB status (TRBSTS) Hardware updates this field with transfer status information before releasing the TRB. • 4'h0: OK • 4'h1: MissedIsoc: Isochronous interval missed or incomplete • 4'h2: SetupPending - During the current control transfer data/status phase, another SETUP was received. • 4'h4: TransferInProgress - During the current transfer, an end transfer command was received.	R/W
27	Reserved. Hardware may update these bits for internal usage.	R/W
26	SPR • If GUCTL1[DEV_TRB_OUT_SPR_IND] is 1'b1, this field indicates the short packet received for the last processed TRB for the transfer. If a short packet has been received, this TRB field is set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. • If GUCTL1[DEV_TRB_OUT_SPR_IND] is 1'b0, this field is reserved. Hardware may update these bits for internal usage.	R/W
25:24	Packet count M1 (PCM1) For high-speed, High bandwidth isochronous IN endpoints, this field in an Isoc-First TRB represents the total number of packets in the Buffer Descriptor minus 1.	R/W
23:0	Buffer size (BUFSIZ) If CHN=0 and HWO=0 for the TRB, this field represents the total remaining Buffer Descriptor buffer size in bytes. Valid Range: 0 bytes to (16 MB - 1 byte). The hardware decrements this field to represent the remaining buffer size after data is transferred.	R/W

Table continues on the next page...

Table 28-8. Device descriptor structure field definitions (continued)

Field	Description	Hardware access
	For a Link TRB, the buffer size should be "0".	
DW 0F-0C		
31:30	Reserved.	R/W
29:14	Stream ID / SOF Number For stream-based bulk endpoints: The Stream ID of the transfer this TRB is associated with. Stream ID must be the same in all TRBs (R/W). For isochronous endpoints: The (micro)frame number in which the last packet of this TRB's buffer was transmitted or received (debug purposes only) (RO).	R/W
13:12	Reserved.	R/W
11	Interrupt on Complete (IOC) When IOC is set in a TRB, and once the transfer for this buffer is completed, the core will issue XferInProgress event with IOC bit set in the event's status. This indicates the buffer is available for software to reuse or release.	R
10	Interrupt on Short Packet / Interrupt on Missed ISOC (ISP/IMI) Applicable to OUT endpoints when a short packet is received, and CSP=1 and LST=0. If this bit is 1, the core generates an XferInProgress event. For Isochronous endpoints: If this bit is 1, the core generates an XferInProgress event when the interval represented by the Buffer Descriptor completes with a "Missed Isoc" status.	R
9:4	TRB Control (TRBCTL) Indicates the type of TRB: 1: Normal (Control-Data-2+ / Bulk / Interrupt) - Set TRBCTL to 1 for all TRBs used in data stage except the first TRB 2: Control-Setup 3: Control-Status-2 - Set TRBCTL to 3 for a SETUP request without data stage 4: Control-Status-3 - Set TRBCTL to 4 for a SETUP request with data stage 5: Control-Data - Set TRBCTL to 5 for the first TRB of a data stage 6: Isochronous-First - Set TRBCTL to 6 for the first TRB of a Service Interval 7: Isochronous 8: Link TRB Others: Reserved	R
3	Continue on Short Packet (CSP) Applicable to OUT endpoints only when a short packet is received. If this bit is 1, the core will continue to the next Buffer Descriptor. This setting is required for isochronous endpoints. If this bit is 0, the core will generate an XferComplete event and remove the stream.	R
2	Chain Buffers (CHN) Applicable to IN and OUT endpoints. Set to 1 by software to associate this TRB with the next TRB. A Buffer Descriptor is defined as one or more TRBs. The CHN bit is used to identify the TRBs that comprise a Buffer Descriptor. The CHN bit is always 0 in the last TRB of a Buffer Descriptor and when the LST field is set to 1.	R

Table continues on the next page...

Table 28-8. Device descriptor structure field definitions (continued)

Field	Description	Hardware access
1	Last TRB (LST) Indicates this is the last TRB in a list. After completing the transfer for the associated buffer the core will stop the transfer for the endpoint / bulk-stream and issues an XferComplete event. The stream is automatically removed by the hardware.	R
0	Hardware Owner of Descriptor (HWO) Indicates that hardware owns the TRB. Software sets this bit to 1 when it creates the TRB, and cannot modify it until hardware resets this bit to 0. However, there are exceptions for short packets on OUT endpoints and Link TRBs. Because the hardware autonomously checks this bit to determine if the entire TRB is valid, software must set this field to '1' after preparing the other three DWORDs of the TRB with valid information.	R/W

28.5.2.1.1 Normal (Control-Data/Bulk/Interrupt), Isochronous, and Status Transfer Request Block Structure

The Normal TRB is used for Bulk/Control-Data/Interrupt endpoint transfers. The data buffers can be scattered anywhere and each may have different sizes. The TRB Buffer Pointer and the Buffer Size fields point to buffer address and size respectively. The Stream ID for bulk endpoints will be programmed by the application.

For isochronous endpoints, the first TRB in a service interval must have the Isoc-First type, the last TRB in a service interval must have CHN=0, and any other TRBs have CHN=1. The starting (micro)frame time is communicated via the Start Transfer command, and the core tracks the (micro)frame times of the subsequent Buffer Descriptors.

Figure 28-6. Normal (Control-Data/Bulk/Interrupt) Descriptor Structure

28.5.2.1.2 Setup and Status TRB Structure

To receive a SETUP packet, the driver queues up a single Setup TRB, whose buffer pointer value may be set to any address, including the address of the TRB. The buffer size must be set to 8. The core will write the 8 bytes of the received SETUP to the address requested. If the address of the TRB is used, there is no need for a separate data buffer to receive a SETUP packet. After completing Setup stage, driver will schedule Data stage and Status stage transfers. For more information, see Control transfer programming model

Figure 28-7. SETUP Descriptor Structure with Buffer Pointing to Setup TRB

After interpreting the SETUP bytes, software will determine if the next stage of the control transfer is a data stage or status stage.

If the SETUP bytes require a 3-stage control transfer, the TRB type used in the Data stage must be Control- Data for the first TRB of the Buffer Descriptor. When the host moves on to the Status stage, the TRB Type must be Control-Status-3.

If the SETUP bytes require a 2-stage control transfer, the TRB Type must be Control-Status-2.

The Status TRB is a zero-length TRB, with an unspecified Buffer Pointer value and a Buffer Size of zero. There is no data buffer associated with a Status TRB.

28.5.2.1.3 Link TRB Structure

The Link TRB is used to link back to the starting TRB for reusing TRBs in a circular fashion.

Figure 28-8. Link TRB Structure

If software prepares a circular TRB list which is shorter than the number of cached TRBs (four), the core automatically detects the loop and will not re-fetch a TRB that has already been fetched. For example, for four cached TRBs the software setup three TRBs in the following way:

- TRB-1: Normal, HWO=1
- TRB-2: Normal, HWO=1
- TRB-3: Link to TRB 1

The core will fetch TRBs 1, 2, and 3. Then the core will follow the link to TRB-1, note that its address is the same as the TRB-1 that has already been fetched into the cache, and will temporarily stop fetching TRBs. When TRB-1 completes due to traffic on the USB, the core will write TRB-1 back to memory with the HWO field set to '0', generate a XferInProgress event if necessary, and will automatically attempt to fetch TRB-1 again. In most cases, the core will see the HWO field set to '0' and will stop fetching until software updates the TRB, sets the HWO field back to '1', and issues an Update Transfer command.

28.5.2.1.4 Chaining buffers (CHN) and interrupt on completion (IOC) usage

This figure shows a chaining buffer example for an isochronous IN and a bulk IN transaction.

Figure 28-9. Chaining buffers for isochronous and bulk IN

In the isochronous IN application shown in the above figure, there are three video frames to be sent to the host in each microframe. Each video buffer size is maximum of 4 KB and the size may vary for each microframe depending on the compression. The CHN bit

in the TRB-3 is 0, which indicates this is the last buffer of a transfer for the given microframe. The device schedules TRB-1, 2, and 3 in the first bInterval. Similarly, TRB-6 indicates (CHN=0) microframe boundary, and TRB-4, 5, and 6 will be sent during the next bInterval.

For the last packet of last transfer in a microframe, the device internally sets last packet flag when responding at SuperSpeed for isochronous IN transfers.

The application can also use the IOC bit to receive an interrupt when the buffer-A transfer is completed, so it can use buffer-A to send data on the bInterval following the next bInterval. As long as the driver can service the interrupt and set up data before another bInterval, the USB transfers can continue without interruption. Depending upon the system interrupt latency, the buffer size can be adjusted.

If interrupt latency is high occasionally, the software may not have time to set up the linked TRB before the next bInterval. In this case, the core, on seeing HWO bit set to 0 in the TRB, stops processing further TRBs of this endpoint. When hardware receives the Update Transfer command from software, it will re-fetch the TRB. In the case of bulk application, the core will issue the NRDY signal. In isochronous transfers, zero- length packets are sent to the host until software enables the transfer again.

In the bulk IN example, the CHN bit indicates USB Transfer boundary. During a TRB transfer, CHN indicates whether to send the bytes from next TRB buffer as part of the current transfer. For example, even though TRB-1 has 2,345 bytes, the last 297 (2345 - 2 * 1024 = 297) bytes will be combined with the data in TRB-2 and sent as a 1,024 byte packet on the USB since CHN=1. On the other hand, when there is a short packet left in TRB-3, the short packet will be sent separately, since CHN=0.

The device writes data into TRB-1 and TRB-2 for the first transfer. The CHN bit (CHN=0) in TRB-2 indicates that this is the last buffer of the transfer, then TRB-3 and TRB-4 are written for the second transfer. Similarly, the CHN bit (CHN=0) in TRB-4 indicates the second transfer boundary.

If an interrupt latency is high, software may not have time to set up the TRB. In this case, the core, on seeing the HWO bit set to 0 in the TRB, will stop processing further TRBs for the endpoint. Hardware will re-fetch the TRB when software issues the update transfer command. In the case of bulk, control, or interrupt endpoints, the core will issue NRDY. In the case of isochronous endpoints, packets will be dropped until software enables the transfer again.

In the bulk OUT example below, when a short packet is received for TRB-1, TRB-3 will be used for the next OUT transfer and TRB-2 will be closed.

Figure 28-10. Chaining buffers, example 1

Figure 28-11. Chaining buffers, example 2

Figure 28-12. Bulk OUT with two transfers

28.5.2.1.5 Interrupt on Short Packet (ISP) and Continue on Short Packet (CSP) Usage

These two bits are used to schedule single or multiple OUT transfer. In most applications where only one OUT transfer per endpoint is scheduled, the software always sets ISP=1 and CSP=0. If the device receives a short packet, then it indicates a USB transfer completion through XferComplete events.

In an Ethernet-over-USB application, where software knows it is going to receive short packets, it can set up multiple transfer size buffers in one step by setting CSP=1 in all the TRBs. On a short packet, the device will update the byte count and move to next TRB. Software can also set ISP once in n number of TRBs to receive an XferInProgress event so it can process the previous short packets.

The following example shows software setting multiple 1500-byte Ethernet transfers and enabling the XferInProgress event once for four Ethernet packets to reduce the number of interrupts.

Figure 28-13. ISP and CSP Usage

28.5.2.1.6 Example of Setting Up TRBs

Figure 28-14. Bulk IN TRB Examples

Figure 28-15. Setup/Control/Status TRB Examples

Figure 28-16. Isochronous IN TRB, Example 2

Figure 28-17. Isochronous IN TRB, Example 1

28.5.3 Device Programming Model

28.5.3.1 Register Initialization

The USB3 core contains global registers (prefixed by G*) and device registers (prefixed by D*) that are programmed to start operation and handle certain events. This section describes which registers must be accessed depending on the event that software is attempting to handle:

- Power-On or Soft Reset
- USB Reset Event

- Connect Done Event
- SetAddress Device Request
- SetConfiguration Device Request
- Disconnect Event
- Device-Initiated Disconnect and Reconnect

28.5.3.1.1 Device Power-On or Soft Reset

This section explains device core initialization after power-on or soft reset. The application must follow this initialization sequence for device mode operation.

When the core is first powered on, software initializes the following registers. The order of operations is not important, except for the first and last steps (DCTL[CSFTRST]=1 and DCTL[RUN_STOP]=1).

Table 28-9. Power-On or Soft Reset Register Initialization

Register	Description
DCTL	Set the CSftRst field to '1' and wait for a read to return '0'. This resets the device core.
GSBUSCFG0/1	Leave the default values, refer Global SoC bus configuration register 0 (GSBUSCFG0) and Global SoC bus configuration register 1 (GSBUSCFG1)
GTXTHRCFG/ GRXTHRCFG	This is required only if threshold is enabled.
GUSB2PHYCFG	Program PHY as per Global USB2 PHY configuration register (GUSB2PHYCFG) or leave the default values if the correct power-on values were selected. Note: The PHY must not be enabled for auto-resume in device mode.
GUSB3PIPECTL	Program the following PHY [DatWidth] or leave the default values if the correct power-on values were selected.
GTXFIFOSIZn	Write these registers to allocate prefetch buffers for each Tx endpoint. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GRXFIFOSIZ0	Write this register to allocate the receive buffer for all endpoints. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GEVNTADR/ GEVNTSIZ/ GEVNTCOUNT	Depending on the number of interrupts allocated, program the Event Buffer Address and Size registers to point to the Event Buffer locations in system memory, the sizes of the buffers, and unmask the interrupt. Note: USB operation stops if the Event Buffer memory is insufficient, because the core stops receiving/ transmitting packets.
GCTL	Program this register to override scaledown, RAM clock select, and clock gating parameters.
DCFG	Program device speed and periodic frame interval
DEVTEN	At a minimum, enable USB Reset, Connection Done, and USB/Link State Change events
DEPCMD0	Issue a DEPSTARTCFG command with DEPEVT.XferRscIdx set to 0 and CmdIOC set to 0 to initialize the transfer resource allocation. Poll CmdAct for completion.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command for physical endpoints 0 & 1 with the following characteristics, and poll CmdAct for completions: USB Endpoint Number = 0 or 1 (for physical endpoint 0 or 1) FIFONum= 0

Table continues on the next page...

Table 28-9. Power-On or Soft Reset Register Initialization (continued)

Register	Description
	XferNRdyEn and XferCmplEn = 1 Maximum Packet Size = 512 Burst Size = 0 EPType = 2'b00 (Control)
DEPCMD0/ DEPCMD1	Issue a DEPXFERCFG command for physical endpoints 0 & 1 with DEPCMDPAR0_0/1 set to 1, and poll CmdAct for completions
DEPCMD0	Prepare a buffer for a setup packet, initialize a setup TRB, and issue a DEPSTRTXFER command for physical endpoint 0, pointing to the setup TRB. Poll CmdAct for completion. Note: The core attempts to fetch the setup TRB via the master interface after this command completes.
DALEPENA	Enable physical endpoints 0 & 1 by writing 0x3 to this register
DCTL	Set DCTL[RunStop] to '1' to allow the device to attach to the host. At this point, the device is ready to receive SOF packets, respond to control transfers on control endpoint 0, and generate events.

After the controller has been started, wait for the following events:

- Wait for a DEVT.USBReset event. This indicates that a reset is detected on the USB.
- Wait for a DEVT.ConnectionDone event. This event indicates the end of the reset on the USB. On this event, read the DSTS register to get the connection speed.

28.5.3.1.2 Initialization on USB reset

To initialize the core as a device, during USB Reset, the application must perform the following steps:

Table 28-10. Initialization on USB reset

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0)
DEPCMDn	Issue a DEPCSTALL (ClearStall) command for any endpoint in STALL mode prior to the USB Reset (excluding control endpoints)
DCFG	Set DevAddr to '0'

NOTE

- Special Reset Considerations for Default Control Endpoint 0: The default control endpoint is not affected by a USB Reset, so software must continue following the software flow for control transfers explained in "Control Transfer Programming Model" even across a USB Reset event.

Resources can only be assigned to the default control endpoint once after a power on or soft reset.

28.5.3.1.3 Initialization on connect done

When this event is received, software must perform the following steps:

Table 28-11. Initialization on connect done

Register	Description
DSTS	Read this register to obtain the connection speed
GCTL	Program the RAMClkSel field to select the correct clock for the RAM clock domain. This field is reset to 0 after USB reset, so it must be reprogrammed each time on Connect Done.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command (with Config Action set to "Modify") for physical endpoints 0 & 1 using the same endpoint characteristics from Power-On Reset, but set MaxPacketSize to 512 (SuperSpeed), 64 (High-Speed), 8/16/32/64 (Full-Speed), or 8 (Low-Speed).
GUSB2CFG/ GUSB3PIPECTL	Depending on the connected speed, write to the other PHY's control register to suspend it
GTXFIFOSIZn	(optional) Based on the new MaxPacketSize of IN endpoint 0, software may choose to re-allocate the TXFIFO sizes by writing to these registers.

28.5.3.1.4 Initialization on SetAddress request

When the application receives a SetAddress request in a SETUP packet, it performs the following steps:

Table 28-12. Initialization on SetAddress request

Register	Description
DCFG	Program the DCFG register with the device address received as part of the SetAddress request when SETUP packet is decoded.
DEPCMD1	After receiving the XferNotReady(Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the DCFG register is programmed with the new device address.

At this point, the device is ready to receive micro-SOF/ITP and is configured to receive control transfers on control endpoint 0 with a new address assigned.

28.5.3.1.5 Initialization on SetConfiguration or SetInterface Request

When the application receives a SetConfiguration or SetInterface request in a SETUP packet, it performs the following steps:

Table 28-13. Initialization on SetConfiguration or SetInterface Request

Register	Description
DALEPENA	Set this register to 0x3 to disable all endpoints other than the default control endpoint 0.
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0).
DEPCMD1	Issue a DEPCFG command (with Config Action field set to "Modify") for physical endpoint 1 using the current endpoint characteristics to re-initialize the TXFIFO allocation.
DEPCMD0	Issue a DEPSTARTCFG command with DEPCMD0.XferRscIdx set to 2 to re-initialize the transfer resource allocation.
DEPCMDn	Issue a DEPCFG command (with the Config Action field set to "Initialize") for each endpoint that is present in the new configuration (except for the default control endpoint). Note: Control endpoints are bi-directional and must use consecutive even/odd physical endpoint numbers for the OUT/IN direction (such as 2/3, or 4/5), and the FIFONum must be configured to the same value in both directions.
DEPCMDn	Issue a DEPXFERCFG command for each endpoint that is present in the new configuration (except for the default control endpoint 0).
GTXFIFOSIZn	(optional) Based on the new configuration of IN endpoints, software may choose to re-allocate the TXFIFO sizes by writing to these registers.
DALEPENA	Enable the logical endpoints that are active in the new configuration.
DEPCMD1	After receiving the XferNotReady(Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the previous steps to ensure the host does not access any other endpoints that are being set up.

At this point, the core is prepared to accept Start Transfer commands for the newly-configured endpoints. For information on how to set up transfers, see the Operational Model.

28.5.3.1.6 Alternate initialization on SetInterface request

When handling a SetInterface device request, another possibility is that software may reconfigure existing endpoints instead of starting the configuration from the beginning. This is only possible if no TXFIFOs need to be reassigned. This flow also assumes that the endpoints that are being removed or reconfigured in the new interface have halted their traffic prior to the host issuing the SetInterface request.

Table 28-14. Alternate initialization on SetInterface request

Register	Description
DEPCMDn	Make sure there are no transfers still active on the endpoints that are changing in the new interface. This is normally ensured by the host prior to issuing the SetInterface, but if not, issue an End Transfer command for the transfer on each endpoint that is changing.
DEPCMDn	Issue a DEPCFG command (with the Config Action field set to "Initialize") for each endpoint that is changing in the new configuration. The side effect of the DEPCFG command is that the endpoint's sequence number is automatically set to 0.
DALEPENA	Write this register with all the endpoints that are enabled (including the ones that are not changing).
DEPCMDn	Using the StartXfer command, acknowledge the status stage response for the SetInterface request.

28.5.3.1.7 Initialization on Disconnect Event

When the application receives a Disconnect event, it must set DCTL[ULSTCHNGREQ] to 5. Other than this, the core does not require any initialization. Because the DCTL[RUN_STOP] bit is still '1', the device attempts to reconnect to the host, at which time a USB Reset and Connect Done event occurs. However, if the application does not want to attempt to reconnect to the host, it should perform the steps in the next section.

NOTE

When DCFG[DEVSPD] is programmed for 2.0 only mode (such as, High-Speed or Full-Speed), and if the application wants to issue any commands to clear any pending transfers during a Disconnect interrupt, then it has to disable USB_GUSB2PHYCFG[SUSPENDUSB20] before issuing any commands and re-enable it after the commands have completed.

28.5.3.1.8 Device-initiated disconnect

If the application wants to disconnect from the host, it should perform the following actions:

Table 28-15. Initialization on device-initiated disconnect

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0)
DCTL	Set DCTL[RUN_STOP] to '0' to disconnect from the host
DSTS	Poll [DEVCTRLHLT] until it is '1'

At this point, the device is disconnected from the host and will not attempt to reconnect.

28.5.3.1.9 Reconnect after Device-Initiated Disconnect

If the application decides to reconnect to the host, it must follow the steps in Device Power-On or Soft Reset.

28.5.3.2 Operational Model

The following sections describes the processes and data structures that the core uses to implement the USB 3.0 specification.

28.5.3.2.1 USB and Physical Endpoints

Endpoints are referred to in two ways:

- As a USB endpoint number: USB endpoints are defined in the USB specification.
- As a physical endpoint resource number: The hardware has a fixed number of physical endpoint resources. Each resource is unidirectional and can be configured to refer to either direction of any USB endpoint.

The software always works on physical endpoints and it knows how physical endpoint corresponds to USB endpoints. DALEPENA is the only register that has one enable bit per USB endpoint.

During SetConfiguration, software maps physical endpoint resources to the required USB endpoints. When a USB request comes in, the USB endpoint number gets converted to a physical endpoint number in the core. Similarly, when a USB packet is sent out, the physical endpoint number is converted to the USB endpoint and sent out.

A USB control endpoint requires two physical endpoints. One physical endpoint is mapped to the OUT direction of the Control endpoint, and the other one is mapped to the IN direction of the Control endpoint. Specifically the two physical endpoints are used as the following:

- The Setup stage of any control transfer uses the OUT direction physical endpoint.
- For a control write or 2-stage transfer, the Data stage (if present) uses the OUT direction physical endpoint and the Status stage uses the IN direction physical endpoint.
- For a control read transfer, the Data stage uses the IN direction physical endpoint and the Status stage uses the OUT direction physical endpoint.

28.5.3.2.2 Event Buffers

Hardware passes command completion, transfer progress, and asynchronous events to software through one or more Event Buffers.

To configure an Event Buffer, the software performs the following steps:

1. Sets up an empty buffer in system memory.
2. Writes the address of the beginning of the buffer into GEVNTADR. This address must be aligned to the Event Buffer size.
3. Writes the size of the buffer and interrupt mask into GEVNTSIZ. Depending on your system interrupt latency, enough Event Buffer space must be allocated to avoid lost interrupts or reduced performance.

4. Write a 0 into the GEVNTCOUNT register. This must be the last step, as it enables the Event Buffer. After the Event Buffer has been configured, software must not change the size or address.

There is one interrupt line per Event Buffer that indicates there are one or more events present. Software reads one or more events out of the buffer and indicates to hardware how many events it processed by writing the byte count to the GEVNTCOUNT register.

Clock crossing delays may result in the interrupt's continual assertion after software acknowledges the last event. Therefore, when the interrupt line is asserted, software must read the GEVNTCOUNT register and only process events if the GEVNTCOUNT is greater than 0.

The first event produced by the core after the Event Buffer is configured will be written to the address specified in GEVNTADR. Most events are 32 bits, and subsequent events will be written to the address (PreviousEventAddress + 4). When that address exceeds the sum of GEVNTADR and GEVNTSIZ, the core wraps around to the first GEVNTADR value. In this way, the Event Buffer operates like a circular buffer with hardware writing to the "tail" of the buffer and software reading from the "head."

Most events are exactly 4 bytes in size, but there is one exception: The Vendor Device Test LMP Received Event (VndrDevTstRcved) is a 12 byte event that includes a header in the first 4 bytes and the contents of the LMP in the following 8 bytes.

When an event occurs within the core, hardware checks the enable bit that corresponds to the event to decide whether the event will be written to the Event Buffer or not. The Event Buffer contains one of the following types of information, depending on the value of the lower bits of the event:

- Device Endpoint-Specific Event (DEPEVT) (Event[0] = 0x0)
  - The DEPCFG endpoint-specific command specifies the bits that are enabled and which Event Buffer to use for these events.
- Device-Specific Event (DEVT) (Event[0] = 0x1, Event[7:1] = 0x00)
  - The Generic Command Complete event is enabled through the DGCMD[CmdIOC] field when the command is issued.
  - The Event Buffer Overflow Event cannot be disabled and is written to the Event Buffer that encounters the overflow.
  - The rest of the Device-Specific events are enabled through the DEVTEN register.
  - Except for the Event Buffer Overflow Event, these events are written to the Event Buffer specified in the DCFG[IntrNum] field.

The core always leaves one entry free in each Event Buffer. When the Event Buffer is almost full, hardware writes the Event Buffer Overflow event and the USB will eventually get stalled when endpoints start responding NRDY or the link layer will stop returning credits (in SuperSpeed). This event is an indication to software that it is not processing events quickly enough. During this time, events will be queued up internally. When software frees up Event Buffer space, the queued up events will be written out and the USB will return to normal operation.

