XT32H05X Series

Reference manual

Rev 1.0.0

Original Release Date : 15-Mar-2023
Revised : 1-Jun-2023

Revision History

Release	Date	Summary of Change
V0.0.0	15/03/2023	Initial version
V1.0.0	01/06/2023	Update for XT32H05X

Contents

XT32H0 SERIES ..... **I**

REVISION HISTORY ..... **I**

CONTENTS ..... **II**

LIST OF FIGURES ..... **I**

LIST OF TABLES ..... **III**

1 INTRODUCTION **1**

1.1 GENERAL INFORMATION 1
1.2 DEVICE SUMMARY 1
1.3 FEATURES AND PERIPHERALS 1
1.4 GLOSSARY 4

2 MEMORY AND BUS ARCHITECTURE **5**

2.1 SYSTEM ARCHITECTURE 5
2.2 MEMORY 8
2.2.1 Introduction *8*
2.2.2 Memory mapping *8*

3 RESET AND OPERATION MODE **11**

3.1 RESET 11
3.1.1 System resets and sources *11*
3.2 BOOT 12
3.2.1 Boot modes *12*
3.2.2 Boot sequence *12*
3.3 OPERATION MODE 13

4 FLASH CONTROLLER (FLASH) **15**

4.1 INTRODUCTION 15
4.2 FLASH MAIN FEATURES 15
4.3 FLASH OPERATION DESCRIPTION 15
4.3.1 Read Operation *15*
4.3.2 Programming Operation *16*
4.3.3 Page Erase Operation *18*
4.3.4 Mass/Chip Erase Operation *18*
4.3.5 Configure operation time *19*
4.4 FLASH REGISTERS 20
4.4.1 FLASH_CTRL(Flash control register) *20*
4.4.2 FLASH_SMW_SET_OPTION *21*

4.4.3 FLASH_SMW_SET_HEMWHV 22
4.4.4 FLASH_SMW_SET_TIMER 22
4.4.5 FLASH_NSMW_SET_WHV 23
4.4.6 Flash register map 24

5 INTERNAL CLOCK GENERATOR (ICG) 25

5.1 INTRODUCTION 25
5.2 ICG MAIN FEATURES 26
5.3 ICG FUNCTIONAL DESCRIPTION 27
5.3.1 Clock Sources 27
5.3.2 High speed clock setting 27
ICG_TIMER_ADV_CLK_DIV 29
5.3.3 High speed RC oscillator clock calibration 29
5.4 ICG REGISTERS 30
5.4.1 ICG_CLK_CFG 30
5.4.2 ICG_HCLK_DIV 32
5.4.3 ICG_PCLK_DIV 32
5.4.4 ICG_DMA_CLK_DIV 32
5.4.5 ICG_BR_CLK_DIV 33
5.4.6 ICG_I2C1_CLK_DIV 34
5.4.7 ICG_I2C2_CLK_DIV 34
5.4.8 ICG_GPIO_CLK_DIV 34
5.4.9 ICG_RC16M_CTRL 35
5.4.10 ICG_PLL_DIV 35
5.4.11 ICG_PLL_PD_CTRL 40
5.4.12 ICG_PLL_LFCFG_CTRL 41
5.4.13 ICG_PLL_VCOCAL_CTRL 41
5.4.14 ICG_PLL_IREF_CTRL 42
5.4.15 ICG_TIMER_ADV_CLK_DIV 43
5.4.16 ICG_TIMER_B1_CLK_DIV 43
5.4.17 ICG_TIMER_B2_CLK_DIV 43
5.4.18 ICG_WDT_W_CLK_DIV 44
5.4.19 ICG_TIMER_EXCLK_ENABLE_1 44
5.4.20 ICG_TIMER_EXCLK_ENABLE_2 46
5.4.21 ICG_ADC_CLK_DIV 48
5.4.22 ICG register map 49

6 ALWAYS ON BLOCK (AON) 52

6.1 INTRODUCTION 52
6.2 AON MAIN FEATURES 52
6.3 PMU 52

6.3.1 Programmable Voltage Detector (PVD) 52
6.3.2 PMU 52
6.3.3 Low speed clock 54
6.4 AON FUNCTIONAL DESCRIPTION 54
6.4.1 Generate clock 54
6.4.2 Deep sleep mode 55
6.4.3 Low speed RC oscillator calibration 55
6.5 AON REGISTERS 56
6.5.1 AON_CFG 56
6.5.2 AON_RTC_REFCLK_DIV 56
6.5.3 AON_CLK_CTRL 57
6.5.4 AON_LDO_CFG 57
6.5.5 AON_IVREF_CFG_1 60
6.5.6 AON_IVREF_CFG_2 61
6.5.7 AON_IVREF_CFG_3 61
6.5.8 AON_DEEPSLEEP_TIME 62
6.5.9 AON_DEEPSLEEP_CTRL 62
6.5.10 AON register map 63