Event Buffer Content for Device Endpoint-Specific Events (DEPEVT)

Table 28-16. Device Endpoint-n Events: DEPEVT

Field	Description
31–16	Event Parameters (EventParam) For XferNotReady, XferComplete, and Stream events on Bulk Endpoints: [31-16]StreamID. Applies only to bulk endpoints that support streams. This indicates the Stream ID of the transfer for which the event is generated For XferInProgress: [31-16] Isochronous Microframe Number (IsocMicroFrameNum). Indicates the microframe number of the beginning of the interval that generated the XferInProgress event (debug purposes only) For XferNotReady events on Isochronous Endpoints: [31-16] Isochronous Microframe Number (IsocMicroFrameNum). Indicates the microframe number during which the endpoint was not ready **NOTE:** USB 3.0 core represents USB bus time as a 14-bit value on the bus and also in the DSTS register (DSTS[SOFFN]), but as a 16-bit value in the XferNotReady event. Use the 16-bit value to interact with Isochronous endpoints via the StartXfer command. The extra two bits that the USB 3.0 core produces will be necessary for handling wrap-around conditions in the interaction between software and hardware. EPCmdCmplt events For all EPCmdCmplt events [27-24]- Command Type. The command type that completed (Valid only in a DEPEVT event. Undefined when read from the DEPCMD.EventParam field). For EPCmdCmplt event in response to Start Transfer command. [22-16] Transfer Resource Index (XferRscIdx). The internal hardware transfer resource index assigned to this transfer. This index must be used in all Update Transfer and End Transfer commands.
15–12	Event Status (EventStatus) Within an XferNotReady event- [15]: Indicates the reason why the XferNotReady event is generated: 0- XferNotActive- Host initiated a transfer, but the requested transfer is not present in the hardware 1- XferActive- Host initiated a transfer, the transfer is present, but no valid TRBs are available [14]: Not Used [13:12]: For control endpoints, indicates what stage was requested when the transfer was not ready: 2'b01- Control Data Request 2'b10- Control Status Request

Table continues on the next page...

Table 28-16. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
	Within an XferComplete or XferInProgress event- [15]: LST bit of the completed TRB (XferComplete only) [15]: MissedIsoc: Indicates the interval did not complete successfully (XferInProgress only) [14]: IOC bit of the TRB that completed [13]: Indicates the TRB completed with a short packet reception or the last packet of an isochronous interval [12]: Reserved If the host aborts the data stage of a control transfer, software may receive a XferComplete event with the EventStatus field equal to '0'. This is a valid event that must be processed as a part of the Control transfer programming model. Within a Stream Event- [15:12]: - 4'h2- StreamNotFound- This stream event is issued when the stream-capable endpoint performed a search in its transfer resource cache, but could not find an active and ready stream. - 4'h1- StreamFound- This stream event is issued when the stream-capable endpoint found an active and ready stream in its transfer resource cache, and initiated traffic for that stream to the host. The ID of the selected Stream is in the EventParam field. In response to a Start Transfer command- 4'h2- Indicates expiry of the bus time reflected in the Start Transfer command. 4'h1- Indicates there is no transfer resource available on the endpoint. In response to a Set Transfer Resource (DEPXFERCFG) command- 4'h1- Indicates an error has occurred because software is requesting more transfer resources to be assigned than have been configured in the hardware. In response to a End Transfer command- 4'h1- Indicates an invalid transfer resource was specified.
11–10	This field is reserved.
9–6	4'h7- Endpoint Command Complete (EPCmdCmplt) Indicates software may issue another Device Endpoint command to the endpoint. When issued in response to an End Transfer command, indicates that DMA stopped for the endpoint. For all other commands, this event does not imply that all effects of the commandtook place. The DEPCMD register contains the same status information present in the Event Status Bits field. 4'h6- Stream Event (StreamEvt) Indicates that a stream-capable endpoint initiated a search within its transfer resource cache. The result of the search is in the EventStatus field (Found or NotFound). 4'h5- Reserved 4'h4 - Reserved 4'h3- XferNotReady Event (XferNotReady) Indicates receipt of a transaction when no TRBs are available for the endpoint. For isochronous IN endpoints, a zero-length packet is automatically sent by hardware and NRDY for non-isochronous endpoints. The application must enable this event if it plans to issue Start Transfer on demand.

Table continues on the next page...

Table 28-16. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
	This event can happen when software issues a Start Transfer or Update Transfer. In this case, software must ignore this event because it has already issued the Start Transfer or Update Transfer. XferNotReady is generated when the core responds NRDY to the host on the USB. It is useful in the beginning of a transfer when software wants to wait for the host to start polling the endpoint before setting up TRBs, but it is not efficient to use this event to determine when the core has run out of TRBs. For determining when the core has processed TRBs and needs software to setup more, use the IOC field in a TRB along with the XferInProgress event that is generated when the core completes a TRB with IOC=1. For non-stream-capable endpoints, the hardware filters multiple events if the host continues transactions, even after the core responds with NRDY. This internal filter is reset after software issues any endpoint command to this endpoint. For stream-capable endpoints, this event is generated each time the host attempts a transaction, even if the core responds with NRDY. For isochronous endpoints, this event is generated only once prior to the Start Transfer command to communicate the current bus time. For additional information, see the Event Status field. 4'h2- XferInProgress Event (XferInProgress) Applies to IN and OUT endpoints. Indicates an EP/Stream specific event happened and it is continuing the transfer. For additional information, see the Device Programming Model. 4'h1- XferComplete Event (XferComplete) Applies to IN and OUT endpoints. Indicates that a EP/Stream transfer completed and the core stopped the transfer. For additional information, see the Device Programming Model. 4'h0- Reserved
5–1	Physical Endpoint Number (0-31)
0	1'b0- Indicates that this is an endpoint-specific event.

Event Buffer Content for Device-Specific Events (DEVT)

Table 28-17. Device-specific events: DEVT

Field	Description
31-25	Reserved
24-16	Event Information Bits (EvtInfo) For a USB/Link State change Event, this field indicates the state of the link- EvtInfo[4] - SuperSpeed event. Set to 1 for SS; Set to 0 for non-SS. EvtInfo[3:0] - Link State. Indicates link state at the time of the event. Follows same encoding as DSTS link state bits.
15-12	Reserved
11-8	11- Event Buffer Overflow Event (EvntOverflow) The core writes this event to indicate there is no space in the event buffer, and one or more Device-specific events may have been dropped after this event. Endpoint-Specific events will not be dropped, but they will be delayed which can cause a drop in performance on the USB.

Table continues on the next page...

Table 28-17. Device-specific events: DEVT (continued)

Field	Description
	Software uses this event as a warning that the Event Buffer is too small and the core should be reconfigured with a larger Event Buffer or software needs to process events more quickly. In order to avoid repeated Event Buffer Overflow Events, software must free up space in the Event Buffer by acknowledging more than 1 event (writing a value greater than 4 to the GEVNTCOUNT register). This event cannot be disabled, and is always written to the Event Buffer that encounters the overflow. 10- Generic Command Complete Event (CmdCmplt) The core writes this event to indicate the generic command is complete. 9- Erratic Error Event (ErrticErr) The core writes this event to report erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the USB 3.0 core goes into Suspended state and a USB/Link state change event (ULStChng) is generated to the application. If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover. For SuperSpeed operation, the PHY erratic error event indicates that the PIPE is not responding to the PHY command (for example, pipe3_PowerDown change or receiver detection). After the core requested any PHY command, if it did not receive pipe3_PhyStatus within 100ms, the core will timeout and generate the erratic error event. After generating the erratic error, the core assumes that the PHY command is completed successfully and proceeds with the next pending PHY command. Software must reset the controller on receiving the erratic error. 8- Reserved 7- Start of (micro)Frame (Sof) 6- USB Suspend Entry Event This event provides a notification that the link has gone to a suspend state (L2, U3, or L1). The existing Link State Change event (3) provides the same information, but is generated for every link state change. 5- Reserved 4- Resume/Remote Wakeup Detected Event (WkUpEvt) This event is generated when the host initiates a resume condition on the USB bus. This event is not generated by the device initiating a remote wakeup to the host. 3- USB/Link State Change (ULStChng) The core writes this event to indicate a change of USB or Link state. Bits 23-16 (EvtInfo) of this event indicate the Link Status. The event is generated in SuperSpeed when the link LTSSM state changes, except when LTSSM exits from HOT_RESET, POLL or LTSSM enters into RECOVERY. 2- Connection Done (ConnectDone) 1- USB Reset (USBRst) 0- Disconnect Detected Event (DisconnEvt) Note: This event is generated only if VBUS is off.
7-1	7’h00 - Device Specific Event
0	1’b1 - Non-Endpoint-Specific Event

28.5.3.2.3 Transfer and Buffer Rules

Buffers that are used to transfer data to and from an endpoint are defined using a Buffer Descriptor, which consists of one or more Transfer Request Blocks (TRBs). A CHN flag in the TRB is used to identify the TRBs that comprise a Buffer Descriptor. Therefore, a Buffer Descriptor refers to a consecutive set of TRB data structures where the CHN flag is set in all TRBs, except the last. Note that a Buffer Descriptor may consist of a single TRB, whose CHN flag will not be set.

Software communicates the location of the first TRB by using the Start Transfer command on an endpoint. Even endpoints that are not stream-capable use this command.The core fetches TRBs from external memory starting at the address provided in the Start Transfer command parameters, continuing in a linear fashion and following the Link TRB until a TRB is encountered that has its LST bit set to '1' or HWO bit set to '0'. Therefore, software must ensure that it has valid TRBs prepared before issuing the Start Transfer command to prevent the core from reading uninitialized memory.

Descriptor fetch requests are buffered within the hardware and handled separately from the progress of the current transfer. Therefore, it is possible that the core completes a transfer with XferComplete but still continues reading TRBs that it has not cached yet. If software is immediately de-allocating the memory for TRBs based on the XferComplete event, it is recommended that software issue an End Transfer command for the endpoint/transfer resource prior to de-allocating the memory. The completion of the End Transfer command flushes out any pipelined descriptor fetches and avoids a potential bus error.

While processing TRBs, two conditions may cause the core to write out an event and raise an interrupt line:

- TRB Complete:
  - For OUT endpoints, a packet is received which reduces the remaining byte count in the TRB buffer to zero.
  - For IN endpoints, an acknowledgement is received for a transmitted packet which reduces the remaining byte count in the TRB buffer to zero.
- Short Packet Received:
  - For OUT endpoints only. While writing to a TRB buffer, the endpoint receives a packet that is smaller than the endpoint's MaxPacketSize.

This table shows the USB 3.0 Core Actions Based on TRB Control Bits.

Table 28-18. USB 3.0 Core Actions Based on TRB Control Bits

Direction	TRB Complete Packet	ISP	IOC	CHN	LST	CSP	Action
IN	Yes	-	X	0	1	-	XferComplete event

Table continues on the next page...

Table 28-18. USB 3.0 Core Actions Based on TRB Control Bits (continued)

IN	Yes	-	0	X	0	-	No event
IN	Yes	-	1	X	0	-	XferInProgress event
OUT	Yes No	-	X	0	1	-	XferComplete event
OUT	Yes No	-	0	X	0	-	No event
OUT	Yes No	-	1	X	0	-	XferInProgress event
OUT	X Yes	X	X	X	X	0	XferComplete event [1]
OUT	X Yes	X	X	0	1	1	XferComplete event
OUT	X Yes	X	X	1	0	1	Search for CHN=0, accumulate IOC and ISP, then follow CHN=0 rules [2]
OUT	X Yes	X	1	0	0	1	XferInProgress event
OUT	X Yes	0	0	0	0	1	No event
OUT	X Yes	1	0	0	0	1	XferInProgress event

1. When a TRB receives whose CSP bit is 0 and CHN bit is 1 receives a short packet, the chained TRBs that follow it are not written back (for example, the BUFSIZ and HWO fields remain the same as the software-prepared value)
2. In the case of an OUT endpoint, if the CHN bit is set (and CSP is also set), and a short packet is received, the core retires the TRB in progress and skip past the TRB where CHN=0, accumulating the ISP and IOC bits from each TRB. If ISP or IOC is set in any TRB, the core generates an XferInProgress event. Hardware does not set the HWO bit to 0 in skipped TRBs. If the endpoint type is isochronous, the CHN=0 TRB will also be retired and its buffer size field updated with the total number of bytes remaining in the BD.

On XferComplete and XferInProgress events, status bits in the event indicate LST, IOC, Short Packet Received, or Bus Error status. In the "fast-forward" case, the IOC status is accumulated from the skipped TRBs, not just the TRB that received the short packet.

When the hardware writes back the TRBs, it updates the BUFSIZ field to represent the remaining unused buffer.

28.5.3.2.3.1 Number of TRBs Rule

- Software must set up only one TRB for a control setup or status stage.
- If software is preparing multiple transfers for an IN endpoint, it may be necessary to place a 0-length TRB between transfers that are MaxPacketSize aligned to indicate transfer boundaries to the host. This is not necessary, if the class driver and the host can handle bursting between transfers.
- If software is preparing multiple transfers for an OUT endpoint, it needs to place a MaxPacketSize TRB between transfers, if it expects the host to transmit a 0-length packet between transfers.

28.5.3.2.3.2 TRB Control Bit Rules

Transfer control bits must conform to the following restrictions:

For OUT endpoints, the CSP bit must be the same in every TRB within a Buffer Descriptor (either set or clear).

- The core autonomously checks the HWO field of a TRB to determine if the entire TRB is valid.
- Therefore, software must ensure that the rest of the TRB is valid before setting the HWO field to '1'. In most systems, this means that software must update the fourth DWORD of a TRB last. Every time software validates a TRB by setting HWO=1, it must also issue an Update Transfer command to the core.
- Software sets the HWO bit to 1 when it creates the TRB, and cannot modify it until hardware resets it to 0. However,
  - Software must detect when a "fast-forward" occurs on an OUT endpoint that receives a short packet, since some TRBs in a chain may still have their HWO bit set to 1 while belonging to software.
  - Hardware will not clear the HWO bit of a Link TRB. Therefore, software can only modify a Link TRB if the TRB prior to the Link TRB has its HWO bit set to 0.
- The LST bit must not be set to 1 for isochronous endpoints.
- For a Setup or Status TRB, set CHN=0, LST=1, and CSP=0.
- For the data stage of a control transfer, set CSP=0 in all TRBs and LST=1 in the last TRB of the data stage.
- For Link TRBs, the LST, CHN, IOC, ISP, CSP, and Stream ID fields are ignored and must be set to 0.
- The Link TRB's chain bit is implicitly equal to the chain bit of the TRB before it. If the TRB before it has CHN=1, then the Link acts as if it's CHN=1. If the TRB before it has CHN=0, then the Link acts as if it's CHN=0. A Link TRB cannot be the last TRB in a Buffer Descriptor.
- When CSP=1 and the core receives a short packet, it searches for the end of the Buffer Descriptor by finding the Normal TRB that has CHN=0. Therefore, it is illegal to setup a circular buffer with all Normal TRBs with CHN=1.

28.5.3.2.3.3 Buffer size rules and zero-length packets

The hardware contains a cache that holds four TRBs per transfer.

For IN endpoints, the following rules apply:

- The number of chained TRBs necessary to construct a single packet must never exceed 3 . A maximum of one link TRB can be present in the chain.
- If software wants to indicate a transfer completion to the host by sending a zero-length packet after a multiple of MAXPACKETSIZE, it must set up a zero-length TRB following the last TRB in the transfer.

For OUT endpoints, the following rules apply:

- If the first TRB has CHN=1, its buffer size must be ≥ 1 byte
- The total size of a buffer descriptor must be a multiple of MAXPACKETSIZE
- A received zero-length packet still requires a MAXPACKETSIZE buffer. Therefore, if the expected amount of data to be received is a multiple of MAXPACKETSIZE, software should add MAXPACKETSIZE bytes to the buffer to sink a possible zero-length packet at the end of the transfer.

For IN and OUT endpoints, the following rule applies:

- The BUFSIZ field in a link TRB must be set to 0.

28.5.3.2.3.4 Transfer setup recommendations

Software can either set up transfers before the host attempts to move data on an endpoint ("preset" transfers) or can set up transfers on demand ("on-demand" transfers). When using preset transfers, software can safely disable the XferNotReady event in the endpoint configuration. However, when using on-demand transfers, the XferNotReady event must be enabled and software may not use the "No Response" variant of the Update Transfer command. The XferNotReady event is issued when the host attempts to move data on an endpoint when one of the following conditions is present:

- No previous transfer was started with Start Transfer.
- Not enough hardware-owned (HWO=1) TRBs are available to handle the requested data movement. The XferNotReady event must not be disabled for control endpoints because the event is an integral part ofcontrol transfer handling.

Although there are many valid ways to set up transfers, it is recommended that you choose one of three general mechanisms:

- When software wants to set up one transfer at a time and has the entire buffer available for transfer, it must set up TRBs that point to the data buffers and in the last TRB it must set the LST bit and issue Start Transfer, which points to the first TRB location.
    - When the USB transfer completes, the core will notify the software through the XferComplete event. The LST bit will be set in the status field of the event. The XferComplete will also release the Transfer Resource.
    - A premature XferComplete event can happen before all the data buffers are exhausted, if there is a Bus Error or, in an OUT transfer a short packet has been received and CSP=0. During these conditions the TRB will have an updated

BUFSIZ field which represents the amount of buffer remaining after the successful part of data transfer.
    - Software can also set up an IOC bit TRB, so that it gets notified when data buffers are used up and can free them up sooner than waiting for the entire transfer to complete. On completing a TRB with IOC set, and not the last one, the core will issue an XferInProgress event with IOC set in the status field of the event.
- When software wants to set up one transfer at a time, but it has fewer data buffers available than the full transfer size, it must set up circular TRBs (using a Link TRB), and also set up IOC bits in the TRBs. Depending on buffer allocation and interrupt frequency it can set IOC for once in "x" number of TRBs.
    - As soon as the USB transfer for a TRB is completed, and if IOC is set, the core will generate an XferInProgress event with IOC set in the status field.
    - On seeing the event, software reuses the TRB and updates it with the next data buffer. It also issues an Update Transfer command, indicating it has updated a TRB.
    - If software is slow and hardware finds a TRB with HWO reset to 0, it waits for an Update Transfer command. Upon seeing the Update Transfer, it prefetches the TRB and continues the transfer.
    - When software reaches the end of the transfer, it sets the TRB LST bit. When hardware completes the TRB, it issues an XferComplete event and releases the Transfer Resource.
- When the Device software has multiple transfers to set up, it must set up circular TRBs and also set up CHN bits in all TRBs, except the last of each transfer. For OUT endpoints, multiple transfers can be supported when the CSP field is set to '0' if each transfer has a multiple of MaxPacketSize bytes, since a short packet will end the transfer. If multiple transfers may contain short packets, the CSP field must be set to '1' to enable the next transfer to continue even if a short packet is received. Depending on buffer allocation and interrupt frequency, it can either set the IOC for once in "x" number of TRBs or in the last TRB of each transfer.
    - For OUT endpoints, the transfer can finish prematurely due to a short packet from the host. In this case, the core processes the remaining TRBs, skipping the updates to these TRBs until it reaches TRB of the next transfer. While skipping these TRBs, if any of the TRB it encounters the interrupt setting of ISP or IOC, it will generate the corresponding event once and ignore the interrupt settings of the remaining TRBs until a TRB of next transfer is reached.
    - Device software can reclaim these skipped TRBs even though the HWO still indicates 1 (hardware-owned).
    - For IN endpoints, software cannot stop providing transfers while it is ending transfers with CHN=0 and LST=0, otherwise it is possible that the endpoint is left in a flow-controlled state on the USB.

28.5.3.2.4 Transfer Resource Usage and Transfer State

Software allocates one Transfer Resource for an endpoint during initialization. When software issues a Start Transfer command, this Transfer Resource is used. When an XferComplete event happens, the Transfer Resource is released back to software. Similarly, when End Transfer completes, the Transfer Resource is released.

Following is a typical transfer usage summary (not complete usage model):

- Software sets up data buffer(s) and TRB(s) in external memory with the TRB(s) pointing to the buffer(s).
- Software issues Start Transfer with the pointer to the first TRB in the command parameters.
- If software sets up all the buffers and TRBs at the same time (Host and Device Software negotiate the transfer size), then just one Start Transfer is enough, and software waits for a XferComplete event.
- If software did not set up all the TRBs, then every time software adds a new TRB (by setting HWO=1) it must issue an Update Transfer command. The hardware will fetch the TRB again and continue the transfer.
- The End Transfer command is used only during error conditions and not used during normal transfers. For example, if software has set up multiple transfers and if a USB Reset event happens, it must remove all the transfers from the queue using End Transfer with the ForceRM bit set 1.
- A transfer is completed when all the data has been transferred or a short packet has been received. If software is queuing in only single transfers at a time (normal method for mass storage - for ethernet over USB, software can set up multiple transfers using the Chain bit and Continue on Short Packet bit), once XferComplete interrupt has happened, the transfer is completed. When starting a new transfer, software now needs to issue the Start Transfer command.
- An OUT transfer's transfer size (Total TRB buffer allocation) must be a multiple of MaxPacketSize even if software is expecting a fixed non-multiple of MaxPacketSize transfer from the Host.

Figure 28-18. Transfer Usage State Machine.

28.5.3.2.5 Transfer Descriptions

The following sections describe typical transfer processes.

28.5.3.2.5.1 Non-Isochronous OUT Transfers

The sequence below describes a regular, non-isochronous OUT transfer (control-data/bulk/interrupt).

Table 28-19. Non-Isochronous OUT Transfer Sequence

Steps	Host	Device Core	Device Software
Software sets up data buffers before a host request			
Step 1	-	-	Software sets up Normal TRBs and enables DMA by issuing Start Transfer.
Step 2	-	Device fetches the TRB and caches it.	-
Step 3	Host sends data packets	-	-
Step 4	-	Core receives the packets in RXFIFO and stores them in to TRB data buffers through DMAs. While the RXFIFO has space and the BUS DMA keeps up with the host data transfers, the device receives all the packets. Once all the data is received, the core updates the TRB status and then generates a XferComplete event. In SS mode, if GRXTHRCFG.USBRxPktCntSel is set to, • 0: the NumP sent depends on the DCFG.NumP and bMaxBurstSize • 1: the NumP sent depends on the RXFIFO space available when the packet is received When DMA is slow or CRC fails: • If the RXFIFO overruns or CRC error is detected, device times out in Non-SS mode or asks for the same data packet (through ACK/SeqNumber/Retry Bit) in SS mode. • In HS mode, on NAK the host will issue PING and device will ACK it (if there is enough space in the RXFIFO) and the host resumes with data again. In SS mode, the host will send the requested data. If the device receives any packet with other than the requested data sequence, it will ignore it.	-
Step 5	-	-	Software processes the XferComplete event.
Software sets up data buffers after the host request.			
Step 6	Host sends data packets.	-	-
Step 7	-	Since DMA is not set up, the device will send NAK in Non-SS mode and will send NRDY in SS mode to the host. It also generates an XferNotReady event.	-
Step 8	-	-	On seeing the XferNotReady event, Software sets up TRBs and enables DMA by issuing Start Transfer.
Step 9	-	Steps 2-5 are followed.	-

Table continues on the next page...

Table 28-19. Non-Isochronous OUT Transfer Sequence (continued)

Steps	Host	Device Core	Device Software
Step 10	-	If the core encounters a short packet before the transfer size data programmed in the TRB is received, it skips the remaining TRBs of the transfer. If the core encounters control bits LST, IOC, or ISP it will generate events and, in the case of an LST TRB, the stream will be completed. Software has to reclaim any TRBs skipped with HWO=1.	On detecting a skipped TRB condition, software reclaims the TRBs with HWO=1.

28.5.3.2.5.2 Isochronous OUT Transfers

This section describes the difference between isochronous and non-isochronous OUT transfers referring to the sequence in Non-Isochronous OUT Transfers

- In Step 2, if the TRB is not fetched in time, the core discards the packet. In addition subsequent packets in the service interval also will be dropped.
- In Step 4, if DMA is enabled and if RXFIFO overrun happens or CRC fails, the core discards the packet. In addition subsequent packets in the service interval also will be dropped.
- In Step 7, if DMA is not enabled, the core drops the data and generates XferNotReady event.

In general, the TRB chain of a service interval will be released if the core receives DATA0 PID in HS USB 2.0 mode or receives a packet with LPF set in SuperSpeed mode.

Special scenarios are:

- If the core misses this last packet indication, it releases TRB at the service interval boundary.
- If the core encounters a short packet before the transfer size data programmed in TRB is received, it skips the remaining TRBs of the transfer.
- If the core encounters control bits 'LST' or 'IOC' or "ISP" it will generate event and, in the case of LST TRB, the stream will be completed, but software has to reclaim TRBs if any TRBs skipped with HWO=1.

28.5.3.2.5.3 Non-isochronous IN transfers

The sequence below describes a regular, non-isochronous IN data transfer (control-data/bulk/interrupt).

Table 28-20. Non-isochrnous IN transfers

Steps	Host	Device Core	Device Software

Table continues on the next page...

Table 28-20. Non-isochrnous IN transfers (continued)

	Software sets up data buffers before a host request		
Step 1	-	-	Software sets up TRBs and enables DMA by issuing Start Transfer.
Step 2	-	Device fetches the TRB, caches it, and prefetches the data into TXFIFO. Only in SS mode, the core sends ERDY (Async ERDY, even before host requesting data provided endpoint is in flow controlled state. LSP flow is same regardless of if the host asked for it.)	-
Step 3	Non-SS host issues a token request, SS host issues ACK TP to request data.	-	-
Step 4	-	If DMA fetches keep up with host requests, the device sends all the data, updates the TRB status, and then generates an XferComplete event when ACKs for all the packets are received. When DMA is slow: • If the TXFIFO becomes empty on a host data request, the device sends a NAK in Non-SS mode. In SS mode, the device sends NRDY, and when data is available in TXFIFO it sends ERDY and waits for the host request again. • If the TXFIFO becomes empty in the middle of a packet transfer (happens only in Threshold mode), in Non-SS mode the device will invert the CRC. In SS mode, it terminates with EDB after appending the corrupted CRC. The core waits for the host to retry the packet before fetching the data again. If the host retries a packet: • In Non-SS mode, a transmitted packet is kept in the TXFIFO until the ACK arrives (this is immediate in Non-SS mode). If the host retries a packet, it is immediately sent out from the TXFIFO. • In SS mode, before ACK can be received, the core could have sent additional packets in a burst. When a retry happens, NRDY is sent to the host and the TXFIFO is flushed. A DMA command is issued on the master bus to prefetch data from where the host is asking. Once data is fetched into the TXFIFO, the device sends ERDY and waits for host request again. • In Threshold mode, when the TX threshold amount of data is fetched, the device sends ERDY to the host.	-
Step 5	-	-	Software processes XferComplete event.
	Software sets up data buffers after a host request.		

Table continues on the next page...

Table 28-20. Non-isochrnous IN transfers (continued)

Step 6	Non-SS host issues a token request. SS host issues ACK TP to request data.	-	-
Step 7	-	Since DMA is not set up, the device sends s NAK in Non- SS mode, or NRDY in SS mode, to the host. The core also generates an XferNotReady event	-
Step 8	-	-	On seeing the XferNotReady event, software sets up Normal TRBs and enables DMA by issuing Start Transfer.
Step 9	-	Steps 2-5 are followed.	-
Step 10	-	When software knows it is talking to a host that always ends transfers with short packets, and therefore always requires a zero-byte packet to end a transfer, it includes a zero-byte TRB (CHN=0) after the last TRB which is an even multiple of MaxPacketSize. When software does not know if the host is going to do another IN token after receiving the exact number of bytes it expected (and it was a multiple of MPS), it does not include a zero-byte TRB, and sets LST in the last TRB. If the host returns with another IN for a zero-byte packet, it uses the XferNotReady/Start Transfer mechanism to add one zero-byte TRB. This is called "On- Demand."	-

28.5.3.2.5.4 Isochronous IN Transfers

This section describes the difference between isochronous and non-isochronous IN transfers referring to the sequence in Non-isochronous IN transfers.

- In Step 4, if the TXFIFO is empty when host IN request arrives, the core returns a zero length packet in both Non-SS and SS modes.
- In Step 7, the device returns zero length packet and generates an XferNotReady event.

Special scenarios:

- When data is fetched and there is no request from the host (possibly due to a missing request or corrupted request) the data will be flushed at the end of the service interval (corresponding Micro- Frame) boundary and the core moves onto fetching next service interval data. If there are some TRBs that are not serviced in the service interval, the core will skip these TRBs.
- If the core encounters control bits LST or IOC, it will generate an event, and in the case of LST TRB, the stream will be completed, but has to reclaim TRBs if any TRBs skipped with HWO=1.

28.5.3.2.6 Handling ENDPOINT_HALT

On receiving a SetFeature (ENDPOINT_HALT) control transfer, software issues a Set Stall command on the endpoint. Because of the delay from when software issues the command and when it is recognized by the core, other transfer events (XferInProgress, XferComplete) may be received prior to the endpoint being halted. For a description of the Set Stall endpoint command and its effects, see Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).

On ClearFeature (ENDPOINT_HALT), software must first remove all pending transfers for the endpoint through the End Transfer command. It may then issue a Clear Stall command on the endpoint followed by Start Transfer to start transfers again. For a description of the Clear Stall endpoint command and its effects, see Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).

28.5.3.2.7 Handling L1 Event During a Transfer

This section discusses the scenario in which the software already issued the Start Transfer command but the LPM occurred before or during the data transfer.

When an IN transfer is in progress, (that is, started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to determine if the TXFIFO is empty or not. If the TXFIFO is not empty, it can initiate a remote wakeup.

NOTE

- The GDBGFIFOSPACE register is a debug register which gives the 'Space Available" for the endpoint selected with a write prior to the read. If this space available is less than the particular TXFIFO's maximum size, then the TXFIFO is not empty.
- Software can enable the Link State change event for USB 2.0 as there are not a lot of events compared to SS.

28.5.3.3 Isochronous Transfer Programming Model

Isochronous endpoints get guaranteed service by the host during a "Service Interval". The service interval for an endpoint is communicated to the host via the endpoint descriptor, and it is configured in the hardware through the DEPCFG command. However, unexpected behavior on the bus may prevent the host from servicing the endpoint as frequently as expected.

Although the endpoint receives "guaranteed" service during the interval, there are no bus-level acknowledgments, which means that there is no guarantee that the host (IN endpoints) or the device (OUT endpoints) receives correct data.

Therefore, the isochronous programming model differs from the bulk programming model to accommodate these two factors:

- No bus-level acknowledgement of packet transmission or reception
- No guarantee that all the data setup for an interval has been transmitted or received

For isochronous endpoints, software has the following responsibilities:

- Set up enough TRBs to keep up with the rate of data transmission or reception
- For FIFO-based IN endpoints, guarantee the validity of the TX data at least 1 micro Frame before the beginning of the interval that the data will be transmitted

Hardware has the following responsibilities:

- Transmit or receive data at the appropriate bus time
- For FIFO-based IN endpoints, delay fetching until 1 micro Frame before the beginning of the interval that the data will be transmitted
- Maintain a 1-to-1 correspondence between Buffer Descriptors and the intervals on the bus
- Report missed isoc intervals and update the buffer sizes in TRBs to reflect the amount of data moved

28.5.3.3.1 Definitions

- bInterval: The field in an endpoint descriptor used to communicate the service interval of the device. Its value ranges from 1 to 16, however, the USB 3.0 core supports a range of 1 to 14.
- Service interval: An integral number of microframes (2^[bInterval-1]) during which the host will poll the device to move data.
- Beginning of interval: The interval begins when the least significant (bInterval-1) bits of the bus time are all 0's.
- End of interval: The interval ends when the least significant (bInterval-1) bits of the bus time are all 1's.
- Data payload: The number of bytes to be moved during the service interval. It will be transmitted using MaxPacketSize packets.
- Buffer descriptor: A set of one or more TRBs with CHN=1 and the last one having CHN=0. The group of TRBs represents the data buffers for one service interval.
- Last TRB of a Buffer Descriptor: The TRB within a Buffer Descriptor that has CHN=0. When an interval has completed, this TRB will contain the total remaining buffer size of the Buffer Descriptor andus also the completion status of the interval.
- Microframe (µF): A unit of time on the USB that lasts 125 µs.

- Start of Frame (SOF): The beginning of a microframe.
- FIFO-Based Isoc IN: A sub-type of Isoc IN endpoints that has the characteristic of always assuming the HWO bit in a TRB is '1' and delays fetching of transmit data.
- Prefetch delta: For FIFO-based IN endpoints, TX data may be prefetched as early as 1 μF before the beginning of a Service Interval, but never earlier.
- Retire a TRB: When HW sets the HWO bit to 0 in a TRB and writes it back.
- Retire a Buffer Descriptor: When HW retires at least the first and last TRB of a Buffer Descriptor.
    - Other TRBs of the Buffer Descriptor may not be retired if an interval was missed (IN or OUT) or if a short packet/last packet is received (OUT). In this case, software may reclaim those TRBs even though HWO=1.
- Missed interval: When the device does not move all the data in an interval. This may occur when the host does not poll for all the data, or because of application-side delays that prevent all the data from being moved.
    - For IN endpoints, this occurs when the host does not request all the data prepared in the Buffer Descriptor, or not all the data is fetched from the system bus in time.
    - For OUT endpoints, this occurs when the host does not send a packet, a packet is dropped due to CRC error, or the system bus is too busy to accept the data.

28.5.3.3.2 Endpoint configuration

An isochronous endpoint is setup in much the same way as a bulk endpoint, with the following exceptions:

- The bInterval_m1 value in DEPCFG Parameter 1 must be set to the value reported in the endpoint descriptor minus 1. When the core is operating in Full Speed, bInterval_m1 must be set to 0.
- The MaxPacketSize value in DEPCFG Parameter 0 must be set to the same value reported in the endpoint descriptor.
- The "FIFO-based" bit in DEPCFG Parameter 1, bit 31, must be set for FIFO-based isochronous endpoints.
- For the best performance, set the TXFIFOs corresponding to Isochronous IN endpoints to a high priority.

After the endpoint is configured, it can be enabled by setting the appropriate bit in the DALEPENA register.

28.5.3.3.3 Transfer configuration

Software describes isochronous transfers through a series of Buffer Descriptors. Each Buffer Descriptor corresponds directly to one service interval of data. The fields within an isochronous TRB are the same as bulk TRBs with the following exceptions:

- The SOF field in an IN endpoint TRB is a don't care. For an OUT endpoint, the core will write this field with the timestamp of the last packet received into the TRB's buffer. This information is not needed for normal operation, only for debug purposes.
- For High-Speed, High-Bandwidth IN endpoints, a maximum of 3 packets can be sent during an interval. The PktCntM1 field ([25:24] of the 3rd DWORD) must be set to the (number of packets in the Buffer Descriptor - 1). For example, if 3 packets are to be transmitted during the interval, this field must be set to 2. For Super-Speed and OUT endpoints, this field is a don't care.
- The first TRB in a Buffer Descriptor must have the TRBCTL field set to the "Isochronous-First" type while all others have this field set to "Isochronous".
- The ISP bit is renamed IMI (Interrupt on Missed Interval) and should be set if software wants to receive an XferInProgress event when at least 1 packet is missed within an interval.
- The IMI bit should be set to the same value in all TRBs of a Buffer Descriptor.
- The CSP bit must be set to 1 (short packets cause the hardware to move to the next Buffer Descriptor, they do not end an isochronous transfer).
- The LST bit should be set to 0 (isochronous transfers normally continue until the endpoint is removed entirely, at which time an End Transfer command is used to stop the transfer).

All other bits and fields (IOC, HWO, CHN, BPTR, BUFSIZ) retain the same behavior as they have for bulk endpoints. If software needs to receive an interrupt after every service interval, it should set the IOC bit to '1' in each TRB. However, if software wants to reduce the interrupt frequency and the application can tolerate some latency, the IOC bit can be set to '0' and the core will not generate a XferInProgress event when the TRB is completed.

For OUT endpoints, the Buffer Descriptor must describe a total buffer size of:

```
(Mult+1)*MaxBurst*MaxPacketSize
```

28.5.3.3.4 Starting a Transfer

The hardware will report the bus time that the host starts polling the endpoint inside the XferNotReady event. Software will use this value as a reference when issuing the Start Transfer command to serve as time synchronization with the host.

1. Set up TRBs and data buffer for the endpoint.
2. After configuring and enabling the endpoint, wait for an XferNotReady event.

The XferNotReady event will contain the time that the host started polling the endpoint in the upper 16 bits.

3. Issue the Start Transfer command to the endpoint with a future microframe time written into the upper 16 bits of the DEPCMD register. The future microframe time must be a value that is an integral multiple of intervals after the time reported in the XferNotReady event and aligned to the beginning of an interval. For example, if bInterval is 3 (4 microframes), and the XferNotReady time is 2, the value can be 4, 8, 12, and so on. The future microframe time must also be no greater than 4 seconds past the time reported in the XferNotReady event.
4. If the future microframe time has already passed when the command is received, the core will respond with an error (bit 13 in the Command Complete event).

In this case, software must issue End Transfer, then wait for another XferNotReady event and attempt the command again with a time that is further in the future.Otherwise, the first Buffer Descriptor will be used for the interval starting with the microframe time specified in the Start Transfer command.

28.5.3.3.5 Core Behavior During an Interval

After a transfer has been started, the hardware will perform the following functions for IN endpoints:

1. Fetch TX data as early as one interval prior to the beginning of the interval (call it A) if the HWO bit is set to one in the TRB.
2. Decrement the buffer size of each TRB as packets are transmitted.
3. Retire TRBs when their buffer size has reached 0, issuing an XferInProgress event if the IOC bit is set.
4. If the next interval (B) starts before all the packets have been transmitted for interval A:
   a. Flush the TXFIFO.
   b. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status.
   c. Retire the Buffer Descriptor of interval B with a "Missed Isochronous" status.
   d. See Checking interval status for a description of how software can determine that an interval ended unexpectedly.
   e. Go to step 1 to prepare for interval C
5. Otherwise, if all the TRBs of interval A are completed, the hardware will prepare for interval B.
6. If the host completes an interval by polling for all the data, but then it polls the endpoint again during the same interval, the hardware will respond with a zero-length packet and no interrupt will be made to software.

7. If the host polls the endpoint prior to the time specified in the Start Transfer command, the hardware will respond with a zero-length packet and no interrupt will be made to software.
8. If the host polls the endpoint during the expected interval and the hardware has no data prepared, the core will respond with a zero-length packet, but no XferNotReady event will be generated because the transfer is active. When the next interval starts, the XferInProgress event is generated (based on ISP and IMI) with the "Missed Isochronous" status, which is the way software finds out that this occurred.

For OUT endpoints, the hardware performs the following functions:

1. If a packet is received for the correct interval represented by the Buffer Descriptor (call it A).
   a. Write the packet into the buffer.
   b. Decrement the buffer size of the TRB.
   c. Write the timestamp of the received packet into the TRB.
   d. Retire a TRB when its buffer size reaches 0, issuing an XferInProgress event if the IOC bit is set.
2. If a short packet or packet with the last packet flag (lpf) is received for the correct interval.
   a. Write the packet into the buffer.
   b. Retire the Buffer Descriptor of interval A with the TRBSTS set to 0.
   c. If the IOC bit is set in the last TRB of the Buffer Descriptor, issue an XferInProgress event with bit [13] set.
   d. Go to step 1 to prepare for interval B.
3. If a packet is received for the correct interval but it has an error (CRC error, RXFIFO overflow), the core will not increment its expected sequence number value which causes future packets within the same interval to be dropped.
4. If a packet is received at a bus time prior to the time specified in the Start Transfer command, it will be dropped.
5. If a packet is received for the correct interval (A), but not enough buffer space is available for the core to write the packet (due to the HWO bit still '0' in one or more of the TRBs of the Buffer Descriptor), no XferNotReady event will be generated. The core will behave as follows:
   a. Wait until software sets the HWO bit to '1' and issues an Update Transfer command
   b. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status
   c. Go to step 1 to prepare for interval B.
6. If a packet is received for the next interval (B) before all the packets have been received for interval A:
   a. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status.
   b. Retire the Buffer Descriptor of interval B with a "Missed Isochronous" status.

c. See section Checking interval status for a description of how software can determine that an interval ended unexpectedly.
d. Go to step 1 to prepare for interval C.

Special Considerations for Isochronous OUT Endpoints

For OUT isochronous endpoints, the hardware detects a missed interval when the host sends a data packet in a future interval. It does not detect the missed interval at the exact interval boundary. Therefore, if the host abruptly stops sending isochronous OUT packets, there will be no interrupt or event indicating this to software. Software should use one of the following mechanisms to detect the interruption of isochronous OUT traffic:

1. The ultimate consumer of the isochronous OUT data will detect an underflow.
2. The device driver can use a timer detect that no XferInProgress events have been received for multiple intervals.

If this occurs, software should issue an End Transfer command to the endpoint and wait for a XferNotReady event which signals that the host is ready to resume isochronous traffic.

28.5.3.3.6 Checking interval status

As packets are transmitted or received during an interval, the buffer size of each TRB will be decremented. When the host is operating normally and polling for all the interval data, the buffer size of all the TRBs of an IN endpoint Buffer Descriptor will be zero after the interval has completed. For OUT endpoints, if the host sends less data than the Buffer Descriptor was setup for, the remaining buffer size may be greater than 0.

When the interval completes, the final remaining buffer size and missed isoc status is written to the last TRB of the Buffer Descriptor. If MissedIsoc is set, then it means the BUFSIZ is not accurate and may indicate that more data was transmitted or received than in reality. If the MissedIsoc is not set, it means the BUFSIZ field is correct.

When hardware detects the end of an interval (including normal and abnormal ends), it performs the following:

- If it has not already been retired, the first TRB of the Buffer Descriptor will be retired.
- Any non-first TRBs with CHN=1 that had not already been retired will not be written back. HWO will still be 1, and software can reclaim them for another transfer.
- The last TRB of the Buffer Descriptor will be retired with:
  - HWO = 0.

  - BUFSIZ = The total remaining buffer size of the Buffer Descriptor.
  - TRBSTS = "Missed Isoc" if any packets were missed, zero otherwise.
- If the IOC bit is set in the last TRB, or the IMI bit is set and packets were missed, hardware will issue an XferInProgress event

Software can get notification of an interval completing by setting the IOC bit in the last TRB of the Buffer Descriptor. The IOC bit may also be set in any other TRB of the Buffer Descriptor if software wants earlier notification that a TRB has completed. Software can also get only a notification of an interval completing unexpectedly by setting the IMI (Interrupt on Missed Isoc) bit in the last TRB of the Buffer Descriptor. When an interval completes unexpectedly and either the IOC or IMI bit is set in the TRB, the MissedIsoc bit (15) of the XferInProgress event will be set. The following table shows which events are generated in each scenario:

Table 28-21. TRB event generation events

Scenario	IOC	IMI	Action
TRB completed with a MaxPacketSize packet	0	0	No interrupt
	0	1	No interrupt
	1	0	XferInProgress (IOC=1)
	1	1	XrefInProgress (IOC=1)
TRB completed with a short packet	0	0	No interrupt
	0	1	No interrupt
	1	0	XferInProgress (Short=1, IOC=1)
	1	1	XrefInProgress (Short=1, IOC=1)
Interval completed due to missed packet (missed isoc)	0	0	No interrupt
	0	1	XferInProgress (MissedIsoc=1, IOC=1)
	1	0	XferInProgress (MissedIsoc=1, IOC=1)
	1	1	XrefInProgress (MissedIsoc=1, IOC=1)

It is normal to lose two intervals at a time when an error occurs during one interval. One interval is lost because of the host (which may not be polling enough) and the second interval is dropped because the device needs time to synchronize up to the next (third) interval.

28.5.3.3.7 Adding intervals to a transfer

Because isochronous endpoints represent a stream of data, the TRBs of an isochronous endpoint will normally be setup in a circular list, such as:

- TRB 1 (CHN=0, IOC=1)
- TRB 2 (CHN=0, IOC=1)
- Link (to TRB 1)

Example: Assume TRB 1 represents the data for the first interval and TRB 2 represents the data for the second interval. Because IOC=1, when the core completes the first interval, it will issue an XferInProgress event which indicates to software that TRB 1 can be analyzed to retrieve the results from interval 1. TRB 1 can the be re-used to represent the data for the third interval by setting HWO=1 and issuing an Update Transfer command.

Each time software sets up TRBs for a new interval, it must follow the guidelines in the Transfer Configuration section. Software will set up TRBs for future intervals when it obtains those buffers from another software layer or when TRBs are retired by the core.

28.5.3.3.8 Moderating Events

During a transfer, software may enable or disable any endpoint-specific event by re-issuing the DEPCFG command with the "Config Action" set to "Modify" and a modified DEPEVTEN field. All other field. All other fields in all other parameters must remain the same as the initial endpoint configuration.

28.5.3.3.9 Other Types of Isochronous Endpoints

The above description of the isochronous endpoint programming model assumes that software is setting up buffers in external memory that apply to certain intervals of data. This is called a "buffer-based" isochronous endpoint.

However, there is another mode that the hardware supports to accommodate other mechanisms of sourcing or sinking isochronous data, called "FIFO-based" and the types are defined by bits [31:30] of DEPCFG Parameter 1. This table illustrates the differences between the models:

Table 28-22. Isochronous Endpoints Models

Type	FIFO-Based [31]	Bulk-based [30]	TRB Fetch	TRB Write back	TX DMA	Retire Buffer Descriptor at passed interval
Buffer-based	0	0	When HWO=1	Yes	When HWO=1, no earlier than one interval ahead of time.	Yes
FIFO-based	1	0	When internal cache space available	No	No earlier than 1 μF before the interval	Yes
Invalid	1	1				

28.5.3.3.9.1 FIFO-based isochronous IN Endpoints

The software of some applications is unable to keep up with the short latency and high bandwidth of isochronous traffic and require the data to be sourced and done a sync through external FIFOs. One example is that there are two external TXFIFOs, one for each interval, and an address-to-FIFO-pop logic is implemented so that when the core attempts to read from interval 1's address, the translation logic converts this into a pop signal for interval 1's TXFIFO. When the core attempts to read from interval 2's address, the translation logic converts this into a pop signal for interval 2's TXFIFO.

To describe a FIFO-based implementation, software sets up an endpoint with the "FIFO-based" configuration bit set. This type of endpoint ignores the HWO bit in all TRBs, assuming that the TRB is always valid, and that the core should never write it back. Software chooses the buffer pointers within the TRBs to correspond to the address needed by the translation logic to specify which FIFO should be popped. By also using the CHN bit, headers and payload can be concatenated from different FIFOs. For example, if there are 4 external FIFOs: HA, PA, HB, PB, where HA/HB contain the 4 byte header for 2 intervals and PA/PB contain the 512 byte payload for 2 intervals, this can be described by using the following 5 TRBs:

1. BUFPTR=HA, IOC=0, CHN=1, BUFSIZ=4
2. BUFPTR=PA, IOC=1, CHN=0, BUFSIZ=512
3. BUFPTR=HB, IOC=0, CHN=1, BUFSIZ=4
4. BUFPTR=PB, IOC=1, CHN=0, BUFSIZ=512
5. Link to (1)

Every interval, the core will be creating a 516 byte packet that consists of 4 bytes from the header FIFO and 512 bytes from the payload FIFO. However, if the host does not poll for the packet, the core will skip 1 or 2 intervals of popping, depending on whether it has already started reading the next interval. External logic (or software) is responsible for flushing and refilling alternate FIFOs so that the core is always reading the correct data for the next interval. In normal intervals, the FIFOs will be empty, but when an interval is missed, there may still be data present in the FIFOs.

Software Requirements:

- The "FIFO-based" bit must be set in the DEPCFG when configuring the endpoint.
- No field within any TRB may be changed after the Start Transfer command is issued.
- Data must be valid in the external FIFO at least 1 uF before the beginning of the interval for which the data is intended.

Core Behavior:

- The earliest the core will read from the FIFO will be 1 μF before the beginning of the interval for which the data is intended.

- The core will not write back the TRB after it has completed it.
- The core will re-read the TRB from external memory even though it is not allowed to change.
- The only indication of missed intervals is the XferInProgress event if IOC or IMI is set in the TRBs. No indication will be made of how much data was actually transmitted.

28.5.3.3.9.2 FIFO-based isochronous OUT Endpoints

The FIFO-based model is also supported for OUT endpoints. In this case, core writes to specific addresses will be translated into external FIFO pushes. The same example from above (4 byte headers and 512 byte payloads) can be applied to OUT endpoints where the headers are split from the payload as the core receives 516 byte packets. The same software requirements and core behavior apply (for example, no writebacks and no indication of how much data was received). External logic assumes that if less than the expected amount of data is pushed into an interval's FIFO, the interval is invalid.

28.5.3.3.10 End a Transfer

Since isochronous endpoints have no bus-level acknowledgements, it is not necessary to set LST=1 in a TRB to gracefully end an isochronous transfer. Software can issue an End Transfer command to end an isochronous transfer. The core will wait until it can complete operations for the endpoint before returning the Command Complete event in response.

28.5.3.4 Control transfer programming model

Control transfers follow a flow of setup/data/status. On the USB bus, there are many error scenarios that can disrupt this flow. The hardware has a special mechanism that automatically recovers from these scenarios as long as software follows this single control transfer programming model.

Figure 28-19. Software flow for control transfers

If hardware does not receive a setup packet, it discards the unexpected data and status stages. It does not issue XferNotReady to the application. The hardware will also not generate a XferNotReady event if it receives a setup packet before a TRB is prepared for it. As control endpoints are bi-directional, the IN and OUT directions of the endpoint are used for the different stages as follows:

Control read:

- EP0 - Setup OUT
- EP1 - Data IN
- EP0 - Status OUT/Status TP

Control write or 2-stage transfer:

- EP0 - Setup OUT
- EP0 - Data OUT (if present)
- EP1 - Status IN/Status TP

Set STALL is always issued on EP0, and the XferComplete/XferNotReady events are generated on the correct direction. For other control endpoints, substitute EP0 with the OUT direction (such as, 2, 4, 6) and EP1 with the IN direction (such as, 3, 5, 7). The same programming model is used for USB 2.0 and USB 3.0 control transfers.

28.5.3.4.1 Two-stage control transfer programming model

1. Software sets up a Setup TRB and issues Start Transfer on EP0 pointing to the Setup TRB.
2. After XferComplete is received, software interprets the setup bytes. If they are bad, issue Set Stall on EP0 and go back to the step 1. If a XferNotReady (Data/Status) event is received before the XferComplete event for the Setup stage, issue Set Stall. This is an error case where the host is attempting to move data or start the status stage for a previous control transfer that has already completed.
3. Wait for an XferNotReady event for the Status stage which will occur on EP1. If an XferNotReady event for the Data stage is received (either direction), issue Set Stall on EP0 and go back to the step 1. This is an error case where the host is attempting to start the data stage when the setup bytes did not indicate a data stage was present.
4. After XferNotReady (Status) is received, start the status stage by issuing Start Transfer on EP1 pointing to a valid Status-2 TRB.
5. After XferComplete is received on EP1, go back to the step 1.