7 GENERAL-PURPOSE IOS (GPIO) 65

7.1 INTRODUCTION 65
7.2 GPIO SIGNAL DESCRIPTIONS 65
7.3 GPIO MAIN FEATURES 65
7.4 GPIO FUNCTIONAL DESCRIPTION 66
7.4.1 Software source outputs 66
7.4.2 Software source inputs 66
7.4.3 Hardware source input and outputs 66
7.4.4 Interrupt sources 66
7.5 GPIO REGISTERS 67
7.5.1 GPIOx_DR 67
7.5.2 GPIOx_DDR 67
7.5.3 GPIOx_CTRL 68
7.5.4 GPIOA_INTEN 68
7.5.5 GPIOA_INTMASK 68
7.5.6 GPIOA_INTSNS 69
7.5.7 GPIOA_INTPOL 69
7.5.8 GPIOA_INTSTAT 69
7.5.9 GPIOA_INTRSTAT 69
7.5.10 GPIOA_DEBOUNCE 70
7.5.11 GPIOA_EOI 70
7.5.12 GPIOx_EXTP 70

7.5.13 GPIO_SYNC 71
7.5.14 GPIO register map 72

8 SYSTEM CONFIGURATION CONTROLLER (SYSCFG) 75

8.1 INTRODUCTION 75
8.2 SYSCFG MAIN FEATURES 75
8.3 SYSCFG FUNCTIONAL DESCRIPTION 75
8.3.1 Interrupt assignment 75
8.3.2 DMA handshaking interface assignment 76
8.3.3 GPIO auxiliary ports 76
8.3.4 Peripheral software reset 76
8.3.5 UART baudrate auto detection 77
8.3.6 Hardware divider 77
8.3.7 RC oscillator Auto Frequency Calibration (AFC) 77
8.3.8 Inter-communication between main core and accelerator 80
8.3.9 Timers port configuration 80
8.4 SYSCFG REGISTERS 81
8.4.1 SYS_ROM_CFG 81
8.4.2 SYS_PVD_INT_CFG 81
8.4.3 SYS_SRAM_CTRL_1 82
8.4.4 SYS_SRAM_CTRL_PD 83
8.4.5 SYS_CFG 84
8.4.6 SYS_IRQ_CFG_1 86
8.4.7 SYS_IRQ_CFG_2 86
8.4.8 SYS_DMA_CFG 89
8.4.9 SYS_WDT_CFG 91
8.4.10 SYS_SOFT_RESET_CFG 93
8.4.11 SYS_DMA_CFG_2 95
8.4.12 SYS_TIMER_ADV_CFG 95
8.4.13 SYS_GPIO_ENAUX_A 95
8.4.14 SYS_GPIO_ENAUX_B 96
8.4.15 SYS_GPIO_ENAUX_C 97
8.4.16 SYS_GPIO_ENAUX_D 98
8.4.17 SYS_UID_REG_1 98
8.4.18 SYS_UID_REG_2 98
8.4.19 SYS_UID_REG_3 99
8.4.20 SYS_UART_ADCFG_REG 99
8.4.21 SYS_UART_ADIF_REG 101
8.4.22 SYS_UART1_ADBAUD_REG 101
8.4.23 SYS_UART2_ADBAUD_REG 101
8.4.24 SYS_UART3_ADBAUD_REG 102

8.4.25 SYS_UART4_ADBAUD_REG 102
8.4.26 SYS_CTSM_CTRL 102
8.4.27 SYS_CTSM_INT_HOST2CTSM 103
8.4.28 SYS_CTSM_INT_CTSM2HOST 103
8.4.29 SYS_HDIV_A 104
8.4.30 SYS_HDIV_B 104
8.4.31 SYS_HDIV_Q 104
8.4.32 SYS_HDIV_R 104
8.4.33 SYS_AFC_CTRL 105
8.4.34 SYS_AFC_CFG 106
8.4.35 SYS_AFC_RES 106
8.4.36 SYS_TIM_A1_IMX 107
8.4.37 SYS_TIM_A2_IMX 108
8.4.38 SYS_TIM_G1_IMX 109
8.4.39 SYS_TIM_G2_IMX 111
8.4.40 SYS_TIM_G3_IMX 112
8.4.41 SYS_TIM_G4_IMX 113
8.4.42 SYS_TIM_CMS_OC 115
8.4.43 SYSCFG register map 117