Although not required, software may look at the Status TRB and examine the Setup Pending bit in the TRB status field. If it is set, it means that this control transfer was aborted on the USB bus and that the host did not receive the Status stage ACK.

28.5.3.4.2 Three-stage control transfer programming model

1. Software sets up a Setup TRB and issues Start Transfer on EP0 pointing to the Setup TRB.
2. After XferComplete is received, software interprets the setup bytes. If they are bad, issue Set Stall on EP0 and go back to the step 1. If a XferNotReady (Data/Status) event is received before the XferComplete event for the Setup stage, issue Set Stall. This is an error case where the host is attempting to move data or start the status stage for a previous control transfer that has already completed.
3. Start the data stage by issuing Start Transfer on EP0 (control write) or EP1 (control read) pointing to a valid Control-Data TRB.

   a. If an XferNotReady (Data) event is received for the incorrect direction, software must issue an End Transfer for the data stage it has already started, then issue Set Stall. This is an error case where the host is attempting to move data in the wrong direction.
   b. If a XferNotReady (Data) event is received for the correct direction, ignore the event and continue to wait for a XferComplete event.
4. After the XferComplete event is received, software waits for an XferNotReady event. Although not required, software may look at the completed Data TRB(s) and examine the Setup Pending bit in the TRB status field and the BUFSIZ. If the Setup Pending bit is set or the BUFSIZ is non-zero, it means that this control transfer was aborted on the USB bus and that the host did not complete the data stage.
5. If a XferNotReady (Data) event is received after the XferComplete for the Data stage, it could mean one of two things:
   a. This host is trying to complete the data stage by moving a 0-length packet. This can occur if the data stage was an exact multiple of max packet size. If this is the case, software sets up an extra TRB (with a BUFSIZ of max packet size for control writes, or 0 for control reads), issues Start Transfer, and goes back to the step 4.
   b. This host is trying to move more data than specified in the wLength field of the setup bytes. In this case, software issues Set Stall on EP0 and goes back to the step 1.
6. When the XferNotReady (Status) event is received, software interprets the data stage (for example, number of bytes transferred, content of the control write buffer) and decides if the data stage was successful. If not, it issues Set Stall on EP0 and goes back to the step 1.
7. Start the status stage by issuing Start Transfer on EP1 (control write) or EP0 (control read), pointing to a valid Status-3 TRB.
8. After XferComplete is received, go back to the step 1.

Although not required, software may look at the Status TRB and examine the Setup Pending bit in the TRB status field. If it is set, it means that this control transfer was aborted on the USB bus and that the host did not actually receive the Status stage ACK. When device software sets STALL with the Set Stall command, it is possible that this command can be delayed on the chip bus due to latency or the chip bus servicing other agents. In this case, software can receive an XferNotReady event if the core detects no TRB active before it detected the Set Stall. Software must service these events knowing that it stalled the control command.

28.5.3.4.3 Handling fewer requests than wLength

In a control IN transfer, if the host asks for less data than wLength, the core skips the TRBs of the data stage and generates an XferComplete event after encountering the TRB that has LST=1. Hardware sets HWO to 0 in those TRBs that it skipped and in the last TRB. In a control OUT transfer, if the host sends less data than wLength, the core treats this like a short packet received into a TRB with CSP=0. It immediately generates a XferComplete event, not setting the HWO bit in the remaining TRBs to 0. Software needs to reclaim those TRBs that still have HWO=1.

28.5.3.4.4 Control OUT transfer examples

The sequence in the following table describes two and three stage control OUT transfers.

Table 28-23. Control OUT transfer

Steps	Host	Device core	Device software
	Setup stage		
1	-	-	Software sets up the Setup TRB and enables DMA by issuing Start Transfer to EP0 (endpoint 0, OUT direction).
2	-	Device fetches the TRB and caches it.	-
3	Host sends SETUP packet.	-	-
4	-	SETUP packet is received in the RXFIFO. There are 24 bytes allocated for each control endpoint, which guarantees reception of the three back-to-back SETUP packets. An ACK handshake is returned in non-SS mode. An ACK TP with NumP=1 is returned in SS mode. After SETUP, bytes are transferred through the DMA, and the core issues an XferComplete event.	-
5	-	If the host starts a Data/Status stage, the core returns NAK in Non-SS mode or NRDY TP in SS mode. If the SETUP request is invalid, core returns STALL (when software sets STALL) in non-SS mode. In SS mode the core returns ERDY, and when the Data/Status stage is received it returns STALL.	Software processes the XferComplete event and decodes the SETUP bytes. If a Data stage is required, it will set up one Control-Data TRB and possibly subsequent Normal TRBs, then issue Start Transfer to EP0 (endpoint 0, OUT direction). If a Data stage is not needed, it will skip to the Status stage. If the SETUP request is invalid, software instead sets the STALL bit by issuing Set Stall command for EP0 (OUT direction).
	Data stage		
6	-	Device fetches the TRB and caches it. In SS mode only, the core will send ERDY.	-

Table continues on the next page...

Table 28-23. Control OUT transfer (continued)

Steps	Host	Device core	Device software
7	Host sends data packets.	-	-
8	-	The NumP for control transfer will be 1. In addition, if another SETUP is received before the data transfer happens on USB, the core will skip the data transfer that software sets up without waiting for it to happen on the USB bus, and send XferComplete (when LST=1 TRB encountered) with the SetupPndg status bit set. Software has to reclaim the TRBs with HWO=1 in the skipped TRBs.	-
9	-	If the host starts a Status stage, the core returns NAK in Non-SS mode or NRDY TP in SS mode.	On seeing the event: • Software decodes the control write data. If the Data stage is OK, then it will wait for XferNotReady with "Control Status Request" set, set up a Control-Status-3 TRB (0-bytes), and enable DMA by issuing Start Transfer to EP1 (endpoint 0, IN direction). • If the Data stage is not OK, then software sets STALL by issuing Set Stall command to EP0.
	Status stage		
10	-	Device fetches the TRB and caches it. In SS mode only, the core will send ERDY	Software sets up a Status stage TRB only after the Data stage of a three-stage control transfer is completed and an XferNotReady event occurred.
11	Host requests status packets.	-	-
12	-	In Non-SS mode, the device will send a zero-length packet if STALL is not set; otherwise, it will send STALL. In SS mode, if STALL is not set, then it sends ACK TP with NUMP=1 else it sends STALL. If another SETUP is received before the status transfer on USB, the core will send XferComplete with the SetupPndg status bit set. The STALL bit will be cleared by the core whenever it receives a SETUP packet. SETUPs are always accepted.	-
13	-	-	On seeing the event: • If there are no other errors, then the transfer has completed successfully.

28.5.3.4.5 Control IN transfer examples

The sequence in the following table describes two and three stage control IN transfers.

Table 28-24. Control IN transfer

Steps	Host	Device Core	Device Software
	Setup stage		
1-5	Steps 1-5 are same as Control OUT transfer examples, except that software sets up Tx buffers and Start Transfer is issued to EP1 (endpoint 0, IN direction).		
	Data stage		
6	-	Device fetches the TRB and caches it. It prefetches the data and puts it in the TXFIFO. The core sends ERDY only in SS mode.	-
7	Host requests data packets	-	-
8	-	This step is similar to the step 4 of Non-isochronous IN transfers. The NumP for control transfer is 1. In addition, if another SETUP is received before the data transfer happens on USB, the core will skip the data transfer software has set up without waiting for it to happen on the USB bus and send XferComplete (when LST=1 TRB encountered) with the SetupPndg status bit set. Software has to reclaim the TRBs with HWO=1 in the skipped TRBs and flush the TXFIFO.	-
9	-	If the host starts the Status stage, the core returns NAK in non-SS mode or NRDY TP in SS mode.	On seeing the event: • If the Data stage is OK, then it sets up Control-Status-3 TRB (zero-bytes) and enables DMA by issuing Start Transfer to EP0 (endpoint 0, OUT direction). • If the Data stage is not OK, then software will set a STALL by issuing Set Stall command to EP0 (OUT direction).
	Status stage		
	Same as in Control OUT transfer examples.		

28.5.3.5 Stream handling in SuperSpeed

The SuperSpeed stream protocol consists of a negotiation between the host and device concerning the selection of a stream, followed by data movement on the selected stream. The core automatically handles the stream protocol by initiating stream selection and also taking into account streams initiated by the host.

NOTE

This section is not applicable for USB 2.0 mode.

28.5.3.5.1 Stream IDs and transfer resources

A stream ID is a 16-bit value which represents an individual flow of data between host and device. For stream-capable endpoints, the stream ID must be non-zero and is used in:

- The Command Parameter field in the DEPCMD register used when issuing a Start Transfer command.
- The Stream ID field in a TRB.
- The EventParam field in an endpoint-specific event (DEPEVT): XferComplete, XferInProgress, XferNotReady, and StreamEvt.

It is necessary to allocate 1 Transfer Resource per endpoint. The cost of each transfer resource is (32 + N_TRBS_PER_XFER x 16) bytes of memory in the descriptor cache. Core can cache maximum four TRBs per transfer.

After the Start Transfer command completes, the hardware returns the 7-bit Transfer Resource Index in the command complete event (and Device physical endpoint-n command register (DEPCMD_0 - DEPCMD_7)). This index is used only for the Update Transfer and End Transfer commands. In all other places, the Stream ID is used.

28.5.3.5.2 Stream selection and stream programming model

Referring to the USB 3.0 Device Stream Protocol State Machine, the Idle state is the state where a device does not have a current stream to work on. When the hardware detects that it is in the Idle state, it will attempt to initiate and move data on a stream by transmitting ERDY to the host.

In order for this to occur, the endpoint must be "primed" and the stream must be both active and ready.

- **Primed**: After a stream-capable endpoint is configured, it is in the disabled state, which prevents the device from initiating any stream selection. After the host performs a Prime transaction, the endpoint is moved into the "Primed" state where it is eligible to initiate stream selection.

- **Active**: Software makes a stream active by issuing a Start Transfer command for the stream. A stream becomes inactive after an XferComplete event or after software issues an End Transfer command.
- **Ready**: After a stream is added via the Start Transfer command, its default state is "ready." The hardware will label the stream as "not ready" if the host gives a NoStream rejection to the hardware's attempt to initiate the stream. If the host performs a Prime, all currently active streams are automatically set to the "ready" state by hardware.

In the Idle state, the hardware selects a stream from its cache to initiate. Once the core has chosen an active stream, it attempts to initiate and move data on that stream until the endpoint enters the Idle state, which occurs when the host rejects the stream selection (NoStream) or the stream is terminated (PP=0) by the host or device. Therefore, software is required to prepare enough HWO=1 TRBs for at least one packet prior to issuing the StartXfer command.

In device-oriented stream selection class drivers (such as UASP), the host will not reject the device's stream selection. However, other class drivers may be developed in which the host initiates stream selection. In this situation, software will not start any transfers and will wait for a XferNotReady event. When that event is received, software will look at the StreamID in the event and start the transfer associated with the given Stream ID. If for some reason the class driver allows the host to reject a stream that has already been initiated by the host or device, software will receive a Stream (NotFound) event. If this occurs, software may replace the existing stream by issuing End Transfer for it and then issuing Start Transfer for another stream.

Some host implementation scenarios exist where the host and device become out of sync in the stream protocol, creating a deadlock condition where the device waits for the host to issue a Prime transaction while the host waits for the device to issue an ERDY. To resolve potential deadlock conditions, software should perform the following:

1. Implement a time out for stalled transfers on stream-capable endpoints.
2. When the timeout elapses with no data transfer for any stream after a StreamEvent (NotFound) event, restart one of the streams by issuing an End Transfer command followed by a Start Transfer command.

This places the stream in the Ready state and causes the core to transmit an ERDY to the host.

28.5.3.5.3 Data movement within a stream

Data movement within a stream is the same as data movement using the Bulk endpoint type (Start Transfer, Update Transfer, and updating TRBs). The XferInProgress, XferComplete, and XferNotReady events are issued as data is transferred between host

and device. The Stream ID field in the events indicate which stream within the endpoint is generating the events. If software receives an XferNotReady event with the EventStatus field indicating XferNotActive, it means that the host is attempting to initiate a stream that has not been added to the hardware's cache through Start Transfer. In response to this event, software should add the requested stream if it is available.

28.5.3.6 Low power operation

28.5.3.6.1 Low power operation of USB

The USB 3.0 protocol allows a SuperSpeed host and device to negotiate with each other and decide when to bring their link into a lower power state (U1 or U2). This is accomplished by each side making individual decisions about whether they have any traffic currently pending or if they expect any traffic soon. The device requests low power entry by transmitting an LGO_U1 (or LGO_U2) Link Command and exits low power by using an LFPS handshake. The USB 3.0 Device Controller uses information from the host (bus side) and from software (application side) to decide when the best time to request low power entry is and when to exit from low power. The same calculation is used when the controller decides whether to accept a low power request from the host. The controller maintains state information about each endpoint and whether it is "active" or "paused". If all enabled endpoints are paused from either the application side or the bus side, low power entry is allowed. If at least one endpoint is not paused, low power entry is not allowed. To indicate that it does not have any pending traffic and allow the link to go into a low power state, the software may pause all enabled endpoints using one of the following mechanisms:

- Allow transfers to complete and not start any new ones
- Not setting the HWO bit in the TRBs to '1'

Exception: Software may prepare a Setup TRB for control endpoints without affecting the ability of the link going into low power. Otherwise, an endpoint is considered active on the application side. For these endpoints, the host indicates that it is paused in one of the following ways:

- Setting the Packet Pending (PP) flag to '0' in the DPH
- Setting the PP flag to '0' and NumP to '0' in the ACK TP
- For interrupt endpoints, by not polling the endpoint for two service intervals
- Note: After an endpoint is configured using the DEPCFG command and before the host sends any packets to it, the endpoint is considered to be paused on the bus side

Exception: If the data packet payload has a CRC error, the core responds with an ACK (Retry=1), and the endpoint is not paused. Otherwise, if PP=1 or Deferred=1, the endpoint is considered to be active on the bus side. Control endpoints have an additional condition as follows:

The core does not allow low power entry during a control transfer (starting from the reception of setup packet and ending at the completion of the status stage).If the core is configured for isochronous support, the following additional rules apply:

- The controller enters U0 at SystemExitLatency microseconds prior to the beginning of every microframe to receive the Isochronous Timestamp Packet (ITP).
- A PING causes every isochronous endpoint to wake up and be considered active, preventing low power entry.
- An isochronous endpoint is paused automatically when it transmits or receives the last packet of its interval (lpf=1) or if two intervals pass and the endpoint has not transmitted or received its last packet. In this way, after the PING, the core does not enter low power until all isochronous endpoints have moved their last packet or two intervals pass without another PING.

If all enabled endpoints are paused by software or the host, the core issues an LGO_U1 (or LGO_U2) link command or it responds with an LAU link command to the low power request of the host or hub. Otherwise, the core rejects the low power requests with an LXU link command.

Additionally,

- For bulk endpoints, the USB 3.0 specification says that the device may go to U1/U2 500ms (tERDYTimeout) after transmitting ERDY. It is the responsibility of the software to detect no activity on a bulk endpoint for 500ms or more and issue End Transfer if there is no activity. When the transfer is removed, the endpoint allows U1/U2 entry again.
- Because U1/U2 entry is based on the presence of packets in the TX FIFO and not based on the TRBs that are available to the core, it is possible that an IN endpoint responds with a NRDY, goes to U1, then U0, then transmits an ERDY if the timing and TXFIFO size works out such that the core cannot keep at least one packet in the TXFIFO at all times.
- In addition, before the final decision made to either initiate or accept low power state, device checks the state of DCTL[INITU2ENA][ACCEPTU2ENA][INITU1ENA][ACCEPTU1ENA]. If the device receives Set Link Function LMP with Force_LinkPM_Accept bit asserted, the device accepts low power request from its link partner independent of endpoints state and DCTL Power control settings.

28.5.3.6.2 Low power operation of core

When the link is in the U1, U2, U3, SS.Inactive, SS.Rxdetect, or SS.Disabled state (for USB 3.0) or the USB is in Sleep, Suspend, or V_{BUS}-off mode (for USB 2.0), the core can be put into a low power mode to save power. The core supports the following low power mode:

- **Clock-gating mode**: In this mode, the clocks connected to most of the core modules are gated off. Modules that still get clocks in the low power mode detect wakeup conditions and remove clock gating to other modules when a wakeup condition is detected. Clock-Gating mode can be used when the USB is in any low power state. Clock-Gating mode is enabled by setting the Disable Clock Gating bit in GCTL to 0.

28.5.3.6.2.1 Clock-Gating Mode

Clock-Gating mode is enabled if the Disable Clock Gating bit in the GCTL is set to 0. When the USB/link is in a low power state, the clocks to most of the core modules are gated off automatically by the core. When a wakeup condition is detected by the core, or when the software schedules new work to the core (in U1 or U2 states) or changes the USB/Link state, the core turns on clocks and the entire core goes into active mode.

28.5.4 Host programming model

Please refer xHCI specification for detailed information.

28.5.4.1 Initializing host registers

In order to initialize the core as a host, the application should perform the steps described in the xHCI specification. Global registers need to be re-initialized as described in Initializing global registers.

28.5.4.2 Host controller capability registers

For register definition, refer to Capability registers length and HC interface version number (CAPLENGTH) to Runtime register space offset (RTSOFF).

28.5.4.3 xHCI implementation details

This section discusses xHCI implementation details specific to the USB 3.0 controller.

28.5.4.3.1 LHCRST behavior

The xHCI Specification does not describe the programming model of light reset. It only specifies that the Operational and Runtime Registers that are not contained in the Aux Power well are at their default values.

Light reset must only be applied when USB_DCTL[Run_Stop] is equal to '0' to ensure that there is no discrepancy between hardware and software states. If light reset is applied during a transfer, the behavior on the USB is undefined, and may cause a packet to be terminated abruptly.

On light reset, the hardware resets all non-Aux registers to their default values which means that the port state (which is contained in PORTSC) does not change, but USB_DCTL[Run_Stop] is immediately set to '0' and all the context information about connected devices is lost. It is the responsibility of software to re-initialize the controller.

28.5.4.3.2 ENT requirements

In the xHCI Specification, the rules about when the ENT (Evaluate Next TRB) flag in a TRB must be set or cleared are not very strict. However, it states that the ENT flag must be set to '1' in the last Normal TRB when a TD ends with an Event Data TRB.

The controller requires software to follow these rules:

1. Regardless of whether the endpoint is stream-capable or not, if a TD ends with an Event Data TRB, the Normal TRB that precedes it must have ENT set to '1'. This Normal TRB may be separated from the Event Data TRB by a Link TRB
2. In all other cases, the ENT flag must be set to '0' Failure to follow these rules results in undefined behavior.

28.5.4.3.3 Behavior on babble error

If a babble error is detected and the received data passes all integrity checks, the host controller may write the received data (up to the expected data length) to the data buffer, and the value of the TRB transfer length field in the babble detected error transfer event shall be consistent with the number of data bytes written to the buffer.

The controller does not write the received data to the data buffer. The entire packet is discarded and the residual byte count is written to the TRB transfer length field.

28.5.4.3.4 Max_exit_latency_too_large message

If periodic transfers are on every microframe (Binterval 1), the controller reports max_exit_latency_too_large if PING scheduling is enabled. The xHCI scheduler must prefetch data from the system memory ahead of next microframe, manage non-periodic transfer, and schedule PING before ISOC. Because the system memory access and USB responses are not predictable Quality of Service (QoS) is not guaranteed if the xHCI host needs to perform all of the above for every microframe when U1/U2 is enabled. Therefore, the expectation is to disable U1/U2 if periodic data needs to be scheduled every microframe. In addition, because the U1 exit latency is in the order of 10usec, only very low system-level power saving is achieved even if the scheduler allows U1 transition by reducing QoS.

U1, U2, and U3 link states and exit time observed during lab testing:

- U1: Lowest power saving
  - In this state the chip is active, USB controller is inactive, only the Tx transmitter of the PHY is inactive.
  - U1 exit time per link partner: ~10 - 19 usec
- U2: Higher power saving than U1
  - In this state the chip is active, USB controller is inactive, both the Rx and Tx transceivers of the PHY are inactive.
  - U2 exit time per link partner: ~85 - 115 usec
- U3: Highest power saving
  - In this state, the chip and USB controller/PHY are inactive.
  - U3 exit time per link partner: 200 - 450 usec

Following are the recommendations for driver to handle periodic endpoints and "max_exit_latency_too_large" message:

- If Binterval of any endpoint is '1', then disable U1/U2 and do not schedule PING ("max_exit_latency" set to 0).
- If periodic scheduling is expected on every microframe, then disable U1/U2 and do not schedule PING.
- If periodic scheduling is expected only on every other microframe, then disable U2 and use U1 exit latency for "max_exit_latency" calculation.
- If ((number of hubs +1) * U2 exit latency) >= (2**(Binterval-1) * 125 usec), then disable U2.
- For all other periodic scheduling use U2 exit latency for "max_exit_latency" calculation.
- If the "max_exit_latency_too_large" error is sent to software, then disable U2 and re-issue ep- config with U1 exit latency.
- If "max_exit_latency_too_large" still happens, then disable U1 and re-issue ep_config with max_exit_latency set to 0 (no PING).

28.5.5 Device physical endpoint-specific commands

28.5.5.1 Command 1: Set endpoint configuration: DEPCFG

This command sets the physical endpoint configuration information.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 28-25. Command 1: Set endpoint configuration parameters: DEPCFG

Field	Description
Parameter 2	
31-0	Set to endpoint state when the Config Action field of Parameter 0 is 1. Otherwise, this is Reserved.
Parameter 1	
31	FIFO-based Set to '1' if this isochronous endpoint represents a FIFO-based data stream where TRBs have fixed values and are never written back by the core.
30	Bulk-based Set to '1' if this isochronous endpoint represents a bulk data stream that ignores the relationship of bus time to the intervals programmed in TRBs.
29-25	USB Endpoint Number bit[29:26]: Endpoint number bit[25]: Endpoint direction: 0 OUT 1 IN Physical endpoint 0 (EP0) must be allocated for control endpoint 0 OUT. Physical endpoint 1 (EP1) must be allocated for control endpoint 0 IN.
24	Stream Capable (StrmCap) Indicates this endpoint is stream-capable (MaxStreams != 0).
23-16	bInterval_m1 Set to the bInterval value minus 1. The valid values for this field are 0 through 13. The bInterval value is reported in the endpoint descriptor. When the core is operating in Full-Speed mode, this field must be set to 0.
15	Set to '1' if this bulk endpoint utilizes the External Buffer Control (EBC) mode. This limits the number of outstanding DMA transfers to one read and one write. This is a requirement for the EBC to operate properly. Using EBC without setting this bit to '1' results in undefined behavior.
14	Reserved

Table continues on the next page...

Table 28-25. Command 1: Set endpoint configuration parameters: DEPCFG (continued)

Field	Description
13-8	Bits 13-8 are used for Device Endpoint Specific Event Enable (DEPEVTEN). If an enable bit is set to 0, the corresponding event is not generated. Bit 13 Stream Event Enable (StreamEvtEn) Bit 12 Reserved Bit 11 Reserved Bit 10 XferNotReady Enable (XferNRdyEn) Bit 9 XferInProgress Enable (XferInProgEn) Bit 8 XferComplete Enable (XferCmplEn)
7-5	Reserved
4:0	Interrupt number (IntrNum) Applies to IN and OUT endpoints. Indicates interrupt/Event Buffer number on which endpoint related interrupts for this endpoint are generated. This must be set to 0.
Parameter 0	
31-30	Config Action 0 Initialize endpoint state: This action is used when an endpoint is configured the first time. It will cause the data sequence number and flow control state to be reset. DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set "Ignore Sequence Number" to 0. 1 Reserved. 2 Modify endpoint state: This action is used when modifying an existing endpoint configuration, such as changing the DEPEVTEN event enable bits, interrupt number, or MaxPacketSize. The data sequence number and flow control state will be unchanged, and DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set "Ignore Sequence Number" to 1.
29-26	Reserved
25-22	Burst Size (BrstSiz) When field is set to- 0 Burst length = 1 1 Burst length = 2, and so on, up to, 15 Burst length = 16 For IN transfers, this value represents the maximum length of the burst that the device attempts when the Host initiates an IN transfer with a TP_ACK. If BrstSiz < the NumP value in the initiating TP_ACK, then the device controller limits the burst length to BrstSiz. If BrstSiz >= the NumP value in the initiating TP_ACK, then the device controller attempts a burst length of NumP. For OUT transfers, this value represents the NumP value utilized in the response TP_ACK from the device controller. The NumP value indicates (to the host) the burst length the device desires. Note: In the special case of BrstSiz=0 (burst length = 1), the TP_ACK from the device controller has NumP=0; which is a flow-control condition. If this value is utilized, then the endpoint enters flow

Table continues on the next page...