9 PORT MUX (PADI) 123

9.1 INTRODUCTION 123
9.2 PADI MAIN FEATURES 131
9.3 PADI FUNCTIONAL DESCRIPTION 131
9.3.1 Configure IO 131
9.4 PADI REGISTERS 132
9.4.1 PADI_PORTMUX_CFG_1 132
9.4.2 PADI_PORTMUX_CFG_2 132
9.4.3 PADI_PORTMUX_CFG_3 133
9.4.4 PADI_PORTMUX_CFG_4 133
9.4.5 PADI_PORTMUX_CFG_5 134
9.4.6 PADI_PORTMUX_CFG_6 134
9.4.7 PADI_PORTMUX_CFG_7 135
9.4.8 PADI_PORTMUX_CFG_8 135
9.4.9 PADI_IO_PULLUP_CFG_1 136
9.4.10 PADI_IO_PULLUP_CFG_2 136
9.4.11 PADI_IO_PULLDOWN_CFG_1 136
9.4.12 PADI_IO_PULLDOWN_CFG_2 137
9.4.13 PADI_IO_DS_CFG_1 137
9.4.14 PADI_IO_DS_CFG_2 137
9.4.15 PADI register map 138

10 DIRECT MEMORY ACCESS CONTROLLER (DMA) 140

10.1 INTRODUCTION 140
10.2 DMA MAIN FEATURES 141
10.3 DMA FUNCTIONAL DESCRIPTION 141
10.3.1 DMA Transfer 141
10.3.2 DMA arbitration 141
10.3.3 DMA channels 141
10.3.4 Handshaking interface 142
10.3.5 Register access 143
10.3.6 Data Transfer programming flow 143
10.3.7 Abort transaction 147
10.4 DMA REGISTERS 149
10.4.1 DMA_DMACFGREG 149
10.4.2 DMA_CHENREG 149
10.4.3 DMA_SARx 150
10.4.4 DMA_DARx 150
10.4.5 DMA_CTLx_L 151
10.4.6 DMA_CTLx_H 152
10.4.7 DMA_CFGx_L 153
10.4.8 DMA_CFGx_H 154
10.4.9 DMA_RAWTFR 155
10.4.10 DMA_RAWBLOCK 155
10.4.11 DMA_RAWSRCTRAN 156
10.4.12 DMA_RAWDSTTRAN 156
10.4.13 DMA_RAWERR 156
10.4.14 DMA_STATUSTFR 156
10.4.15 DMA_STATUSBLOCK 157
10.4.16 DMA_STATUSSRCTRAN 157
10.4.17 DMA_STATUSDSTTRAN 157
10.4.18 DMA_STATUSERR 158
10.4.19 DMA_MASKTFR 158
10.4.20 DMA_MASKBLOCK 158
10.4.21 DMA_MASKSRCTRAN 159
10.4.22 DMA_MASKDSTTRAN 159
10.4.23 DMA_MASKERR 159
10.4.24 DMA_CLEARTFR 160
10.4.25 DMA_CLEARBLOCK 160
10.4.26 DMA_CLEARSRCTRAN 161
10.4.27 DMA_CLEARDSTTRAN 161
10.4.28 DMA_CLEARERR 161
10.4.29 DMA_STATUSINT 162

10.4.30 DMA_REQSRC 162
10.4.31 DMA_REQDST 163
10.4.32 DMA_SGLREQSRC 163
10.4.33 DMA_SGLREQDST 164
10.4.34 DMA_LSTREQSRC 164
10.4.35 DMA_LSTREQDST 165
10.4.36 DMA register map 166