Table 28-25. Command 1: Set endpoint configuration parameters: DEPCFG (continued)

Field	Description
	control for each DP received and a subsequent ERDY is transmitted (according to the USB Specification). However, this may have an undesirable impact on the system throughput. To avoid this impact, it is recommended to set BrstSize=1 to prevent the flow-control condition. The trade-off is that the host could potentially burst two OUT data packets to the device, resulting in an ACK for the first packet, and an NRDY for the second packet.
21-17	FIFO Number (FIFONum) Indicates which transmit FIFO is assigned to this endpoint. For control endpoints, the FIFONum value in the OUT direction must be programmed to the same value as the IN direction. This field should be set to 0 for all other OUT endpoints. Even though there may be more than 16 TXFIFOs in DRD mode, the device mode must use lower 16 TXFIFOs.
16-14	Reserved
13-3	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size (in bytes) for the current USB endpoint. USB 3.0 supports up to 1024 bytes.
2-1	Endpoint Type (EPType) This is the transfer type supported by this USB endpoint. 00 Control 01 Isochronous 10 Bulk 11 Interrupt
0	Reserved

28.5.5.2 Command 2: Set endpoint transfer resource configuration (DEPXFERCFG)

There must be only one transfer resource allocated per endpoint. Start transfer causes the use of the transfer resource. End Transfer or an XferComplete event releases the transfer resource. If software attempts to allocate more transfer resources than have been configured in the hardware, this command will return an error in the CmdStatus/ EventStatus field.

NOTE

If GUSB2PHYCFG[SUSPENDUSB20] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 28-26. Command 2: Set endpoint transfer resource configuration parameters: DEPXFERCFG

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Reserved
Parameter 0	
31-16	Reserved
15-0	Number of Transfer Resources (NumXferRes) Defines the number of Transfer Resources allocated to this endpoint. This field must be set to 1.

28.5.5.3 Command 3: Get endpoint state (DEPGETSTATE)

This command is not used.

28.5.5.4 Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL)

These commands apply to non-isochronous IN and OUT endpoints only.

The application issues a Set Stall command to stall all tokens from the USB host to this endpoint. If the endpoint is in the NRDY state, the STALL state takes priority.

If a transaction is currently in progress when the software issues Set Stall, the behavior depends on the direction of the endpoint:

- For OUT endpoints, the current packet will complete. The core will respond with STALL to the next OUT DP or token.
- For IN endpoints in SuperSpeed, the current burst transaction will complete, and the core will respond with STALL to the next ACK TP. For other speeds, the core will complete the current packet and respond STALL to the next IN token.

For non-control endpoints, the application is responsible for both setting and clearing STALL via the Set Stall/Clear Stall commands. When the application clears the STALL, the endpoint's data sequence number is reset to zero.

For control endpoints, the application issues only the Set Stall command, and only on the OUT direction of the control endpoint. The controller automatically clears the STALL when it receives a SETUP token for the endpoint.

The application must not issue the Clear Stall command on a control endpoint. If the endpoint is in flow control (NRDY) and also in the STALL state, it responds with a STALL for any packet. The only exception is that the controller always responds to SETUP data packets with an ACK handshake, independent of the STALL state.

NOTE

- If GUSB2PHYCFG[6] is '1', it must be set to '0' prior to issuing this command and may be set to ‘1’ after the command completes.
- When issuing Clear Stall command for IN endpoints in SuperSpeed mode, the software must set the "ClearPendIN" bit to '1' to clear any pending IN transcations, so that the device does not expect any ACK TP from the host for the data sent earlier.

28.5.5.5 Command 6: Start transfer (DEPSTRTXFER)

This command applies to IN and OUT endpoints.

A Transfer Descriptor (TD) in Device mode is a list of TRBs that comprises one or more transfers. The pointer to a TD is added to the core's cache by the Start Transfer command and removed by an XferComplete event or by the End Transfer command.

Software issues this command indicating that a descriptor is ready to be processed and DMA can start for this endpoint/stream combination. For IN endpoints, this causes the descriptor to be processed, and data is moved from the corresponding memory buffer to corresponding transmit FIFO when the transfer is started. For OUT endpoints, this causes the descriptor to be processed and data is moved from the receive FIFO to corresponding memory buffer.

In response to the Start Transfer command, the hardware assigns this transfer a resource index number (XferRscIdx) and returns the index in the DEPCMDn register and in the Command Complete event. This index must be used in subsequent Update and End Transfer commands.

It is illegal to issue a Start Transfer command for the same TD if it is already present in the core's cache.

Non-stream capable endpoints rules:

- The Stream ID field must be set to 0.
- The HighPri field is reserved.

Stream-capable endpoint rules:

- The Stream ID field must be set to non-zero and match the Stream ID passed into the Start Transfer endpoint command associated with this transfer. In all the TRBs of a transfer, the Stream ID fields must be the same.
- The HighPri field is reserved.

NOTE

- If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.
- Before issuing a start transfer command, software needs to verify whether the link is in U0. If the link is not in U0, software needs to bring the link to U0 by performing a remote wakeup. This is applicable to both SS and USB 2.0 speeds.
- When an IN transfer is in progress, (that is, started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to determine if the TXFIFO is empty or not. If the TXFIFO is not empty, it can initiate a remote wakeup.

Table 28-27. Command 6: Start transfer parameters: DEPSTRTXFER

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Transfer Descriptor Address (TDAddr Low) Indicates the lower 32 bits of the external memory's start address for the transfer descriptor. Because TRBs must be aligned to a 16-byte boundary, the lower 4 bits of this address must be 0.
Parameter 0	
31-0	Transfer Descriptor Address (TDAddr High) Indicates the higher 32 bits of the external memory's start address for the transfer descriptor.

28.5.5.6 Command 7: Update transfer (DEPUPDXFER)

If software uses circular TRB buffers and updates a TRB, whose Hardware Owner (HWO) bit was 0, by setting HWO=1, it must execute the Update Transfer command, specifying the transfer resource index of the TRB in the DEPCMD register. The hardware uses this information to re-cache the TRB.

Software may issue a special “No Response Update Transfer” command by setting CmdAct=0 and CmdIOC=0. In this case, the hardware does not generate a Command Complete event, does not set the CmdAct bit to '0' (because it will be '0'), and software may immediately issue another command to the same endpoint following this one. This special type of Update Transfer may not be used when software depends on the XferNotReady event to setup TRBs (see “On Demand” transfers in Transfer setup recommendations).

Software may issue an Update Transfer command for a transfer resource that has already completed (either due to XferComplete event or an End Transfer command), and the core will detect that the Update Transfer is unnecessary. However, software must not issue an Update Transfer command for a transfer resource index that has never been started.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

28.5.5.7 Command 8: End transfer (DEPENDXFER)

This command applies to IN and OUT endpoints. Software issues this command requesting DMA to stop for the endpoint/stream specifying the transfer resource index of the TRB and the ForceRM parameter to be set to 1 in the DEPCMD register.

When issuing an End Transfer command, software must set the CmdIOC bit (field 8) so that an Endpoint Command Complete event is generated after the transfer ends. This is necessary to synchronize the conclusion of system bus traffic before the End Transfer command is completed.

For IN endpoints, this command causes descriptor processing to stop and the core stops fetching new data for the endpoint and stops transfers on the USB. The core may truncate a packet being transmitted with the DPPABORT ordered set, does not wait for any pending ACKs from the USB, and does not update the TRB status. For OUT endpoints, this command causes descriptor processing to stop. If there is currently data, it is moved from the receive FIFO to corresponding memory buffer and completed at packet boundary. The core does not update the TRB status.

Use this command under the following conditions:

- When handling USBReset or SetConfiguration, endpoints are closed using this command.
- After receiving a ClearFeature (STALL) control transfer, software issues End Transfer followed by Clear Stall, followed by Start Transfer.

- After an XferInProgress event when the TRB after the one that caused the XferInProgress event has its HWO bit set to '0', this command can be used.
- For isochronous endpoints, if the host stops moving data for many intervals, software may force the end of the transfer and wait for the host to restart.

The hardware does not issue a XferComplete event on End Transfer, but only issues a CommandComplete event.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 28-28. Command 8: End Transfer Parameters: DEPENDXFER

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Reserved
Parameter 0	
31-0	Reserved

28.5.5.8 Command 9: Start new configuration (DEPSTARTCFG)

Software issues this command under the following conditions:

- After power-on-reset with XferRscIdx=0 before starting to configure Endpoints 0 and 1. CmdIOC must be set to '0' and software must poll the CmdAct bit to determine when the command is complete because Endpoint 0 is not yet configured with a valid interrupt number.
- With XferRscIdx=2 when it receives SetConfiguration before starting to configure Endpoints > 1. CmdIOC may be set to '0' or '1'.

This command should always be issued to Endpoint 0 (DEPCMD0).

Hardware resets the transfer resource allocation to the value in the XferRscIdx parameter (must be 0 or 2) upon receiving this command.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

28.5.6 OTG

This section describes OTG for USB 3.0, and the programming requirements for the USB 3.0 core in OTG mode.

28.5.6.1 OTG 2.0 for USB 3.0 Functionality

The following sections describe the OTG 2.0 functions for USB 3.0.

28.5.6.1.1 Core OTG functions

The OTG Interface block within the U2MAC handles the core OTG functions: Session Request Protocol (SRP) and Host Negotiation Protocol (HNP). These OTG protocols are implemented through the regular UTMI+ OTG interface.

The MAC handles HNP for host and peripheral role swapping.

The MAC also handles SRP, which allows an A-Device to turn off VBUS to save power when the USB is not used, and provides a means for a B-Device to monitor VBUS and request the A-Device to activate VBUS.

It includes the following logic necessary to achieve SRP and HNP:

- Control of VBUS through Vbus drive enable (utmiotg_drvvbus) as A-Device
- Control of IDDIG line sampling enable control output (utmiotg_idpullup)
- Control of D+/D- pull-down resistor enables (utmiotg_dppulldown/ utmiotg_dmpulldown)
- Generates SRP (data line pulsing) as B-Device when the session is off.

NOTE

- When the application programs the USB_GCTL[PRTCAPDIR] as 2'b11 to enable OTG mode, then in the A- Device mode, the core will only enumerate as a HS device and hence will not support any SS-capable device to be connected. This restriction is not valid when the USB_GCTL[PRTCAPDIR] is programmed to 2'b01 or 2'b10.
- The utmisrp_chrgvbus and utmisrp_dischrgvbus outputs are provided for legacy PHY connections but they are both

set to 1'b0 from the core since VBUS charging is not supported in OTG 2.0 specifications.
- The application should not program GCTL[0] as 1'b0 when it wants to do a ADP or HNP. In such a case, the application should disable the Clock Gating feature by programming GCTL[0] as 1'b1. The application should re-enable the Clock Gating feature only when the core returns to its original role of operation.

28.5.6.1.1.1 HNP Polling and Enable

This HNP Polling activity involves an OTG device acting as the current host to periodically poll the remote device to check if the remote device wishes to take the host role. It will then grant the role swap opportunity at the earliest opportunity. Being a software activity, HNP Polling is expected to be implemented through software timers and periodic exchange of SetFeature.SetHNPEnable packets.

The core is then informed of successful exchange of these packets to Enable HNP activity within the core.

28.5.6.1.2 ADP functions

The ADP functions involve the following two processes:

- ADP sensing
- ADP probing

ADP probing capability is required in both A and B devices, while sensing is a must only for B-Device. The main functional unit is the ADP controller.

28.5.6.1.2.1 Internal ADP controller

- All ADP timers are maintained in pwrm (Power) module and generate the enable and disable signaling to PHY for ADP probing and sensing.
- PHY contains the following circuitry related to ADP functionality:
  - Comparators for PRB and SNS
  - I_ADP_SRC and I_ADP_SNK
  - Vbus circuitry
- Application software programs the ADP timing registers residing in the pwrm module.
- pwrm module provides a mechanism to application through the interrupts to log and report events pertaining to ADP probing and sensing.

- Only pwrm module needs to be always powered on in this option. USB 3.0 controller may or may not be powered on.
- As a product, both the ADP controller logic and OTG controller are packaged into USB3 core. In the USB3 core, ADP sensing and probing is loosely coupled with the rest of the logic and is directly under software control. This achieves maximum flexibility to interact with OTG core functions.

The ADP controller in the block diagram is a fully digital controller. This has mainly timers inside it to help ADP operation. All ADP related timers are part of pwrm module.

Figure 28-20. Internal ADP controller

For the PHY to generate ADPPRB and ADPSNS, the PHY needs to know the value to compare with VADP_PRB and VADP_SNS respectively. These values can be hardcoded in the PHY or the PHY can have a limited programmable option by which these reference voltage values can be changed. It is assumed that the PHY takes care of this. The pwrm module also implements SRP detection in A-Device mode. This is required if pwrdwn module of the USB 3.0 controller is powered down completely while ADP is in progress and B-Device initiates SRP.

NOTE

- The ADP block works with the suspend clock.
- The ADP timers run with the suspend clock along with the programmed USB_GCTL[PWRDNSCALE] value. The USB_GCTL[PWRDNSCALE] value is used internally to generate 32 KHz pulse for the timers.

28.5.6.1.3 Software flow

The software flow diagram is illustrated in the following figure.

Figure 28-21. Software flow diagram

The following two sections describe the A-device and B-device flows briefly with respect to the above flow diagram.

NOTE

The following concise flows are not very detailed, and are provided here to help understand at an abstract level.

28.5.6.1.3.1 A-device activity concise flow

- Power-on-reset
- Only the OTG driver is active after power-on-reset. The host and device drivers are inactive after power-on-reset since at this point of time it is unsure what role the device assumes.
- The OTG driver initializes ADP and waits until a successful ADP event is received.
- The OTG driver senses ID pin = 0 and initializes the OTG registers.
- The OTG driver waits until a successful SRP event is received and turns on the VBUS.

- The OTG driver requests the USBD (USB driver) to bring up the xHC.
- After successful xHC bring up, device enumeration (and data transfers) occurs during which the device capabilities are exchanged for determining a possible role switch.
- In the event where a role switch is possible, the OTG application software or the B-device may initiate a role switch (The OTG application software may request via the Host application software to create favorable conditions for a role switch).
- The xHC is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the PCD (Peripheral driver) to bring up the Peripheral.
- After successful Peripheral bring-up, device enumeration (and data transfers) occurs.
- In an event where a role switch is possible, OTG application software may initiate a role switch (The OTG application software may request via the Device application software for a role switch).
- The PCD is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the USBD to bring-up the xHC, thus reverting to the original roles.

28.5.6.1.3.2 B-device activity concise flow

- Power-on-reset
- Only the OTG driver is active after power-on-reset. The host and device drivers are inactive after power-on-reset since at this point of time it is unsure what role the device assumes.
- The OTG driver initializes ADP and waits until a successful ADP event is received.
- The OTG driver senses ID pin = 1 and initializes the OTG registers.
- The OTG driver initiates a SRP and waits for the VBUS to be turned on by the A-device.
- The OTG driver requests the PCD to bring up the peripheral.
- After successful peripheral bring up, device enumeration (and data transfers) occurs during which the device capabilities are exchanged for determining a possible role switch.
- In an event where a role switch is possible, the OTG application software or the A-device may initiate a role switch (The OTG application software may request through the device application software for a role switch).
- The PCD is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the USBD to bring up the xHC.
- After successful xHC bring-up, device enumeration (and data transfers) occurs.
- In an event where a role switch is possible, the OTG application software or A-device may initiate a role switch (The OTG application software may request the host application software for a role switch).
- The xHC is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the PCD to bring-up the Peripheral, thus reverting to the original roles.

28.5.6.1.4 Programming model

This section describes the programming requirements for the USB 3.0 core in OTG mode. The core can be configured either as a DRD-device or an OTG, based on bit [13:12] of the Global Control Register's (GCTL) Power-On Initialization Value. If power-on configuration option is chosen as DRD-device, then PrtCapDir has to be explicitly programmed to 2'b11 for operating as an OTG device. The premise for the programming model is as follows.

1. The OTG driver is expected to run concurrently and independently with the PCD and xHCI drivers.
2. The OTG driver doesn't get involved in the actual data transfers but is responsible for communicating messages between the xHCI driver and PCD during role changes.
3. The OTG driver is responsible for handling the HNP, SRP and ADP (internal) events from the core. After power on, the OTG driver is responsible for loading, starting and switching between the xHCI driver and PCD. When started, the xHCI driver initializes the xHCI register set before enumerating the connected device. Similarly, when started, the PCD initializes the device register set and waits for the USB reset event to continue. Once the connection is established between the xHCI driver or the PCD by the OTG driver, the corresponding driver takes over for data transfers.

After every successful HNP switching, it is necessary that the active driver is changed from the xHCI driver to PCD or vice-versa. In such cases, the application may also unload the active driver from the memory and re-use the same memory area for loading the next active driver. The subsequent flow described in this section assumes that the core can get enumerated in any of the speeds (HS/FS) and bases its discussion accordingly. For example, if the core enumerates in HS/FS, follow the OTG 2.0 flow.

28.5.6.1.4.1 Initializing global registers

The global registers of the USB 3.0 core are shared between the device and host modes. It is the responsibility of the individual driver to initialize the core's global register specific to its functions. This is not explicitly mentioned in the flows described in this chapter. This specifically includes initializing the GTXTHRCFG/GRXTHRCFG, GEVNTEN, and GPRTBIMAP registers, and the Global FIFO Size and Global Event Buffer Registers. For more details, refer to Global registers in Table 28-2. However, these global registers that are programmed once at power on can be initialized in the global initialization routine in the OTG driver. Examples of one-time programmable registers are GSBUSCFG, GCTL, GSNPSID, and GUCTL. Examples of fields that are one-time programmable are the PHY interface (UTMI), AXI parameters like burst size, and so on.

The standard xHCI host driver does not get involved in the core-specific global register programming, and therefore, needs to be handled by the Board Support Package software.

28.5.6.1.4.2 Initializing host registers

In order to initialize the core as a host, the application should perform the steps described in the xHCI specification. Global registers need to be re-initialized as described in Initializing global registers.

28.5.6.1.4.3 Initializing device registers

In order to initialize the core as a device, the application should perform the steps described in the section Device Programming Model. Global registers need to be re-initialized as described in Initializing global registers. For specific registers, refer to Device Programming Model.

28.5.6.1.4.4 Initializing OTG registers

The application should initialize the OTG registers based on the initial value of the OSTS.ConIDSts after power on. The following sections depict the programming flow for the OTG application in detail.

28.5.6.1.4.5 Programming flow for OTG in USB 3.0

The following figure shows the programming flow after power-on reset.

Figure 28-22. OTG driver overall programming flow

Following steps describe overall programming flow:

1. During global initialization, read the GHWPARAMS6 register to see if SRP support is enabled in the configured core. If SRP support is enabled, then proceed with the OTG programming flow by programming USB_GCTL[PRTCAPDIR] as 2'b11. If SRP support is not enabled, do not proceed further in the OTG programming flow. The configured device must at least support SRP for cable connection-based (Connector ID) role of operation.

2. Initialize the ADP controller as explained in ADP programming flow.
3. The OTG 2.0 state machine is initially in B-IDLE if IDDIG is 1, or A-IDLE if IDDIG is 0. Enable USB_OEVTEN[ConIDStsChngEvntEn] for any change in the Connector ID status.
4. Initialize the OTG control and configuration register to default values.
5. Read the OSTS register to find out the current status of the Connector ID (ConIDSts). After power-on or soft reset, the ConIDSts will be valid only after the PHY delay from IDPULL=1 to IDDIG active and any filter delay for IDDIG inside or outside the USB 3.0 core.
6. If USB_OSTS[ConIDSts] is 1, the OTG 2.0 state machine is in B-IDLE, follow the B-Device flow. Otherwise, if OTG 2.0 state machine is in A-IDLE, follow the A-Device flow.
7. When there is any connector ID (IDDIG) change resulting in USB_OEVT[ConIDStsChngEvnt], then exit the A-Device/B-Device flow. Stop the active driver and re-initialize the OTG control and status registers.

28.5.6.1.5 Common driver tasks

The common tasks for both A-Device and B-Device flow are as follows:

- Start xHCI driver
- Start peripheral control driver (PCD)
- Switch peripheral
- Switch host
- Stop active driver
- Enable HNP change

The Stop Active Driver is already introduced in the common flow diagram. Other tasks are useful for the A-Device/B-Device flows.

Start xHCI driver

The OTG driver, being responsible for starting the xHCI driver, executes this task. The OTG driver loads the xHCI driver after initializing the global registers, and the xHCI driver is responsible for all the functions thereafter.

Figure 28-23. Start xHCI driver task

Start peripheral control driver

The OTG driver, being responsible for starting the Peripheral Control Driver (PCD), executes this task. The OTG driver only loads the PCD, which is responsible for all the functions thereafter, including the global registers initialization specific to the USB 3.0 core. Note that except the GTXFIFOSIZ and GEVNTADR, none of the other global registers need to be potentially re-programmed with different values.

Figure 28-24. Start peripheral driver task

Stop active driver

This task checks which driver is currently active and stops the active driver. Optionally, it may unload the driver if the OS supports dynamic loading of drivers.

Figure 28-25. Stop active driver task

Switch peripheral and switch host

These tasks are used during HNP and each consists of two individual tasks.

Figure 28-26. Switch peripheral task and switch host task

Enable HNP change

For HS/FS, this task is called when SetFeature.SetHNP is successfully exchanged between the host and the device. The set Peri_Run_Stop_Reset indicates to the device driver to automatically disable all events and perform a peripheral-initiated disconnect when the host puts the device in Suspend. Program the OCTL.DevSetHNPEn as 1, and OCTL.DevSetHNPReq as 1 immediately after the successful completion of SetFeature.SetHNP on the USB. This ensures the HNP attempt from the OTG core in the following Suspend state.

NOTE

By default, at power-on, program GUSB2PHYCFG.SusPHY as 1 to ensure that the PHY enters low power mode during normal suspend/resume without role switch.

Figure 28-27. Enable HNP change

28.5.6.1.6 A-device flow

The following figure shows the A-device flow.

Figure 28-28. A-Device flow diagram

A-device flow:

1. At Start, the A-device core will be in A_IDLE VBUS off state.
2. The OTG software should program the OTG control and configuration registers for A-device operation:
   - OCFG[HNPCAP] = 0
   - OCFG[SRPCAP] = 0
   - OCTL[PERIMODE] = 0
   - OCTL[TERMSELDLPULSE] = 0
   - OCTL[DEVHNPEN] = 0
   - OCTL[HSTSETHNPEN] = 0
   - OCFG[DISPRTPWRCUTOFF] = 0 or 1, based on whether the application requires the core disable the feature to automatically switch off VBUS after 1 second if port is disconnected.
3. Start the xHCI driver.
4. The OTG software should wait for OSTS[HSTRUNSTP] to become 1 to continue. If OSTS[HSTRUNSTP] is 0, the OTG Software should wait for a software event that indicates the change in the xHCI RUN_STOP bit to continue. Alternatively, the OTG Software can poll for OSTS[HSTRUNSTP] bit to be set.
5. OTG Software should program the OTG configuration and event registers to enable A-device events to detect peripheral connect and session end. Refer to A-device flow diagram.
6. Program GUSB2PHYCFG0[SUSPHY] = 1 to enable the core to switch off the PHY clock in the possible non-HNP suspend scenarios.
7. Set OCTL[PRTPWRCTL] when OSTS[XHCIPRTPOWER] is 1. This enables driving of VBUS to the B-device.
8. The A-device may enumerate in HS/FS based on the device signaling.
9. The OTG 2.0 state machine enters a3_ds_host/A-host for HS (FS) if there is a successful B-device connect (ADEVHOSTEVNT).
10. The OTG software should enable the ADEVHNPCHNG and ADEVBHOSTEND events to detect the transition from A-host to A-peripheral and the end of A-peripheral events.
11. If there is an ADEVSESSENDDETEVNT interrupt indicating that the core has now stopped driving the VBUS, the software should wait for ADEVIDLEDETEVNT interrupt and then disable all the OTG events.
12. The Software can enable the ADP flow.
13. After an ADEVHOSTEVNT interrupt, the core is in session. The flow for the A-device in session is shown in earlier in this chapter.

NOTE

- At any step, if CONIDSTSCHNGEVNT occurs, A-device flow is terminated as per the figure "OTG Driver Overall Programming Flow".
- Over current is not detected by the OTG state machine. The xHCI controller detects it, and the OTG driver in turn sees an ADEVSESSENDDETEVNT (from PORTSC[PP] = 0) due to an overcurrent.

OTG state machine mapping to the software flow

- Steps 1-5 are A-IDLE.
- Step 7 is A_WAIT_VRISE in which A-device has detected SRP and therefore waits for VBUS to stabilize.
- Step 8 and step 10 are primarily A_HOST through A_WAIT_VRISE and A_WAIT_BCON. If there is an error in either A_WAIT_VRISE or A_WAIT_BCON, ADEVSESSENDDETEVNT occurs making the state-machine fall back to A-IDLE.

Handling over-current in OTG mode of operation

- When the core is operating in USB 2.0 mode, an over-current condition causes PORTSC[OCA] = 1 and PORTSC[OCC] = 1 on the corresponding USB 2.0 port number. The core also updates a port status change event TRB for the corresponding USB 2.0 port number.
- When the core is operating in USB 3.0 mode, an over-current condition causes PORTSC[OCA] = 1 and PORTSC[OCC] = 1 on the corresponding USB 2.0 and USB 3.0 port numbers. The core also updates a port status change event TRB, one each for the corresponding USB 2.0 port number and the corresponding USB 3.0 port number.
- Note that PORTSC[OCA] may transition quickly from 0 to 1 and 1 to 0 before xHCI driver can see two distinct interrupts. This may occur due to over-current condition being removed quickly. In such a case, the xHCI driver will receive a single interrupt (PORTSC[OCC]) for over-current.