11 NESTED VECTORED INTERRUPT CONTROLLER (NVIC) 174

11.1 NVIC MAIN FEATURES 174
11.2 INTERRUPT AND EXCEPTION VECTORS 174

12 INTERRUPT CONTROLLER (ICTL) 176

12.1 INTRODUCTION 176
12.2 ICTL MAIN FEATURES 176
12.3 ICTL FUNCTIONAL DESCRIPTION 176
12.4 ICTL REGISTERS 178
12.4.1 ICTL_IRQ_INTEN_L 178
12.4.2 ICTL_IRQ_INTEN_H 178
12.4.3 ICTL_IRQ_INTMASK_L 178
12.4.4 ICTL_IRQ_INTMASK_H 179
12.4.5 ICTL_IRQ_INTFORCE_L 179
12.4.6 ICTL_IRQ_INTFORCE_H 179
12.4.7 ICTL_IRQ_RAWSTATUS_L 180
12.4.8 ICTL_IRQ_RAWSTATUS_H 180
12.4.9 ICTL_IRQ_STATUS_L 180
12.4.10 ICTL_IRQ_STATUS_H 180
12.4.11 ICTL_IRQ_MASKSTATUS_L 181
12.4.12 ICTL_IRQ_MASKSTATUS_H 181
12.4.13 ICTL_IRQ_FINALSTATUS_L 181
12.4.14 ICTL_IRQ_FINALSTATUS_H 181
12.4.15 ICTL_FIQ_INTEN 181
12.4.16 ICTL_FIQ_INTMASK 182
12.4.17 ICTL_FIQ_INTFORCE 182
12.4.18 ICTL_FIQ_RAWSTATUS 183
12.4.19 ICTL_FIQ_STATUS 183
12.4.20 ICTL_FIQ_FINALSTATUS 183
12.4.21 ICTL_IRQ_PLEVEL 183
12.4.22 ICTL register map 184

13 CYCLIC REDUNDANCY CHECK CALCULATION (CRC) 187

13.1 INTRODUCTION 187

13.2 CRC MAIN FEATURES 187
13.3 CRC FUNCTIONAL DESCRIPTION 187
13.4 CRC REGISTERS 189
13.4.1 CRC_CFG 189
13.4.2 CRC_DATA 189
13.4.3 CRC_RESULT 190
13.4.4 CRC_XOR_VALUE 190
13.4.5 CRC register map 191

14 MOTOR DRIVER UNIT (MDU) 192

14.1 INTRODUCTION 192
14.2 MDU MAIN FEATURES 192
14.3 MDU SIGNAL DESCRIPTIONS 192
14.4 MDU FUNCTIONAL DESCRIPTION 194
14.4.1 ADC 194
14.4.2 Comparator 209
14.4.3 PGA 209
14.4.4 DAC 210
14.5 MDU REGISTERS 210
14.5.1 MDU_PDCTRL 210
14.5.2 MDU_DAC_CFG 211
14.5.3 MDU_PGA_CFG 213
14.5.4 MDU_ADC_CFG 214
14.5.5 MDU_COMP_CFG1 216
14.5.6 MDU_COMP_CFG2 217
14.5.7 MDU_COMP_CFG3 218
14.5.8 MDU_ADC_CALIB 220
14.5.9 MDU_DMA_CFG 221
14.5.10 MDU_MDU_CFG 221
14.5.11 MDU_INJGRP_TRG 224
14.5.12 MDU_REGGRP_ TRG 226
14.5.13 MDU_REGGRP_TRGCFG 227
14.5.14 MDU_INJGRP_CHCR_1 229
14.5.15 MDU_INJGRP_CHCR_2 230
14.5.16 MDU_INJGRP_CHCR_3 231
14.5.17 MDU_INJGRP_CHCR_4 232
14.5.18 MDU_REGGRP_CHMAP 233
14.5.19 MDU_REGGRP_SGRP_CHMAP_1 233
14.5.20 MDU_REGGRP_SGRP_CHMAP_2 233
14.5.21 MDU_REGGRP_SGRP_CHMAP_3 234
14.5.22 MDU_REGGRP_SGRP_CHMAP_4 234

14.5.23 MDU_REGGRP_SGRP_CHMAP_5 234
14.5.24 MDU_REGGRP_SGRP_CHMAP_6 235
14.5.25 MDU_REGGRP_SGRP_CHMAP_7 235
14.5.26 MDU_REGGRP_SGRP_CHMAP_8 235
14.5.27 MDU_REGGRP_CHCR_1 236
14.5.28 MDU_REGGRP_CHCR_2 237
14.5.29 MDU_REGGRP_CHCR_3 238
14.5.30 MDU_REGGRP_CHCR_4 239
14.5.31 MDU_REGGRP_CHCR_5 240
14.5.32 MDU_REGGRP_CHCR_6 241
14.5.33 MDU_REGGRP_CHCR_7 242
14.5.34 MDU_REGGRP_CHCR_8 243
14.5.35 MDU_INT_STATUS_1 244
14.5.36 MDU_INT_STATUS_2 246
14.5.37 MDU_INT_STATUS_3 246
14.5.38 MDU_INJ_RESULT_1 247
14.5.39 MDU_INJ_RESULT_2 247
14.5.40 MDU_REG_RESULT_1 247
14.5.41 MDU_REG_RESULT_2 247
14.5.42 MDU_REG_RESULT_3 248
14.5.43 MDU_REG_RESULT_4 248
14.5.44 MDU_REG_RESULT_5 248
14.5.45 MDU_REG_RESULT_6 248
14.5.46 MDU_REG_RESULT_7 249
14.5.47 MDU_REG_RESULT_8 249
14.5.48 MDU_REG_RESULT_9 249
14.5.49 MDU_REG_RESULT_10 249
14.5.50 MDU_REG_RESULT_11 250
14.5.51 MDU_REG_RESULT_12 250
14.5.52 MDU_REG_RESULT_13 250
14.5.53 MDU_REG_RESULT_14 250
14.5.54 MDU_REG_RESULT_15 251
14.5.55 MDU_REG_RESULT_16 251
14.5.56 MDU_AWD_CR_1 251
14.5.57 MDU_AWD_CR_2 252
14.5.58 MDU_AWD_CR_3 252
14.5.59 MDU_INT_STATUS_4 253
14.5.60 MDU register map 254

15 ADVANCED TIMER (TIMA) 260

15.1 INTRODUCTION 260

15.2 TIMA MAIN FEATURES 261
15.3 TIMA SIGNAL DESCRIPTIONS 261
15.4 TIMA FUNCTION DESCRIPTION 262
15.4.1 Basic Timer 262
15.4.2 Slave mode configurate 262
15.4.3 Master mode configurate 262
15.4.4 Break configurate 262
15.4.5 Output mode 262
15.4.6 Input mode 263
15.4.7 UIF bit remapping 265
15.4.8 Timer synchronization 265
15.4.9 ADC synchronization 265
15.4.10 Clock source selection 266
15.4.11 DMA control 266
15.4.12 Motor control 266
15.1 TIMA REGISTERS 267
15.1.1 TIM_CTRL1 (Timer control register 1) 267
15.1.2 TIM_CTRL2 (Timer control register 2) 268
15.1.3 TIM_SLVMD (Timer slave mode control register) 271
15.1.4 TIM_REQEN (Timer DMA/interrupt enable register) 274
15.1.5 TIM_STATUS (Timer status register) 276
15.1.6 TIM_EVT (Timer event generation register) 277
15.1.7 TIM_CHMD1 (Timer compare mode register 1 - output) 278
15.1.8 TIM_CHMD1 (Timer capture mode register 1 - input) 280
15.1.9 TIM_CHMD2 (Timer compare mode register 2 - output) 281
15.1.10 TIM_CHMD2 (Timer capture mode register 2- input) 281
15.1.11 TIM_CHEP (Timer capture/compare enable register) 282
15.1.12 TIM_CNTR (Timer counter) 283
15.1.13 TIM_PSCR (Timer prescaler) 284
15.1.14 TIM_PRD (Timer auto-load register) 284
15.1.15 TIM_REP (Timer repetition counter register) 284
15.1.16 TIM_CHR1 (Timer capture/compare register 1) 284
15.1.17 TIM_CHR2 (Timer capture/compare register 2) 285
15.1.18 TIM_CHR3 (Timer capture/compare register 3) 285
15.1.19 TIM_CHR4 (Timer capture/compare register 4) 285
15.1.20 TIM_BKDT (Timer break and dead-time register) 285
15.1.21 TIM_DMAC (Timer DMA control register) 287
15.1.22 TIM_DMAI (Timer DMA information register) 288
15.1.23 TIM_AUX (Timer auxiliary register) 288
15.1.24 TIM_CHMD3 (Timer compare mode register 3 - output) 288
15.1.25 TIM_CHR5 (Timer capture/compare register 5) 289

15.1.26 TIM_CHR6 (Timer capture/compare register 6) 289
15.1.27 TIM_BRK (Timer break control register) 290
15.1.28 TIM_BRK2 (Timer break control register 2) 291
15.1.29 TIM_ISCLT (Timer input selection register) 292
15.1.30 TIM_DTR1 (Timer dead-time register 1) 292
15.1.31 TIM_DTR2 (Timer dead-time register 2) 294
15.1.32 TIMA register map 295