Figure 28-29. A-host -> A-peripheral -> A-host flow diagram

For HS/FS mode:

1. When the core starts the session with A-device as host, the application may periodically exchange SetFeature.SetHNPEnable packets. If B-device wants to become the host, this packet exchange results in the host application knowing about B-Device's intent to become the host, and it directs the OTG driver to set OCTL[HSTSETHNPEN]. This enables the core to cater for HNP from B-device when suspend is initiated. During suspend, ADEVHNPCHNGEVNT is set.
2. If ADEVHNPCHNGEVNT is set, the OTG driver reads OEVT[HSTNEGSTS]. If OEVT[HSTNEGSTS] is set, then the core transitions to A-Peripheral state by executing Switch Peripheral task. Otherwise, wait for ADEVSESSENDDETEVNT and go to ADP probing.
3. When in A-peripheral state, if there is ADEVBHOSTENDEVNT, it signifies that the B-device no longer wants to be the host. Therefore, the driver goes back to Step 1 by executing the switch host task and assumes an A-host role.

   Alternatively, if there is an xHCI[PRTPOWER] de-assertion, the state machine goes to start ADP probing due to the occurrence of ADEVSESSENDDETEVNT.

4. If ADEVBHOSTENDEVNT is set, then the core would be in A-host state by executing switch host task. Otherwise, perform either of the following:
   a. Drop VBUS and go to ADP probing.

      If the role change is successful, the OTG driver hands over the control to the xHCI driver.

Internal ADP flow for A-device

The internal ADP flow of A-device is as follows:

Figure 28-30. Internal ADP flow for A-device

- ADEVSESSENDDETEVNT occurs due to:
  - PORTSC[PRTPOWER] = 1'b0
  - OTG driver stopping VBUS due to:
    - BCON_TOUT when in A_WAIT_BCON
    - A_AIDL_BDIS_TOUT when in A_SUSPEND
- At any step, if ADEVSESSENDDETEVNT occurs, the software starts with ADP probing.
- Once the ADP probing (both external and internal) is started, the software should wait for one of the following events to occur (to stop ADP probing and proceed to Step 2 in A-device flow) to continue.
  - ADP change event
  - SRP detect event
  - New session event from A-device application
  - xHCI port power change event from A-device application

28.5.6.1.7 SRP detection by the core (Timeline for ADevSRPDetEvnt)

ADevSRPDetEvnt is triggered by a valid DLINE pulsing on the USB D+ line by B-Device. The A-Device looks at both the positive and negative edge transition occurrence for a valid data-line pulse time before detecting SRP.

Figure 28-31. Timeline for ADevSRPDetEvnt

28.5.6.1.8 VBUS turned ON by the core (Timeline for ADevBSessEndEvnt)

When the A-device turns on the VBUS, it is unsure at this point of time at what speed the USB enumerates to. So, both the LTSSM and the OTG 2.0 FSMs start concurrently. The LTSSM proceeds with the training sequence after Rx_detect and the OTG 2.0 FSM waits for A_WAIT_BCON.

HS/FS Mode: Before entering A-Host, there is a check done in the OTG state machine to ensure B-Device is connected. If B-Device is not connected within TA_WAIT_BCON, ADevBSessEndDetEvnt is set. The core resets OCTL.PrtPwrCtl, and the OTG state machine enters A_WAIT_VFALL. Then VBUS to B-Device is removed and this results in ADevSessEndEvnt. Note that the A_WAIT_VFALL transition can occur due to over current, which is not depicted in the following figure.

Figure 28-32. Timeline for ADevBSessEndEvnt

28.5.6.1.9 Core entering A-Host in HS/FS mode (Timeline for ADevBHostEvnt)

HS/FS Mode: If the B-Device is connected within TA_WAIT_BCON, ADevBHostEvnt is set.

Figure 28-33. Timeline for ADevBHostEvnt

28.5.6.1.10 B-device flow

The following figure shows the B-device flow.

Figure 28-34. B-device flow diagram

1. At start, the core is in B-IDLE state. Note that for OTG functionality in peripheral mode, DCTL[RUN_STOP] should be set, otherwise the core as B-peripheral does not initiate SRP pulsing. Also, it is important for the device driver to program the speed of operation as B-peripheral (DCFG[SPD]).
2. The OTG software should program the following OTG control registers for B-device operation:
   - OCFG[HNPCAP] = GHWPARAMS6[HNPSUPPORT]
   - OCFG[SRPCAP] = 1
   - OCTL[PERIMODE] = 1
   - OCTL[TERMSELDLPULSE] = 0
   - OCTL[HNPREQ] = 0
   - OCTL[DEVHNPEN] = 0
   - OCTL[HSTSETHNPEN] = 0
   - OEVTEN[OTGBDEVSESVLDEVNTEN] = 1
   - GUSB2PHYCFG0[SUSPHY] = 1
3. Start peripheral driver.
4. Program OCTL[SESREQ] = 1 to initiate SRP by D-line pulsing. The core initiates SRP and waits for a valid VBUS (BVALID).
5. The B-device core continues to wait for BVALID if the A-device does not respond. The application should timeout if SESSVLDDETEVNT does not come within the SRP fail time (TB_SRP_FAIL is 6 seconds).
6. When SRP fails, the application should enable ADP probing.
7. If the A-device asserts a valid VBUS in response to SRP, the core reports an BDEVSESSVLDDETEVNT to indicate the start of a session.
8. If there is a BDEVSESSVLDDETEVNT, the core enters a session on state. For HNP flows in B-peripheral -> B-host -> B-peripheral flow diagram.
9. If there is a BDEVVBUSCHNGEVNT anytime, indicating VBUS is no longer valid, the core enters session-end state.

NOTE

- For OTG 2.0, during HNP process where the B-device is going to assume the role of a host, the B-device application needs to ensure that a USB reset process is programmed within 150 ms (TB_ACON_BSE0) of getting a device connect interrupt.
- At any step, if CONIDSTSCHNGEVNT occurs, the A-device flow is terminated as per "A-device Flow Diagram".

The OTG state machine mapping to the software flow is as follows:

- Steps 1-3 and 6 are B-IDLE
- Step 4 is B-SRP-INIT
- Step 7 and 8 are B-PERIPHERAL

Figure 28-35. B-peripheral -> B-host -> B-peripheral flow diagram

For HS/FS mode:

1. The OTG state machine starts operation as a B-peripheral. The following events are enabled.
   - OEVTEN[OTGBDEVHNPCHNGEVNTEN] = 1
   - OEVTEN[OTGBDEVVBUSCHNGEVNTEN] = 1
   - OEVTEN[OTGBDEVBHOSTENDEVNTEN] = 1

   In this state, the application periodically exchanges SetFeature.SetHNPEnable packets. If the B-device wants to become host, this packet exchange results in a device application sending an ACK to this packet, and it directs the OTG driver to execute Enable HNP change. Ensure that GUSB2PHYCFG0[SUSPHY] is 1'b0 when SetFeature.SetHNP command succeeds. This enables the core to initiate HNP from B-device when a suspend is initiated. During a suspend, BDEVHNPCHNGEVNT is set. OTGBDEVHNPCHNGEVNT starts the transition from B-peripheral ->B-WAIT-ACON to either B-host or B-peripheral.

2. If BDEVHNPCHNGEVNT is set, the OTG driver reads OEVT[HSTNEGSTS]. If OEVT[HSTNEGSTS] is set, it indicates successful connection of A-device as

peripheral, the core therefore transitions to B-host state. After entering the host role, the host application must drive the port reset within 150 ms. If OEVT[HSTNEGSTS] is not set, it indicates failure of HNP and it stays as B-peripheral.

3. When in B-host state, if there is BDEVBHOSTENDEVNT, it signifies that the B-device no longer wants to be the host, therefore suspended the bus and A-device has stopped signaling connect. Therefore, the driver goes back to step 1 and assumes B-peripheral role.
4. The core generates BDEVBHOSTENDEVNT.

NOTE

- Step 3 above is B-host and B-host to B-peripheral.
- OTGBDEVVBUSCHNGEVNT indicates transition back to B-IDLE.

NOTE

- Step 3 above is B-host and B-host to B-peripheral.
- OTGBDEVVBUSCHNGEVNT indicates transition back to B-IDLE.
- If there is any unexpected error scenario like VBUS drop after the OCTL[DEVSETHNPEN] bit is set, the OTG driver must clear the OCTL[DEVSETHNPEN] bit.

Internal ADP flow for B-device

The internal ADP flow of B-device is as follows:

Figure 28-36. Internal ADP flow for B-device

While probing, the application should wait for one of the following events to exit probing and start SRP request again.

- ADP change
- VBus detection
- New session request from B-device application

While sensing, the application should wait for one of the following events to exit sensing.

- ADP sensed failed
- VBUS detection
- New session request from B-device application

The application should continue with ADP probing if ADP sense failed. Else, the application should exit sensing and start SRP request again

28.5.6.1.11 Core entering b3_us_peripheral in B-Peripheral in HS/FS mode (Timeline for BDevSessVldDetEvnt)

The PCD decides the speed in which the USB3 core needs to operate by programming DCFG.DevSpd bits. Accordingly, the core decides to operate in FS to start with.

HS/FS Mode: BDevSessVldDetEvnt is triggered by a valid VBUS detection by B-Device in B-IDLE state. The B-Device will transition to B-PERIPHERAL state before triggering this event. Note that this is common to both SS and HS(FS).

Figure 28-37. Timeline for BDevSessVldDetEvnt in HS/FS mode

28.5.6.1.12 VBUS change detected on USB (Timeline for BDevVBusChngEvnt)

BDevVBusChngEvnt is triggered when a change in bvalid is detected irrespective of OTG state machine.

Figure 28-38. Timeline for BDevVBusChngEvnt

28.5.6.1.13 Internal ADP controller logic

The ADP controller has timers that assist in ADP operation. The ADP-related registers are a part of the pwrm_otgif module. For the PHY to generate ADPPRB and ADPSNS, the PHY needs to know the value to compare with VADP_PRB and VADP_SNS, respectively. These values can be hardcoded in the PHY or the PHY can have limited programmable option by which these reference voltage values can be changed. The

"Wakeup Logic" module implements SRP detection in A-Device mode. This is required if the USB 3.0 controller is powered down completely while ADP is still in progress and B-Device initiates SRP.

NOTE

ADP controller logic is inferred along with, but outside USB 3.0 controller.

- As a product, both ADP controller logic and OTG controller are packaged into USB 3.0 core.
- All ADP timers are maintained in the ADP controller module, which generates the enable and disable signaling to the PHY for ADP probing and sensing.
- PHY contains the following circuitry related to ADP functionality:
  - Comparators for PRB and SNS
  - I_ADP_SRC and I_ADP_SNK
  - V_{BUS} circuitry
- Application software programs the ADP timing registers that resides in the pwrm_otgif module.
- The pwrm_otgif module provides a mechanism to the application via interrupts to log and report events pertaining to ADP probing and sensing.
- In this operation, only the pwrm module needs to be always powered on. The pwrdwn module can be either powered on or off.

28.5.7 Initialization/application information

The PHY can be configured for fixed equalization by programming relevant control registers in the USB 3.0 PHY.

In order to initialize the USB 3.0 PHY, software should perform the following steps:

1. Write 1'b0 to RX_OVRD_IN_HI.RX_EQ_EN [address 16'h1006: bit 6].
2. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_EN_OVRD [address 16'h1006: bit 7].
3. Write a fixed value to RX_OVRD_IN_HI.RX_EQ [address 16'h1006: bits 10–8] [equalization setting] (generally 2–4, based on testing in customer environment).
4. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_OVRD [address 16'h1006: bit 11].

28.5.8 Power management overview

The following power management features are available in the USB 3.0 core:

- Hardware-controlled LPM in USB 2.0 host
- Clock gating in device (U1/U2/U3) and host (U1/U2/U3)

The power saving mechanism is hardware-controlled LPM. This is a USB 2.0 host-only feature where the host controller automatically detects the downstream port is idle for a certain amount of time and autonomously initiates an LPM token to the connected device. If the device accepts, the link is put into the standard USB 2.0 Suspend state.

Finally, clock gating is also supported in the host controller as well as in the device controller. The following table gives the list of power saving mechanisms available in USB 3.0 core.

Table 28-29. Power savings support

USB 3.0 State	USB 2.0 State	Device Power Savings	Host Power Savings
U1 (Hardware Initiated)	None	Core: Clock Gating PHY: P1	Core: Clock Gating PHY: P1
U2 (Hardware Initiated)	LPM-L1 (Hardware Initiated)	Core: Clock Gating, PHY: P2/Sleep	Core: Clock Gating, PHY: P2/ Sleep
U3 (Software Initiated)	Suspend (Software Initiated)	Core: Clock Gating, PHY: P3/Suspend	Core: Clock Gating, PHY: P3/ Suspend
Rx.Detect (Software Initiated)	None	None	Core: Clock Gating, PHY: P3/ Suspend
SS.Disabled (Software Initiated)	None	Core: Clock Gating, PHY: P3/Suspend	None

28.5.8.1 Clock gating

The core implements clock gating in the following situations:

- In USB 2.0 device mode
  - When the UTMI suspend or l1_suspend signal is asserted and the core is idle
- In USB 3.0 device mode, when the link is in U1, U2 or U3 state and the core is idle.

In host mode, clock gating is enabled in the following situations:

- When all USB 2.0 ports are in the suspend or L1 Suspend state, and all USB 3.0 ports are in the U1/U2/U3 state, and the core is idle.

Internal clock gating will apply to:

- Modules that use the ram_clk
- Some modules that use bus_clk

Internal clock gating will not apply to:

- Modules that use mac3_clk: When the USB 3.0 PHY is suspended, these modules switch to using suspend_clk, so their power consumption is reduced but the clock is not completely stopped.
- Modules that use mac2_clk: When the USB 2.0 PHY is suspended, these modules switch to using suspend_clk, so their power consumption is reduced but the clock is not completely stopped.
- The bus_gs module, which uses bus_clk. This module detects wakeup on the slave interface and thus needs a non-gated clock.

Clock gating can be enabled or disabled using the GCTL register. The GCTL register is "sticky," it retains its value across soft resets.

28.5.8.2 Hardware-Controlled LPM

In USB 2.0, the main link power management mechanism was for the host to suspend the link. However, suspending the link requires a long period of idle and was wasteful of power, especially because the host knows ahead of time whether it is going to schedule any traffic. It also required a long period of resume signaling.

The USB 2.0 Link Power Management Addendum [LPM-ECN] added a new token to the USB 2.0 specification, the LPM token. The host has the option of suspending the link quickly instead of waiting for a long period of idle, and communicates its desire to the device by sending an LPM token. The ECN also introduced a terminology for different link states. This new link state was called L1. The traditional USB 2.0 suspend was called L2. The L1 state requires a much shorter resume signaling period too.

An xHCI driver can initiate LPM by writing '2' to the PORTSC.PLS field. However, there is a lot of overhead for an xHCI driver to be constantly trying to keep the link in a low power state. Because of this, the xHCI specification also allows a compatible xHC to be enabled with "Hardware-Controlled LPM" by setting PORTPMSC.HLE as 1. When this happens, the xHC is expected to automatically send LPM tokens when it has no pending transfers and thinks that the link is idle enough to enter L1. This feature is supported by the host core.

28.5.8.2.1 Special consideration for OTG

When the core is configured as an OTG device, it will not initiate LPM when acting in a reverse role as a B- Host. This is because suspend on the link is treated as a request to switch back to its original role.

Chapter 29
Watchdog Timer Unit

29.1 Watchdog unit

The Watchdog module is intended to be used to apply a reset to the system in the event of a software failure and provide a way of recovering from software crashes. The module is based around a 32-bit down counter that is initialized from the Watchdog Load Register, WDOGLOAD. It can be enabled or disabled as required. The watchdog clock generates a regular interrupt, WDOGINT, depending on a programmed value. The counter decrements by one on each positive clock edge of WDOGCLK when WDOGCLKEN is HIGH. Asserting WDOGCLKEN is done by configuring PMU register: Physical Core Time Base Enable (PCTBENR). The watchdog monitors the interrupt and asserts a reset WDOGRES signal when the counter reaches zero and the counter is stopped. On the next enabled WDOGCLK clock edge the counter is reloaded from the WDOGLOAD Register and the count down sequence continues. If the interrupt is not cleared by the time that the counter next reaches zero then the Watchdog module reasserts the reset signal. The following figure shows the watchdog block diagram.

The chip supports nine watchdog units. WDOGn correspond to CORE(n-1), for example WDOG1 serving core 0. There is a dedicated watchdog timer for TrustZone.

For more details on the mapping of the cores to the watchdog units see CCSR Address Map table in the Memory Map chapter.

Figure 29-1. Watchdog components

29.2 Peripheral Identification Registers

Peripheral Identification Registers:

The WDOGPERIPHID0-3 registers are four 8-bit registers that span address locations 0xFE0-0xFEC. The read-only registers provide the following options for the Watchdog Unit (peripheral):

Part number [11:0]

This identifies the peripheral. The three-digit product code 0x805 is used for the watchdog unit.

Designer [19:12]

This is the identification of the designer. Arm Limited is 0x41 (ASCII A).

Revision number [23:20]

This is the revision number of the peripheral. The revision number starts from 0.

Configuration [31:24]

This is the configuration option of the peripheral. The configuration value is 0.

The figure below shows the register bit assignments.

Figure 29-2. WDOGPERIPHID0-3 Register bit assignments

NOTE

All memory accesses to the peripheral identification registers must be 32-bit, using the LDR and STR instructions.

The four 8-bit peripheral identification registers are described in the following subsections:

- Peripheral Identification Register 0 (WDOGPERIPHID0)
- Peripheral Identification Register 1 (WDOGPERIPHID1)
- Peripheral Identification Register 2 (WDOGPERIPHID2)
- Peripheral Identification Register 3 (WDOGPERIPHID3)

29.3 PrimeCell Identification Registers

PrimeCell Identification Registers :

The WDOGPCELLID0-3 Registers are four 8-bit wide registers that span address locations 0xFF0-0xFFC. The registers can conceptually be treated as a 32-bit register. The register is used as a standard cross-peripheral identification system. The WDOGPCELLID Register is set to 0xB105F00D. The figure below shows the register bit assignments.

Figure 29-3. WDOGPCELLID0-3 Register bit assignments

The four, 8-bit PrimeCell identification registers are described in the following subsections:

- PrimeCell Identification Register 0 (WDOGPCELLID0)
- PrimeCell Identification Register 1 (WDOGPCELLID1)
- PrimeCell Identification Register 2 (WDOGPCELLID2)
- PrimeCell Identification Register 3 (WDOGPCELLID3)

29.4 WDOG register descriptions

This section describes the Watchdog registers.

29.4.1 WDOG memory map

CL1_CORE0 base address: C00_0000h

CL1_CORE1 base address: C01_0000h

CL1_CORE2 base address: C02_0000h

CL1_CORE3 base address: C03_0000h

CL2_CORE0 base address: C10_0000h

CL2_CORE1 base address: C11_0000h

CL2_CORE2 base address: C12_0000h

CL2_CORE3 base address: C13_0000h

TZWDT base address: 23C_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Watchdog Load Register (WDOGLOAD)	32	RW	FFFF_FFFFh
4h	Watchdog Value Register (WDOGVALUE)	32	RW	FFFF_FFFFh
8h	Watchdog Control Register (WDOGCONTROL)	32	RW	0000_0000h
Ch	Watchdog Clear Interrupt Register (WDOGINTCLR)	32	WO	Table 29-
10h	Watchdog Raw Interrupt Status Register (WDOGRIS)	32	RO	0000_0000h
14h	Watchdog Interrupt Status Register (WDOGMIS)	32	RO	0000_0000h
C00h	Watchdog Lock Register (WDOGLOCK)	32	RW	0000_0000h
F00h	Watchdog Integration Test Control Register (WDOGITCR)	32	RW	0000_0000h
F04h	Watchdog Integration Test Output Set Register (WDOGITOP)	32	WO	0000_0005h
FE0h	Peripheral Identification Register 0 (WDOGPERIPHID0)	32	RO	0000_0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
FE4h	Peripheral Identification Register 1 (WDOGPERIPHID1)	32	RO	0000_0018h
FE8h	Peripheral Identification Register 2 (WDOGPERIPHID2)	32	RO	0000_0014h
FECh	Peripheral Identification Register 3 (WDOGPERIPHID3)	32	RO	0000_0000h
FF0h	PrimeCell Identification Register 0 (WDOGPCELLID0)	32	RO	0000_000Dh
FF4h	PrimeCell Identification Register 1 (WDOGPCELLID1)	32	RO	0000_00F0h
FF8h	PrimeCell Identification Register 2 (WDOGPCELLID2)	32	RO	0000_0005h
FFCh	PrimeCell Identification Register 3 (WDOGPCELLID3)	32	RO	0000_00B1h

29.4.2 Watchdog Load Register (WDOGLOAD)

29.4.2.1 Offset

Register	Offset
WDOGLOAD	0h

29.4.2.2 Function

The WDOGLOAD Register is a 32-bit register containing the value from which the counter is to decrement. When this register is written to, the count is immediately restarted from the new value. The minimum valid value for WDOGLOAD is 1.

29.4.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	WDOGLOAD															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	WDOGLOAD															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

29.4.2.4 Fields

Field	Function
31-0 WDOGLOAD	WDOGLOAD The value from which the counter is to decrement.

29.4.3 Watchdog Value Register (WDOGVALUE)

29.4.3.1 Offset

Register	Offset
WDOGVALUE	4h

29.4.3.2 Function

The WDOGVALUE Register gives the current value of the decrementing counter.

29.4.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WDOGVALUE															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WDOGVALUE															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

29.4.3.4 Fields

Field	Function
31-0 WDOGVALUE	WDOGVALUE The current value of the decrementing counter.

29.4.4 Watchdog Control Register (WDOGCONTROL)

29.4.4.1 Offset

Register	Offset
WDOGCONTROL	8h

29.4.4.2 Function

The WDOGCONTROL Register is a read/write register that enables the software to control the watchdog unit.

29.4.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														RESEN	INTEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.4.4.4 Fields

Field	Function
31-2 —	- Reserved .
1 RESEN	RESEN Enable Watchdog reset output (*WDOGRES*). Acts as a mask for the reset output. Set to 1 to enable the reset, and to 0 to disable the reset.
0 INTEN	INTEN Enable the interrupt event (*WDOGINT*). Set to 1 to enable the counter and the interrupt, and set to 0 to disable the counter and interrupt. Reloads the counter from the value in WDOGLOAD when the interrupt is enabled, and was previously disabled.

29.4.5 Watchdog Clear Interrupt Register (WDOGINTCLR)

29.4.5.1 Offset

Register	Offset
WDOGINTCLR	Ch

29.4.5.2 Function

A write of any value to the WDOGINTCLR Register clears the watchdog interrupt and reloads the counter from the value in WDOGLOAD.

29.4.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															Raw_Watchdog_Interrupt
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

29.4.5.4 Fields

Field	Function
31-1 —	- Reserved
0 Raw_Watchdog_Interrupt	Raw_Watchdog_Interrupt Raw interrupt status from the counter

29.4.6 Watchdog Raw Interrupt Status Register (WDOGRIS)

29.4.6.1 Offset

Register	Offset
WDOGRIS	10h

29.4.6.2 Function

The WDOGRIS Register is read-only. It indicates the raw interrupt status from the counter. This value is ANDed with the interrupt enable bit from the control register to create the masked interrupt, which is passed to the interrupt output pin.

29.4.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															Raw_Watchdog_Interrupt
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.4.6.4 Fields

Field	Function
31-1 —	- Reserved
0 Raw_Watchdog_Interrupt	Raw_Watchdog_Interrupt Raw interrupt status from the counter

29.4.7 Watchdog Interrupt Status Register (WDOGMIS)

29.4.7.1 Offset

Register	Offset
WDOGMIS	14h

29.4.7.2 Function

The WDOGMIS Register is read-only. It indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the INTEN bit from the control register, and is the same value which is passed to the interrupt output pin.

29.4.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															Watchdog_Interrupt
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.4.7.4 Fields

Field	Function
31-1 —	- Reserved
0	Watchdog_Interrupt

Field	Function
Watchdog_Interrupt	Enabled interrupt status from the counter

29.4.8 Watchdog Lock Register (WDOGLOCK)

29.4.8.1 Offset

Register	Offset
WDOGLOCK	C00h

29.4.8.2 Function

The WDOGLOCK Register is write-only. Use of this register causes write-access to all other registers to be disabled. This is to prevent rogue software from disabling the watchdog functionality. Writing a value of 0x1ACCE551 enables write access to all other registers. Writing any other value disables write accesses. A read from this register returns only the bottom bit:

- 0 indicates that write access is enabled (not locked)
- 1 indicates that write access is disabled (locked).

29.4.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Enable_register_writes															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Enable_register_writes															Register_write_enable_status
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.4.8.4 Fields

Field	Function
31-1 Enable_register_writes	Enable_register_writes Enable write access to all other registers by writing 0x1ACCE551. Disable write access by writing any other value.
0 Register_write_enable_status	Register_write_enable_status Register_write_enable_status 0b - write access to all other registers is enabled (default) 1b - write access to all other registers is disabled

29.4.9 Watchdog Integration Test Control Register (WDOGITCR)

29.4.9.1 Offset

Register	Offset
WDOGITCR	F00h

29.4.9.2 Function

The WDOGITCR Register is read/write. It is a single-bit register that enables integration test mode. When in this mode, the masked interrupt output, WDOGINT, and reset output, WDOGRES, are directly controlled by the test output set register.