16 GENERAL-PURPOSE TIMER (TIMG) 298

16.1 INTRODUCTION 298
16.2 TIMG MAIN FEATURES 299
16.3 TIMG SIGNAL DESCRIPTIONS 299
16.4 TIMG FUNCTIONAL DESCRIPTION 300
16.5 TIMG REGISTERS 301
16.5.1 TIM_CTRL1 301
16.5.2 TIM_CTRL2 302
16.5.3 TIM_SLVMD 304
16.5.4 TIM_REQEN 305
16.5.5 TIM_STATUS 306
16.5.6 TIM_EVT 307
16.5.7 TIM_CHMD1 (Output) 308
16.5.8 TIM_CHMD1 (Input) 310
16.5.9 TIM_CHMD2 (Output) 311
16.5.10 TIM_CHMD2 (Input) 311
16.5.11 TIM_CHEP 312
16.5.12 TIM_CNTR 313
16.5.13 TIM_PSCR 314
16.5.14 TIM_PRD 314
16.5.15 TIM_REP 314
16.5.16 TIM_CHR1 314
16.5.17 TIM_CHR2 315
16.5.18 TIM_CHR3 315
16.5.19 TIM_CHR4 315
16.5.20 TIM_BKDT 316
16.5.21 TIM_AUX 316
16.5.22 TIM_CHMD3 317
16.5.23 TIM_CHR5 317
16.5.24 TIM_CHR6 318
16.5.25 TIM_BRK 318
16.5.26 TIM_ISCLT 319
16.5.27 TIMG register map 320

17 BASIC TIMERS (TIMB) 322

17.1 INTRODUCTION 322
17.2 TIMB MAIN FEATURES 322
17.3 TIMB SIGNAL DESCRIPTIONS 323
17.4 TIMB FUNCTIONAL DESCRIPTION 323
17.4.1 Usage flow 324
17.5 TIMB REGISTERS 325
17.5.1 TIMERx_LDR 325
17.5.2 TIMERx_CCVR 325
17.5.3 TIMERx_CCR 326
17.5.4 TIMERx_EOI 326
17.5.5 TIMERx_STAT 327
17.5.6 TIMERsINTSTAT 327
17.5.7 TIMERsEOI 328
17.5.8 TIMERsRAWINTSTAT 328
17.5.9 TIMB register map 329

18 INDEPENDENT WATCHDOG (WDTI) 331

18.1 INTRODUCTION 331
18.2 WDTI MAIN FEATURES 331
18.3 WDTI FUNCTIONAL DESCRIPTION 331
18.4 WDTI REGISTERS 332
18.4.1 WDT_CR(WDT control register) 332
18.4.2 WDT_TORR(WDT timeout period register) 332
18.4.3 WDT_CCVR(WDT current counter value register) 333
18.4.4 WDT_CRR(WDT restart register) 333
18.4.5 WDTI register map 334

19 WINDOWS WATCHDOG (WDTW) 335

19.1 INTRODUCTION 335
19.2 WDTW MAIN FEATURES 335
19.3 WDTW FUNCTIONAL DESCRIPTION 335
19.4 WDTW REGISTERS 336
19.4.1 WDT_CR(WDT control register) 336
19.4.2 WDT_TORR(WDT timeout period register) 336
19.4.3 WDT_CCVR(WDT current counter value register) 337
19.4.4 WDT_CRR(WDT restart counter register) 337
19.4.5 WDT_STAT(WDT status register) 337
19.4.6 WDT_EOI(WDT clear interrupt register) 338
19.4.7 WDTW register map 339

20 REAL-TIMER CLOCK (RTC) 340

20.1 INTRODUCTION 340
20.2 RTC MAIN FEATURES 340
20.3 RTC FUNCTIONAL DESCRIPTION 340
20.3.1 Configure RTC 340
20.3.2 Set Alarm 341
20.4 RTC REGISTER 342
20.4.1 RTC_CCVR(RTC current counter value register) 342
20.4.2 RTC_CMR(RTC counter match register) 342
20.4.3 RTC_CLR 342
20.4.4 RTC_CCR 342
20.4.5 RTC_STAT 343
20.4.6 RTC_RSTAT 343
20.4.7 RTC_EOI 344
20.4.8 RTC register map 344