29.4.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															Integration_Test_Mode_Enable
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.4.9.4 Fields

Field	Function
31-1 —	- Reserved
0 Integration_Test_Mode_Enable	Integration_Test_Mode_Enable When set to 1, places the Watchdog into integration test mode

29.4.10 Watchdog Integration Test Output Set Register (WDOGITOP)

29.4.10.1 Offset

Register	Offset
WDOGITOP	F04h

29.4.10.2 Function

The WDOGITOP Register is write-only. When in integration test mode, the enabled interrupt output and reset output are driven directly from the values in this register.

29.4.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														Integration_Test_WDOGINT_value	Integration_Test_WDOGRES_value
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

29.4.10.4 Fields

Field	Function
31-2 —	- Reserved
1 Integration_Test_WDOGINT_value	Integration_Test_WDOGINT_value Value output on WDOGINT when in Integration Test Mode
0 Integration_Test_WDOGRES_value	Integration_Test_WDOGRES_value Value output on WDOGRES when in Integration Test Mode

29.4.11 Peripheral Identification Register 0 (WDOGPERIPHID0)

29.4.11.1 Offset

Register	Offset
WDOGPERIPHID0	FE0h

29.4.11.2 Function

The WDOGPERIPHID0 Register is hard-coded and the fields within the register determine the reset value.

29.4.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								PartNumber0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

29.4.11.4 Fields

Field	Function
31-8 —	- Reserved
7-0 PartNumber0	PartNumber0 These bits read back as 0x05.

29.4.12 Peripheral Identification Register 1 (WDOGPERIPHID1)

29.4.12.1 Offset

Register	Offset
WDOGPERIPHID1	FE4h

29.4.12.2 Function

The WDOGPERIPHID1 Register is hard-coded and the fields within the register determine the reset value.

29.4.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Designer0				PartNumber1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

29.4.12.4 Fields

Field	Function
31-8 —	- Reserved
7-4 Designer0	Designer0 These bits read back as 0x1
3-0 PartNumber1	PartNumber1 These bits read back as 0x08

29.4.13 Peripheral Identification Register 2 (WDOGPERIPHID2)

29.4.13.1 Offset

Register	Offset
WDOGPERIPHID2	FE8h

29.4.13.2 Function

The WDOGPERIPHID2 Register is hard-coded and the fields within the register determine the reset value.

29.4.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Revision				Designer1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

29.4.13.4 Fields

Field	Function
31-8 —	- Reserved
7-4 Revision	Revision These bits read back as 0x1
3-0 Designer1	Designer1 These bits read back as 0x4

29.4.14 Peripheral Identification Register 3 (WDOGPERIPHID3)

29.4.14.1 Offset

Register	Offset
WDOGPERIPHID3	FECh

29.4.14.2 Function

The WDOGPERIPHID3 Register is hard-coded and the fields within the register determine the reset value.

29.4.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Configuration							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

29.4.14.4 Fields

Field	Function
31-8 —	- Reserved
7-0 Configuration	Configuration These bits read back as 0x00

29.4.15 PrimeCell Identification Register 0 (WDOGPCELLID0)

29.4.15.1 Offset

Register	Offset
WDOGPCELLID0	FF0h

29.4.15.2 Function

The WDOGPCELLID0 Register is hard-coded and the fields within the register determine the reset value.

29.4.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								WdogPCellID0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

29.4.15.4 Fields

Field	Function
31-8 —	- Reserved
7-0 WdogPCellID0	WdogPCellID0 These bits read back as 0x0D

29.4.16 PrimeCell Identification Register 1 (WDOGPCELLID1)

29.4.16.1 Offset

Register	Offset
WDOGPCELLID1	FF4h

29.4.16.2 Function

The WDOGPCELLID1 Register is hard-coded and the fields within the register determine the reset value.

29.4.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								WdogPCellID1							
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

29.4.16.4 Fields

Field	Function
31-8 —	- Reserved
7-0 WdogPCellID1	WdogPCellID1 These bits read back as 0xF0

29.4.17 PrimeCell Identification Register 2 (WDOGPCELLID2)

29.4.17.1 Offset

Register	Offset
WDOGPCELLID2	FF8h

29.4.17.2 Function

The WDOGPCELLID2 Register is hard-coded and the fields within the register determine the reset value.

29.4.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								WdogPCellID2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

29.4.17.4 Fields

Field	Function
31-8 —	- Reserved
7-0 WdogPCellID2	WdogPCellID2 These bits read back as 0x05

29.4.18 PrimeCell Identification Register 3 (WDOGPCELLID3)

29.4.18.1 Offset

Register	Offset
WDOGPCELLID3	FFCh

29.4.18.2 Function

The WDOGPCELLID3 register is hard-coded and the fields within the register determine the reset value.

29.4.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								WdogPCellID3							
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

29.4.18.4 Fields

Field	Function
31-8 —	- Reserved
7-0 WdogPCellID3	WdogPCellID3 These bits read back as 0xB1

Appendix A
LS1044A

A.1 Introduction

The LS1044A is the lowest power version of the LS1088A. The LS1044A integrated multicore processor supports four Arm® Cortex®-A53 processor cores.

It does not support AIOP.

This figure shows the major functional units within the chip.

Figure A-1. LS1044A block diagram

A.2 Overview of differences between LS1044A and LS1088A

The LS1044A is a lower power version of the LS1088A.

Table A-1. Product options

Features	LS1088A	LS1044A
Cores		
CPU Cores	8x Cortex-A53	4x Cortex-A53
	1.2GHz to 1.6GHz	1.2GHz to 1.6GHz
Threads	8	4
Cores power gated per cluster	0	4
Memory subsystem		
DDR Controller	32/64-bit DDR4	32/64-bit DDR4
	1600MT/s - 2100MT/s	1600MT/s - 2100MT/s
L2 Cache	2x 1MB	1MB
Peripherals		
Advanced IO Processor (AIOP)	Yes	No
Hardware Security Engine	Encrypt/Decrypt	
Ethernet MACs (WRIOP)	2x 10GE and 8x 1GE	1x 10GE and 8x 1GE
PCIe 3.0 Controllers	3	
USB 3.0/2.0 Controllers	2x USB3.0	
SATA 3.0 Controller	1x SATA3.0	
Secure Digital Host Controller	eSDHC	
Hardware-based virtualization and partitioning	Yes	
Single-source clocking	Yes (single external clock oscillator)	
Parallel Flash Interface	32-bit addr, 16-bit data	
SerDes	Eight lanes	
Pin Compatible	Yes	
Package	23x23mm, FCBGA, 0.8mm pitch	

A.3 LS1044A clocking and RCW differences

RCW[C1_PLL_SEL]
MUX
CGA_PLL1
CGA_PLL2
1/2
1/4
1/3
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/4
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/4
Cluster A to
A53 cluster 1
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/3
CGA_PLL1/4
CGA_PLL2/2
CGA_PLL2/3
QSPI
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/3
CGA_PLL2/4
CGA_PLL1/2
CGA_PLL1/3
Platform Clock
eSDHC
Platform PLL
Platform Clock
SYSCLK
32KHz CLK
GEN
CLK_32K (to RTC
input clock mux,
LP SECMON)
Clock Control
Sync Generation
Sync:HSIO Complex(1:2)
Sync:SerDes(1:2)
Sync:GIO Initiator(1:2)
Sync:GIO Target(1:2)
Sync:GIO_DIV2(1:4)
Sync:Debug(1:1)
Sync:SYSCLK(1:1)
Sync:CFP(1:N)
Sync:DPAA(1:1)
Sync:DPAA Div 2(1:2)
USB3PHY1
USB3PHY2
SD1_REF_CLK2
SD1_REF_CLK2_B
DIFF_SYSCLK
DIFF_SYSCLK_B
SS CLK IO
SerDes1
SD2_REF_CLK2
SD2_REF_CLK2_B
SerDes2
DDR CGU
DDRCLK
DDRPLL
DDR (1x Data Rate Clock)
RCW[MEM_PLL_RAT]
RCW[MEM_PLL_CFG]

Figure A-2. LS1044A Clocking diagram

A.4 LS1044A Register Differences from LS1088A

Table A-2. Unavailable Register Bits

Register Name	Bit Number	Description
Device Disable Register 3 (DCFG_DEVDISR3)	16 (AIOP)	Set to module is disabled

Table A-3. SVR and PCI Device IDs

Device	PCI / RapidIO Device ID (Hex)	SVR (Hex)
LS1044AE (with security)	0x80E2	0x87032210
LS1044A (without security)	0x80E3	0x87032310

A.5 LS1044A SerDes

LS1044A supports the same SerDes options as the LS1088A. It supports 1x 10GE and use SerDes 1 lane 0 (lane D).

Appendix B
LS1048A

B.1 Introduction

The QorIQ LS1048A Integrated Multicore Processor combines four Arm® Cortex®-A53 processor cores.

It supports AIOP.

This figure shows the major functional units within the chip.

Figure B-1. LS1048A block diagram

B.2 Overview of differences between LS1048A and LS1088A

The LS1048A is a lower power version of the LS1088A.

Table B-1. Product options

Features	LS1088A	LS1048A
Cores		
CPU Cores	8x Cortex-A53	4x Cortex-A53
	1.2GHz to 1.6GHz	1.2GHz to 1.6GHz
Threads	8	4
Active Clusters	2	1
Cores power gated per cluster	0	4
Memory subsystem		
L2 Cache	2x 1MB	1MB
DDR Controller	32/64-bit DDR4	32/64-bit DDR4
	1600MT/s - 2100MT/s	1600MT/s - 2100MT/s
Peripherals		
Advanced IO Processor (AIOP)	Yes	Yes
Hardware Security Engine	Encrypt/Decrypt	
Ethernet MACs (WRIOP)	2x 10GE and 8x 1GE	1x 10GE and 8x 1GE
PCIe 3.0 Controllers	3	
USB 3.0/2.0 Controllers	2x USB3.0	
SATA 3.0 Controller	1x SATA3.0	
Secure Digital Host Controller	eSDHC	
Hardware-based virtualization and partitioning	Yes	
Single-source clocking	Yes (single external clock oscillator)	
Parallel Flash Interface	32-bit addr, 16-bit data	
SerDes	Eight lanes (4+4, S1+S2)	
Pin Compatible	Yes	
Package	23x23mm, FCBGA, 0.8mm pitch	

B.3 LS1048A clocking and RCW differences

RCW[C1_PLL_SEL]
MUX
CGA_PLL1
CGA_PLL2
1/2
1/4
1/3
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/4
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/4
Cluster A to A53 cluster 1
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/3
CGA_PLL1/4
CGA_PLL2/2
CGA_PLL2/3
QSPI
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/3
CGA_PLL2/4
CGA_PLL1/2
CGA_PLL1/3
Platform Clock
eSDHC
Platform PLL
Platform Clock
SYSCLK
32KHz CLK GEN
CLK_32K (to RTC input clock mux, LP SECMON)
Clock Control Sync Generation
Sync:HSIO Complex(1:2)
Sync:SerDes(1:2)
Sync:GIO Initiator(1:2)
Sync:GIO Target(1:2)
Sync:GIO_DIV2(1:4)
Sync:Debug(1:1)
Sync:SYSCLK(1:1)
Sync:CFP(1:N)
Sync:DPAA(1:1)
Sync:DPAA Div 2(1:2)
USB3PHY1
USB3PHY2
SD1_REF_CLK2
SD1_REF_CLK2_B
DIFF_SYSCLK
DIFF_SYSCLK_B
SS CLK IO
SD2_REF_CLK2
SD2_REF_CLK2_B
SerDes1
SerDes2
DDR CGU
DDRCLK
DDRPLL
DDR (1x Data Rate Clock)
RCW[MEM_PLL_RAT]
RCW[MEM_PLL_CFG]

Figure B-2. LS1048A Clocking diagram

B.4 LS1048A Register Differences from LS1088A

Table B-2. SVR and PCI Device IDs

Device	PCI Device ID (Hex)	SVR (Hex)
LS1048AE (with security)	0x80E0	0x8703_2110
LS1048A (without security)	0x80E1	0x8703_2010

B.5 LS1048A SerDes

LS1048A supports the same SerDes options as the LS1088A. It supports 1x 10GE and use SerDes 1 lane 0 (lane D).

Appendix C
LS1084A

C.1 Introduction

The QorIQ LS1084A Integrated Multicore Processor combines eight Arm® Cortex®-A53 processor cores.

It does not support AIOP.

This figure shows the major functional units within the chip.

Figure C-1. LS1084A block diagram

C.2 Overview of differences between LS1048A and LS1088A

The LS1084A is a lower power version of the LS1088A.

Table C-1. Product options

Features	LS1088A	LS1084A
Cores		
CPU Cores	8x Cortex-A53	8x Cortex-A53
	1.2GHz to 1.6GHz	1.2GHz to 1.6GHz
Threads	8	8
Cores power gated per cluster	0	0
Memory subsystem		
DDR Controller	32/64-bit DDR4	32/64-bit DDR4
	1600MT/s - 2100MT/s	1600MT/s - 2100MT/s
L2 Cache	2x 1MB	2x 1MB
Peripherals		
Advanced IO Processor (AIOP)	Yes	No
Hardware Security Engine	Encrypt/Decrypt	
Ethernet MACs (WRIOP)	2x 10GE and 8x 1GE	2x 10GE and 8x 1GE
PCIe 3.0 Controllers	3	
USB 3.0/2.0 Controllers	2x USB3.0	
SATA 3.0 Controller	1x SATA3.0	
Secure Digital Host Controller	eSDHC	
Hardware-based virtualization and partitioning	Yes	
Single-source clocking	Yes (single external clock oscillator)	
Parallel Flash Interface	32-bit addr, 16-bit data	
SerDes	Eight lanes	
Pin Compatible	Yes	
Package	23x23mm, FCBGA, 0.8mm pitch	

C.3 LS1084A clocking and RCW differences

RCW[C1_PLL_SEL]
RCW[C2_PLL_SEL]
MUX
CGA_PLL1
CGA_PLL2
1/2
1/4
1/3
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/4
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/4
Cluster A to A53 cluster 1
Cluster A to A53 cluster 2
CGA_PLL1
CGA_PLL1/2
CGA_PLL1/3
CGA_PLL1/4
CGA_PLL2/2
CGA_PLL2/3
QSPI
CGA_PLL2
CGA_PLL2/2
CGA_PLL2/3
CGA_PLL2/4
CGA_PLL1/2
CGA_PLL1/3
Platform Clock
eSDHC
Platform PLL
Platform Clock
SYSCLK
32KHz CLK GEN
CLK_32K (to RTC input clock mux, LP SECMON)
Clock Control Sync Generation
Sync:HSIO Complex(1:2)
Sync:SerDes(1:2)
Sync:GIO Initiator(1:2)
Sync:GIO Target(1:2)
Sync:GIO_DIV2(1:4)
Sync:Debug(1:1)
Sync:SYSCLK(1:1)
Sync:CFP(1:N)
Sync:DPAA(1:1)
Sync:DPAA Div 2(1:2)
USB3PHY1
USB3PHY2
SD1_REF_CLK2
SD1_REF_CLK2_B
DIFF_SYSCLK
DIFF_SYSCLK_B
SS CLK IO
SD2_REF_CLK2
SD2_REF_CLK2_B
SerDes1
SerDes2
DDR CGU
DDRCLK
DDRPLL
DDR (1x Data Rate Clock)
RCW[MEM_PLL_RAT]
RCW[MEM_PLL_CFG]

Figure C-2. LS1084A Clocking diagram

C.4 LS1084A Register Differences from LS1088A

Table C-2. Unavailable Register Bits

Register Name	Bit Number	Description
Device Disable Register 3 (DCFG_DEVDISR3)	16 (AIOP)	Set to module is disabled

Table C-3. SVR and PCI Device IDs

Device	PCI / Device ID (Hex)	SVR (Hex)
LS1084AE (with security)	0x80C2	0x87030210
LS1084A (without security)	0x80C3	0x87030310

C.5 LS1084A SerDes

LS1084A supports the same SerDes options as the LS1088A.

Appendix D
Event Processing Unit (EPU)

D.1 EPU register descriptions

All registers are accessible with 32-bit accesses. The EPU registers are configurable via DCSR.

For example, to access the EPEVTCR0 register, the following physical address must be used.

($DCSR_BASE + $EPU_BASE + register offset) = 0x07_0000_0000 + 0x6_0000 + 0x50= 0x07_0006_0050

D.1.1 EPU_dcsr memory map

EPU base address: 6_0000h

Offset	Register	Width (In bits)	Access	Reset value
50h - 74h	Event Processor EVT Pin Control Register n (EPEVTCR0 - EPEVTCR9)	32	RW	0000_0000h
FD0h - FECh	Event Processor Debug Reservation Register n (EPRSRV0 - EPRSRV7)	32	RW	0000_0000h

D.1.2 Event Processor EVT Pin Control Register n (EPEVTCR0 - EPEVTCR9)

D.1.2.1 Offset

For a = 0 to 9:

Register	Offset
EPEVTCRa	50h + (a × 4h)

D.1.2.2 Function

The SoC's EVT pins are muxed among different functionalities. For each EVT pin, a control bit is set within this register to indicate the input or output direction. A value of zero means the pin is an input to EPU. A value of one means the pin is an output of EPU.

D.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EVT_SEL				0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DIR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D.1.2.4 Fields

Field	Function
31-28 EVT_SEL	Event Select. Value determines which event is mapped to this EVT pin when it is configured as an output 0000b - SCU0 drives EVTn when DIR=1 0001b - SCU1 drives EVTn when DIR=1 0010b - SCU2 drives EVTn when DIR=1 0011b - SCU3 drives EVTn when DIR=1 0100b - SCU4 drives EVTn when DIR=1 0101b - SCU5 drives EVTn when DIR=1 0110b - SCU6 drives EVTn when DIR=1 0111b - SCU7 drives EVTn when DIR=1 1000b - SCU8 drives EVTn when DIR=1 1001b - SCU9 drives EVTn when DIR=1 1010b - SCU10 drives EVTn when DIR=1

Table continues on the next page...

Field	Function
	1011b - SCU11 drives EVTn when DIR=1 1100b - SCU12 drives EVTn when DIR=1 1101b - SCU13 drives EVTn when DIR=1 1110b - SCU14 drives EVTn when DIR=1 1111b - SCU15 drives EVTn when DIR=1
27-1 —	Reserved
0 DIR	Direction 0b - = EVTn is a device input 1b - = EVTn is a device output driven by EPU

D.1.3 Event Processor Debug Reservation Register n (EPRSRV0 - EPRSRV7)

D.1.3.1 Offset

For a = 0 to 7:

Register	Offset
EPRSRVa	FECh + (a × -4h)

D.1.3.2 Function

The debug reservation assist registers consist of a set of hardware semaphores and a bank of read/write scratch registers (reservation block). The intent is that these can be used to facilitate the implementation of a resource sharing protocol for device debug resources among the various tools and agents that are likely to want access to them.

The debug reservation block is a bank of scratch registers with read/write access which are intended for use in developing a resource sharing protocol. The nature of the protocol is at the discretion of software and is not enforced in any way by hardware.

D.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

D.1.3.4 Fields

Field	Function
31-0 SCRATCH	Scratch space available for implementing a resource sharing protocol.

Appendix E
Terminology, Conventions, and Resources

E.1 About this content

The primary objective of this document is to define the functionality of *LS1088A QorIQ Integrated Multicore Processor*. The *LS1088A* provides integration of processing power for networking and communications peripherals, resulting in higher device performance.

NOTE

The diagrams in this content are provided to aid in understanding the overall functionality and features of this product. They do not depict the implementation details of the product, which are subject to change.

E.2 Audience

It is assumed that the reader understands operating systems, microprocessor system design, and the basic principles of RISC processing.

E.3 Acronyms and abbreviations

This table describes commonly-used acronyms and abbreviations used in this document.

Table E-1. Acronyms and abbreviations

Acronym/ Abbreviation	Meaning
8b/10b	8-bit/10-bit encoding
AIC	Antenna interface controller
AMC	Advanced mezzanine card
ATM	Asynchronous transfer mode

Table continues on the next page...

Table E-1. Acronyms and abbreviations (continued)

Acronym/ Abbreviation	Meaning
ATMU	Address translation and mapping unit
BD	Buffer descriptor
BTB	Branch target buffer
BUID	Bus unit ID
CAM	Content-addressable memory
CCB	Core complex bus
CCSR	Configuration control and status register
CLASS	Chip-level arbitration and switching system
CRC	Cyclic redundancy check
DCSR	Debug configuration and status register
DDR	Double data-rate
DIP	Dual inline package
DPLL	Digital phase-locked loop
DTLB	Data translation lookaside buffer
dTSEC	Data path three-speed Ethernet controller
DUART	Dual universal asynchronous receiver/transmitter
ECC	Error checking and correction
ECM	e500 coherency module
EEST	Enhanced Ethernet serial transceiver
EHCI	Enhanced host controller interface
eLBC	Enhanced local bus controller
EPROM	Erasable programmable read-only memory
EEPROM	Electrically-erasable programmable read-only memory
eTSEC	Enhanced three-speed Ethernet controller
FCS	Frame-check sequence
FIFO	First in, first out
GCI	General circuit interface
GMII	Gigabit media-independent interface
GPCM	General-purpose chip-select machine
GPIO	General-purpose I/O
GPR	General-purpose register
I2C	Inter-integrated circuit
IFC	Intergrated Flash controller
IPG	Interpacket gap
IrDA	Infrared Data Association
ISDN	Integrated services digital network
ITLB	Instruction translation lookaside buffer
IU	Integer unit
HSSI	High-speed serial interface

Table continues on the next page...

Table E-1. Acronyms and abbreviations (continued)

Acronym/ Abbreviation	Meaning
LAE	Local access error
LAW	Local access window
LBC	Local bus controller
LIFO	Last-in-first-out
LRU	Least recently used
LSB	Least significant byte
lsb	Least significant bit
LSU	Load/store unit
MAC	Multiply accumulate, media access control
MDI	Medium-dependent interface
MII	Media independent interface
MMU	Memory management unit
MSB	Most significant byte
msb	Most significant bit
NMI	Non-maskable interrupt
NMSI	Nonmultiplexed serial interface
No-op	No operation
OCeaN	On-chip network
OC	
OSI	Open systems interconnection
PCS	Physical coding sublayer
PIC	Programmable interrupt controller
PMA	Physical medium attachment
PMD	Physical medium dependent
PLL	Phase-locked loop
POR	Power-on reset
PRI	Primary rate interface
PWM	Pulse-width modulation
RAID	Redundant array of independent drives
RGMII	Reduced gigabit media-independent interface
RIO	RapidIO
RTOS	Real-time operating system
RWITM	Read with intent to modify
RMW	Read-modify-write
Rx	Receiver
RxBD	Receive buffer descriptor
SATA	Serial advanced technology attachment
SCP	Serial control port
SDLC	Synchronous data link control

Table continues on the next page...

Table E-1. Acronyms and abbreviations (continued)

Acronym/ Abbreviation	Meaning
SDMA	Serial DMA
SD/MMC	Secure digital/multimedia card
SDOS	SmartDSP operating system
SEC	Security Engine
SerDes	Serializer/Deserializer
SFD	Start frame delimiter
SI	Serial interface
SIU	System interface unit
SMC	Serial management controller
SPI	Serial peripheral interface
SPR	Special-purpose register
SRAM	Static random access memory
TAP	Test access port
TBI	Ten-bit interface
TDM	Time-division multiplexed
TLB	Translation lookaside buffer
TSA	Time-slot assigner
Tx	Transmitter
TxBD	Transmit buffer descriptor
UART	Universal asynchronous receiver/transmitter
UPM	User-programmable machine
uTCA	Micro telecommunications computing platform
UTP	Unshielded twisted pair
VA	Virtual address
ZBT	Zero bus turnaround

E.4 Notational conventions

This table shows notational conventions used in this content.

Table E-2. Notational conventions

Convention	Definition
General	
Cleared	When a bit takes the value zero, it is said to be cleared.
Set	When a bit takes the value one, it is said to be set.
mnemonics	Instruction mnemonics are shown in lowercase bold.

Table continues on the next page...

Table E-2. Notational conventions (continued)

Convention	Definition
italics	Italics can indicate the following: • Variable command parameters, for example, **bcctr***x* • Titles of publications • Internal signals, for example, $\overline{core\ int}$
0x	Prefix to denote hexadecimal number
h	Suffix to denote hexadecimal number
0b	Prefix to denote binary number
b	Suffix to denote binary number
rA, **r**B	Instruction syntax used to identify a source GPR
rD	Instruction syntax used to identify a destination GPR
REGISTER[FIELD]	Abbreviations for registers are shown in uppercase text. Specific bits, fields, or ranges appear in brackets. For example, MSR[LE] refers to the little-endian mode enable bit in the machine state register.
x	In some contexts, such as signal encodings, an unitalicized x indicates a don't care.
x	An italicized *x* indicates an alphanumeric variable
n	An italicized *n* indicates either: • An integer variable • A general-purpose bitfield unknown
Â¬	NOT logical operator
&	AND logical operator
|	OR logical operator
||	Concatenation, for example, TCR[WPEXT] || TCR[WP]
Signals	
$\overline{\text{OVERBAR}}$	An overbar indicates that a signal is active-low.
lowercase_italics	Lowercase italics is used to indicate internal signals
lowercase_plaintext	Lowercase plain text is used to indicate signals that are used for configuration.
Register access	
Reserved	Ignored for the purposes of determining access type
R/W	Indicates that all non-reserved fields in a register are read/write
R	Indicates that all non-reserved fields in a register are read only
W	Indicates that all non-reserved fields in a register are write only
w1c	Indicates that all non-reserved fields in a register are cleared by writing ones to them

E.5 Related resources

This table shows related resources that may be helpful. Additional literature is published as new processors become available. For current literature, visit www.NXP.com or contact your local FAE.

Table E-3. Related resources

Resource	Purpose
Data Sheet	Provides specific data regarding bus timing, signal behavior, and AC, DC, and thermal characteristics, as well as other design considerations
DPAA reference manual	Provides specific data regarding data path acceleration architecture including its implementation in the chip.
SEC reference manual	Provides specific data regarding the security engine implementation in the chip.
AEC	Provides specific data regarding bus timing, signal behavior, and AC, DC, and thermal characteristics, as well as other design considerations

For the Arm modules, refer the following resources:

Table E-4. Related resources from Arm

Resource	IP Revision
Arm® CoreLink™ GIC-500 Generic Interrupt Controller	r1p1
Arm® Cortex®-A53 MPCore Processor	r0p4
Arm® CoreLink™ MMU-500 System Memory Management Unit	r2p4
Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile	v8.0

Appendix F
Revision History

F.1 Substantive changes from revision 0 to revision 1

Substantive changes from revision 0 to revision 1 are as follows:

F.1.1 Overview Revision History

Reference	Description
DDR memory controller	Updated the DDR speed from 2.1 GT/s to 2.133 GT/s.

F.1.2 Memory Map Revision History

Reference	Description
DCSR address map	Added the DCSR memory map section.

F.1.3 Signal Descriptions Revision History

No substantive changes

F.1.4 Reset, Clocking, and Initialization Revision History

Reference	Description
Platform Clock Control/Status Register (CLKCSR)	Removed note from the CLKOEN1 bit description.

Table continues on the next page...

Reference	Description
	In the CLKOSEL1 bit description, updated CLKOSEL1[1:0] to CLKOSEL1[4:0]. In the CLKOSEL1 bit description, updated the 11000b and 11001b settings.
Selecting the RCW Source and "Flash Mode"	Removed footnote reference from the bit setting 7'b100_1001.
Reset Control Word (RCW) Field Descriptions	Removed duplicate note from the MEM_PLL_RAT description. In the HWA_CGA_M1_CLK_SEL description, added the following sentence: "This bit selects the QuadSPI input clock." In the HWA_CGA_M2_CLK_SEL description, added the following sentence: "This bit selects the eSDHC input clock." In the BOOT_LOC description, updated RCW[IFC_MODE] to RCW[FLASH_MODE]. In the USB1_CLK_FSEL and USB2_CLK_FSEL bit descriptions, added the following sentence: "This bit should be set to 0b100111." In the SRDS_PLL_REF_CLK_SEL_S1 description, removed the setting for 161.13 MHz. Added RCW[461] as QSPI_OCT_EN. Updated the bit description for 0b010 in RCW[EC2].
Figure 4-2	Updated the SerDes clock distribution.

F.1.5 Interrupt Assignments Revision History

Reference	Description
ISC register descriptions	Updated the register endianness to little endian.

F.1.6 Arm Modules Revision History

No substantive changes

F.1.7 Service Processor Revision History

Reference	Description
CCSR Write	Added the following note USB PHY registers should not be accessed during the PBI phase before the system is ready.
Error Codes	Added row for value 0xA4.

F.1.8 DCFG Revision History

Reference	Description
Device Config register descriptions	Removed the following register: 0x24h - 0x2Ch: General-Purpose POR Configuration Register (GPPORCR2) - General-Purpose POR Configuration Register (GPPORCR4). 88h: Device Disable Register 7 (DEVDISR7) A0h: Processor Version Register (PVR) 600h -604h: DMA Control Register (DMACR1) - DMA Control Register (DMACR2) 240h - 27Ch: Scratch Read / Write Register n (SCRATCHRW17) - Scratch Read / Write Register n (SCRATCHRW32) 620h - 638h: General Control Register (GENCR1) - General Control Register (GENCR7) 640h: Core General Status Register (CGENSR1) 660h: Core General Control Register (CGENCR1) 700h: SRIO Prescaler Timer Enable Control Register (SRIOPSTECR) 904h - 90Ch: QSPI Control Register (QSPICR2) - QSPI Control Register (QSPICR4) 940h: UART Connectivity Control Register (UCCR)
QSPI Control Register (QSPICR1)	Updated the description of QSPICR1[QSPI_CLK_DIV] bit description.
Reset Configuration Word Status Register 28 (RCWSR28)	Updated the USB1_CLK_FSEL and USB2_CLK_FSEL bit descriptions to add the following: This bit should be set to 0b100111.
Reset Configuration Word Status Register 6 (RCWSR6)	In the HWA_CGA_M1_CLK_SEL bit description, added the following sentence: "This bit selects the QuadSPI input clock." In the HWA_CGA_M2_CLK_SEL bit description, added the following sentence: "This bit selects the eSDHC input clock."
Reset Configuration Word Status Register 9 (RCWSR9)	In the BOOT_LOC bit description, updated RCW[IFC_MODE] to RCW[FLASH_MODE].
Reset Configuration Word Status Register 30 (RCWSR30)	In the SRDS_PLL_REF_CLK_SEL_S1 bit description, removed the setting for 161.13 MHz.
Pin Control Analog Register Descriptions	Removed the following registers: 0h - 10h: Slewrate Analog Control 1 (slewrate_ana_cntrl1) - Slewrate Analog Control 5 (slewrate_ana_cntrl5) 20h: Drive Strength Analog Control 1 (drv_strgth_ana_cntrl1) 24h - 2Ch: LVTTL Analog Control 1 (lvttl_ana_cntrl1) - LVTTL Analog Control 3 (lvttl_ana_cntrl3)

Table continues on the next page...

Reference	Description
System Version Register (SVR)	Updated the SVR[VAR_PER] bit description to add settings for various device personalities.
SVR and PCI Device ID	Removed this section.
DCFG_PIN_CTRL_PIN_CTRL	Added base address.

F.1.9 DPAA2 Overview and Implementation Revision History

No substantive changes

F.1.10 QUICC Engine Block Revision History

No substantive changes

F.1.11 Cache Coherent Interconnect Revision History

Reference	Description
LS1088A transaction shareability configuration	Updated the register reference for SEC and qDMA.

F.1.12 DDR Revision History

Reference	Description
DDR Features	Removed the duplicate entry for chip interleaving support. Removed the partial array self refresh support.
DDR register descriptions	Removed the following registers: C0h - CCh: Chip select a configuration 2 (CS0_CONFIG_2 - CS3_CONFIG_2)
DDR SDRAM Address Multiplexing	Removed partial array self refresh support throughout. Reversed the order of bits to 34-0 in all tables. Updated the DDR4 Address Multiplexing for 64-Bit Data Bus with no Interleaving table. Updated the Example of Address Multiplexing for 32/64-Bit Data Bus Interleaving Between Two Banks table. Updated the Example of Address Multiplexing for 64-Bit Data Bus Interleaving Between Four Banks table.

Table continues on the next page...

Reference	Description
DDR Memory Test Pattern n Register (DDR_MTP0 - DDR_MTP9)	Updated the instantiation of MTP register from 1-10 to 0-9.
Chip select a configuration (CS0_CONFIG - CS3_CONFIG)	In the BA_BITS_CS bit description, updated the setting for 01 from 3 logical bank bits to reserved. Updated the BG_BITS_CS bit description.
DDR SDRAM control configuration 2 (DDR_SDRAM_CFG_2)	Removed the following note from the OBC_CFG bit description: "Note that OBC_CFG can only be set if using a 64-bit data bus, and it cannot be set if DDR_SDRAM_CFG_3[CRC_EN] is set."
DDR SDRAM timing configuration 5 (TIMING_CFG_5)	Added the following note in the WODT_OFF bit description: Value of 4 clock cycles must be selected for this field.
DDR SDRAM timing configuration 2 (TIMING_CFG_2)	Updated bits 9-12 from WR_DATA_DELAY to reserved.
DDR SDRAM timing configuration 7 (TIMING_CFG_7)	Updated the PAR_LAT bit description.
Error Checking and Correcting (ECC)	Reversed the order of Syndrome bits to lsb0 in both tables.
Table 12-17	Removed row for CSn_CONFIG_2.
Programming Summary	Updated description of WRLVL_EN to "Should be set to 1 as write leveling is recommended. All other fields in DDR_WRLVL_CNTL, DDR_WRLVL_CNTRL_2, and DDR_WRLVL_CNTL_3 should be programmed appropriately based on the DRAM specifications and board layout specifics."
DDR Control Driver Register 1 (DDRCDR_1) DDR Control Driver Register 2 (DDRCDR_2)	Updated the reset values for DDRCDR_1 and DDRCDR_2 to 0x0.

F.1.13 DUART Revision History

Reference	Description
UART FIFO control register (UFCR1 - UFCR2)	Updated the bit 3 as reserved.

F.1.14 Enhanced Secured Digital Host Controller (eSDHC) Revision History

Reference	Description
Throughout	Added the following signals: SDHC_CLK_SYNC_IN SDHC_CLK_SYNC_OUT SDHC_CMD_DIR SDHC_DAT0-3_DIR

Table continues on the next page...

Reference	Description
Host controller capabilities register (HOSTCAPBLT)	Updated the reset value to 0x35F2_0000.
Host controller capabilities register 2 (HOSTCAPBLT2)	Updated the reset value to 0x0000_AF07. Added the DDR50 bit.
Tuning block control register (TBCTL)	Updated the reset value to 0x0000_0002.
SD Clock Control Register (SDCLKCTL)	Added this register.
Auto CMD Error Status Register / System Control 2 Register (AUTOCERR_SYSCTL2)	Updated the UHSM bit description.
Tuning block procedure	Added the following note: Tuning data timeout occur with default value of SYSCTL[DTOCV] when SD card send tuning daResolution w.r.t. MC WDT reset request functional supportta after 450 µs. According to SD host controller specification a card may take up to 150 ms (or at least 100 ms) to respond to the first tuning block.
clocking constraints	Added the Clocking constraints section.
DDR	Added the DDR section.
Interfacing Card	Added the Interfacing card section.
System Control Register when ESDHCCTL[CRS=1] (SYSCTL_ESDHCCTL_CRS_1)	Updated the reset value to 0x0000_0000.

F.1.15 FlexTimer Module Revision History

Reference	Description
FlexTimer as Implemented on the chip	Added the following note: The fixed frequency clock is connected to the RTC pin.

F.1.16 GPIO Revision History

No substantive changes

F.1.17 Integrated Flash Controller (IFC) Revision History

Reference	Description
Internal connectivity of WP signal	Removed 25-/28-bit address mode support and added the note.
Figure 17-1	Removed the signal notation from the block diagram.
External signal descriptions	Added the SOF_L_B signal in the WE_B signal description.

F.1.18 I2C Revision History

Reference	Description
Advantages of the I2C bus	Removed this section.
Module block diagram	Updated the arbitration control block in the block diagram.
Signal overview	Updated the second paragraph.
I2C register descriptions	Updated the memory map description.
Divider and hold values	Updated the section.
Transmit/receive sequence	Updated IBDIS to MDIS.
Table 18-16	Added steps i and j.
Table 18-18	Added steps j and k.
Register figure conventions	Removed this sub-section under the Memory map and register definition section.
Figure 18-9	Updated the figure.
Features	Removed "Maximum communication length...".
STOP mode	Removed "Program IBCR[TXRX]=0." Removed the following sentence: "You must ensure that the bus is free when STOP is requested."

F.1.19 PCI Express Revision History

Reference	Description
MEM Match Modes	Added the following note: The target (translation) address and base address of the inbound iATU window must be aligned based on the window size programmed in the corresponding BAR MASK register.
PCI Express Revision ID Register (Revision_ID_Register) PCI Express Revision ID Register (Revision_ID_Register)	Updated the reset value from 0x01 to 0x10.
PCI Express Device Control Register (Device_Control_Register)	Updated the reset value from 0x2810 to 0x2910.
Gen3 Control Register (GEN3_RELATED_OFF) Gen3 Control Register (GEN3_RELATED_OFF)	Updated reset value of the Gen3 control register from ox0 to 0x00800001.
PCI Express Device Capabilities 2 Register (Device_Capabilities_2_Register)	Updated the reset value from 0x0000_003F to 0x0000_001F.
PCI Express Class Code Register (programming interface) (Class_Code_Register_a)	Updated the 24-bit Class_Code_Register to separate it into 3 8-bit registers (offsets 0x9, 0xA, and 0xB).
PCI Express Class Code Register (sub class) (Class_Code_Register_b)	Renamed the class code registers and updated reset values for programming interface and base class.

Table continues on the next page...

Reference	Description
PCI Express Class Code Register (base class) (Class_Code_Register_c) PCI Express Class Code Register (programming interface) (Class_Code_Register_a) PCI Express Class Code Register (sub class) (Class_Code_Register_b) PCI Express Class Code Register (base class) (Class_Code_Register_c)	
PCI Express Device Capabilities Register (Device_Capabilities_Register) PCI Express Device Capabilities Register (Device_Capabilities_Register)	Updated description of Device_Capabilities_Register[FLRC].
PCI Express Device Control 2 Register (Device_Control_2_Register) PCI Express Device Control 2 Register (Device_Control_2_Register)	Added the IDO_CPL_EN and IDO_REQ_EN bits.
PCI Express Link Control 2 Register (Link_Control_2_Register) PCI Express Link Control 2 Register (Link_Control_2_Register)	Added the following statement in the Link_Control_2_Register[TxM] bit description: This bit is applicable only for the Gen2 speed.
iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0) iATU Region Control 1 Register (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)	Updated bit 16-17 from AT to reserved.
iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0) iATU Region Control 2 Register (IATU_REGION_CTRL_2_OFF_INBOUND_0)	Updated bit 18 from AT_MATCH_EN to reserved.
PCI Express Link Capabilities Register (Link_Capabilities_Register)	Updated reset value of the Link_Capabilities_Register from 0x0043_F483 to 0x0043_F443.
PCI Express Link Status Register (Link_Status_Register)	Updated NEG_LINK_W bit description to add setting for 0x4.
Lane Reversal	Added row for x2 link without lane reversal.
PCI Express Interrupt Pin Register (Interrupt_Pin_Register)	Removed values for INTB, INTC, and INTD from the Interrupt_Pin_Register[Interrupt_pin] field description.

F.1.20 Power Management Revision History

Reference	Description
Central SoC Power Management Unit Features	Removed the LPM35/40 mode.
Cluster Skyros SINACT Set Control Register (CLSINACTSETR)	Updated the register description.

F.1.21 QuadSPI Revision History

Reference	Description
Peripheral Bus Register Descriptions	Added note in QuadSPI_RSER[TBFDE] bit description.
QuadSPI as implemented on the chip	Added the features and list of sections that are not supported in the chip.

F.1.22 SATA Revision History

Reference	Description
Port Phy1Cfg register (PPCFG)	Updated the TTA bit description.
Port x command list base address register (PxCLB) Port x command list base address upper 32-bit register (PxCLBU) Port x FIS base address register (PxFB) Port x FIS base address upper 32-bit register (PxFBU) Port x interrupt status register (PxIS) Port x command and status register (PxCMD) Port x command issue register (PxCI)	Added register description for these registers.

F.1.23 SerDes Module Revision History

Reference	Description
MDIO registers	Separated the MDIO registers into separate MDIO register space sections.
Non Networking SerDes Protocol Lane Configuration	Updated RCW[SRDS_PRTCL_S1_LN#] to RCW[SRDS_PRTCL_S2_LN#]. Updated 0-3 lane assignment from D-A to A-D.
Rules for Selecting Valid Networking SerDes Options	Updated 57 and 58 as power down values for SerDes1 PLL2.
Features	Updated IEEE standard. Removed 10GBASE-R.
SerDes PLLa Reset Control Register (PLL1RSTCTL - PLL2RSTCTL)	Updated the reset value of PLL1RSTCTL and PLL2RSTCTL registers from 0x00000020 to 0x0467_452F.
SerDes PLLa Control Register 1 (PLL1CR1 - PLL2CR1)	Updated the reset value of PLL1CR1 and PLL2CR1 registers from 0x8000_0000 to 0x0800_4100.

Table continues on the next page...

Substantive changes from revision 0 to revision 1

Reference	Description
Speed Switch Control Register 0 - Lane a (LNASSCR0 - LNDSSCR0)	Updated the reset value of LNASSCR0 to LNHSSCR0 registers from 0x9800_2B00 to 0x9828_2B00.
SerDes PLLa Reset Control Register (PLL1RSTCTL - PLL2RSTCTL)	Updated the register description. Updated the RST_REQ bit description.
SerDes PLLa Control Register 0 (PLL1CR0 - PLL2CR0)	Updated the PLL1CR0/PLL2CR0[REFCLK_SEL, FRATE_SEL, DLYDIV_SEL] bit descriptions.
SerDes PLLa Control Register 1 (PLL1CR1 - PLL2CR1)	Updated the bit 21 as reserved.
SerDes PLLa Control Register 5 (PLL1CR5 - PLL2CR5)	Updated the SEL_REFCLK_AMP_DIS bit description.
SerDes Transmit Calibration Control Register 1 (TCALCR1)	Updated the ANA_OUT_REFCLK_EN and DIG_OUT_REFCLK_EN bit descriptions.
Lane a Protocol Select Status Register 0 (LNAPSSR0 - LNDPSSR0)	Updated the TYPE bit description for bit 6-7. Updated the LANE bit width from 2-0 to 3-0.
Protocol Configuration Register 0 (PCCR0)	Updated the PEXA_CFG and PEXB_CFG bit descriptions.
General Control Register 0 - Lane a (LNAGCR0 - LNDGCR0)	Updated the LNaGCR0[RRAT_SEL, RRST_B, IF20BIT_EN, PROTS] bit descriptions.
Speed Switch Control Register 0 - Lane a (LNASSCR0 - LNDSSCR0)	Removed settings for Others. Added the ISLEW_RCTL_0 and OSLEW_RCTL_0 bits. Updated the LNaSSCR0[RXEQ_BST_0, BASE_WAND_0] bit descriptions
Receive Equalization Control Register 0 - Lane a (LNARECR0 - LNDRECR0)	Updated the LNARECR0[RXEQ_BST, GK2OVD, GK3OVD, GK2OVD_EN, GK3OVD_EN, BASE_WAND] bit descriptions.
Transmit Equalization Control Register 0 - Lane a (LNATECR0 - LNDTECR0)	Updated the AMP_RED bit description.
Speed Switch Control Register 1- Lane 0 (LNASSCR1 - LNDSSCR1)	Removed settings for Others. Added ISLEW_RCTL_1 and OSLEW_RCTL_1 bits.
SGMIIa Protocol Control Register 1 (SGMIIACR1 - SGMIIDCR1) QSGMIIa Protocol Control Register 1 (QSGMIIACR1 - QSGMIIBCR1)	Updated the MDEV_PORT bit description.
SGMII Extended Status (C45_SGMII_XTND_STAT)	Added this register.
SGMII Extended Status (SGMII_XTND_STAT)	Added this register.
QSGMII Extended Status (QSGMII_XTND_STAT)	Added this register.
1000Base-KX	Updated step 4 for 1000BaseKX initialization.
XFI/10GBase-KR	Updated the first paragraph.
Table 23-1	Updated EC1 RGMII (MAC5) to EC2 RGMII (MAC5).
Valid Reference Clocks and PLL Configurations for SerDes Protocols	Updated the note for spread-spectrum reference clock.

F.1.24 Serial Peripheral Interface (SPI) Revision History

Reference	Description
Throughout	Updated the slave mode feature support.
SPI register descriptions	Updated the note in the memory map description.
Status Register (SR)	Updated the SR[TFFF] bit description. Added the CMDFFF bit.
Status Register Extended (SREX)	Swapped the bit 11 (RXCTR4) and bit 14 (TXCTR4).
Draining the TX FIFO	Removed the second paragraph starting with "If an external bus master initiates.....".
Classic SPI transfer format (CPHA = 0)	Updated SS to SS_B.
Power saving features	Added the following sentence: All power-saving features require a logic external to SPI.
Delay settings	Added the bit name (STAR[PDT]) in delay scalar values.

F.1.25 Trust Zone Modules Revision History

Reference	Description
Trust Zone Modules	Separated this chapter from the Trust Architecture 3.0 chapter.
Overview	Added this section.
TZC-400 overview	Updated the TZC-400 overview diagram to remove unsupported filter units.
• Compliance • Interfaces • TZC-400 interfaces • Clock and reset signals • Reset signals • Identity input signals • Fast Path Identity inputs • Interrupt signal • Revision and configuration signal • Lock transaction sequences • Access latencies	Removed these sections.
About the TZC-400	Removed the second paragraph.
TZC400 register descriptions	Added bit settings for each register in the memory map. Added register descriptions for PIDn and CIDn registers. Removed the PID4-7 registers. Removed the Region ID access registers. Removed the Fail ID register.
Access filtering with regions	Updated this section.
Region setup	Removed row for Non-secure ID filtering.
Speculative accesses	Removed paragraph starting with "An example of where....".

Table continues on the next page...

Substantive changes from revision 0 to revision 1

Reference	Description
Denied ACE-Lite transactions	Removed note starting with "For failed exclusive read accesses.....". Removed the paragraph starting with "Even though the TZC400 only has one interrupt....".
TZC-400 overview	Removed the sentence starting with: "The filter unit operates in its own clock..."
Features	Removed the following bullet: "Identity-based filtering of non-secure accesses."
Region 0 Top Address Low Register (REGION_TOP_LOW_0)	Added the HIGH_4KB bit.
Access permissions to regions Reset	Updated these sections.
Barriers	Removed the note.
Changing TZC-400 region settings on an active system	Updated this section.
Fast path selection	Updated this section.
Region access rules	Removed the following bullet: "Regions 1 and higher can have address regions that overlap with each other, but only if they are set for different filters."
Gate keeper	Removed the note.
TZPC register descriptions	Corrected the register access in the TZPC memory map. Updated the bit definition of each register. Removed the following registers: TZDECPROT2_Stat, TZDECPROT2_Set, and TZDECPROT2_Clr
About the TrustZone Protection Controller	Updated the second paragraph. Removed the last two bullets for AMBA APB feature support.
TZPC Functional description	Updated this section.
Secure RAM Region Size Register (TZPCR0SIZE) TZDECPROT Register 0 Stat (TZDECPROT0_Stat) TZDECPROT Register 0 Set (TZDECPROT0_Set)	Updated the register function.
TZPC Identification Register 0 (TZPCPCELLID0) TZPC Identification Register 1 (TZPCPCELLID1) TZPC Identification Register 2 (TZPCPCELLID2) TZPC Identification Register 3 (TZPCPCELLID3) TZPC Peripheral Identification Register 0 (TZPCPERIPHID0) TZPC Peripheral Identification Register 1 (TZPCPERIPHID1) TZPC Peripheral Identification Register 2 (TZPCPERIPHID2) TZPC Peripheral Identification Register 3 (TZPCPERIPHID3)	Removed note from TZPCPCELLIDn and TZPCPERIPHIDn registers. Updated the register description of TZPCPERIPHIDn and TZPCPCELLIDn registers.
About the AXI TrustZone memory adapter	Updated the second paragraph.
Preliminary material	Removed this section under TZPC and TZMA.

F.1.26 Trust Architecture 3.0 Revision History

Reference	Description
Secure Boot and Trust Architecture	Removed the existing content and added content for the overview of the Secure Boot and Trust Architecture.
TZC-400, TZPC, and TZMA	Moved these sections to Trust Zone Modules chapter.

F.1.27 TMU Revision History

Reference	Description
The TMU module as implemented on the chip	Added the note.
Throughout	Updated the temperature measurement range to 0-110C.
Initialization Information	Moved this section to the beginning and updated this section to remove temperature points exceeding 110C. Added step for programing TTRnCR registers.

F.1.28 Universal Serial Bus 3.0 Revision History

Reference	Description
External Signals	Updated USB1_VBUS as an Input.
USB3.0 register descriptions	Updated the base address of USB1 and USB2 controllers.
USB PHY SuperSpeed register descriptions	Updated the base address of USBPHY1 and USBPHY2 controllers. Updated the offsets of the MPLL_LOOP_CTL and LANE0_RX_OVRD_IN_HI registers.
Table 28-2	Added the following note: The default value of ref_clk is 125 MHz. Refer to Reset, Clocking, and Initialization chapter for chip-specific value of ref_clk.
Device status register (DSTS)	Updated the bit description for bit 28 as follows: Save/Restore error. This bit is currently not supported.
USB 3.0 Controller and PHY Configuration	Added the following sentence: USB PHY registers should not be accessed during the PBI phase before the system is ready.

F.1.29 Watchdog Timer Unit Revision History

Reference	Description
WDOG register descriptions	Updated the endianness of WDOG registers from big endian to little endian. Updated the base address.
Watchdog Value Register (WDOGVALUE)	Updated the register access from RO to RW.

F.1.30 LS1044A Revision History

Reference	Description
LS1044A SerDes	Added the following sentence: It supports 1x 10GE and use SerDes 1 lane 0 (lane D).
LS1044A Register Differences from LS1088A	Removed the SVR table and added the SVR values in the SVR and PCI Device IDs table.

F.1.31 LS1048A Revision History

Reference	Description
LS1048A SerDes	Added the following sentence: It supports 1x 10GE and use SerDes 1 lane 0 (lane D).
LS1048A Register Differences from LS1088A	Removed the SVR table and added the SVR values in the SVR and PCI Device IDs table.

F.1.32 LS1084A Revision History

Reference	Description
LS1084A Register Differences from LS1088A	Removed the SVR table and added the SVR values in the SVR and PCI Device IDs table.

F.1.33 EPU Revision History

Reference	Description
EPU register descriptions	Added this appendix.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018–2020 NXP B.V.

Document Number LS1088ARM
Revision 1, 11/2020