21 INTER-INTEGRATED CIRCUIT (I2C) 345

21.1 INTRODUCTION 345
21.2 I2C MAIN FEATURES 345
21.3 I2C SIGNAL DESCRIPTIONS 345
21.4 I2C FUNCTIONAL DESCRIPTION 346
21.4.1 Operation Mode 346
21.4.2 Communication flow 346
21.4.3 Register access 347
21.4.4 Operation mode setting 347
21.4.5 Master data transfer 347
21.4.6 Slave data transfer 350
21.4.7 Clock configuration 353
21.4.8 DMA transfer 355
21.5 I2C REGISTERS 357
21.5.1 I2C_CON (I2C control register) 357
21.5.2 I2C_TAR (I2C target address register) 358
21.5.3 I2C_SAR (I2C slave address register) 359
21.5.4 I2C_HS_MADDR (I2C high speed master mode code address register) 359
21.5.5 I2C_DATA_CMD (I2C Rx/Tx data buffer and command register) 359
21.5.6 I2C_SS_SCL_HCNT (Standard Speed Clock SCL High Count Register) 360
21.5.7 I2C_SS_SCL_LCNT (Standard Speed Clock SCL Low Count Register) 360
21.5.8 I2C_FS_SCL_HCNT (Fast or Fast Plus Mode Clock SCL High Count Register) 361
21.5.9 I2C_FS_SCL_LCNT (Fast or Fast Plus Mode Clock SCL Low Count Register) 361
21.5.10 I2C_HS_SCL_HCNT (High Speed Clock SCL High Count Register) 362
21.5.11 I2C_HS_SCL_LCNT (High Speed Clock SCL Low Count Register) 362
21.5.12 I2C_INT_STAT (I2C Interrupt Status Register) 362

21.5.13 I2C_INT_MASK (I2C Interrupt Mask Register) 364
21.5.14 I2C_INT_RAWSTAT (I2C raw interrupt status register) 365
21.5.15 I2C_RX_TL (I2C receive FIFO threshold register) 366
21.5.16 I2C_TX_TL (I2C transmit FIFO threshold register) 367
21.5.17 I2C_INT_CLR (I2C clear interrupt register) 367
21.5.18 I2C_INT_CLR_RXU (I2C clear RX_UNDER interrupt register) 367
21.5.19 I2C_INT_CLR_RXO (I2C clear RX_OVER interrupt register) 368
21.5.20 I2C_INT_CLR_TXO (I2C clear TX_OVER interrupt register) 368
21.5.21 I2C_INT_CLR_RDR (I2C clear RD_REQ interrupt register) 368
21.5.22 I2C_INT_CLR_TXA (I2C clear TX_ABRT interrupt register) 369
21.5.23 I2C_INT_CLR_RXD (I2C clear RX_DONE interrupt register) 369
21.5.24 I2C_INT_CLR_ACT (I2C clear ACTIVITY interrupt register) 369
21.5.25 I2C_INT_CLR_STOP_DET (I2C clear STOP_DET interrupt register) 370
21.5.26 I2C_INT_CLR_START_DET (I2C clear START_DET interrupt register) 370
21.5.27 I2C_INT_CLR_GCALL (I2C clear GEN_CALL interrupt register) 370
21.5.28 I2C_ENABLE (I2C enable register) 370
21.5.29 I2C_STATUS (I2C status register) 372
21.5.30 I2C_TX_FL (I2C transmit FIFO level register) 373
21.5.31 I2C_RX_FL (I2C receive FIFO level register) 373
21.5.32 I2C_SDA_HOLD (I2C SDA hold time length register) 373
21.5.33 I2C_TX_ABRT_SRC (I2C transmit abort source register) 374
21.5.34 I2C_DMA_CR (DMA control register) 375
21.5.35 I2C_DMA_TDLR (DMA transmit data level register) 375
21.5.36 I2C_DMA_RDLR (DMA receive data level register) 376
21.5.37 I2C_SDA_SETUP (I2C SDA Setup Register) 376
21.5.38 I2C_ACK_GENERAL_CALL (I2C ACK general call register) 376
21.5.39 I2C_ENABLE_STATUS (I2C enable status register) 377
21.5.40 I2C_FS_SPKLEN (I2C SS and FS spike suppression limit register) 377
21.5.41 I2C_HS_SPKLEN (I2C HS spike suppression limit register) 378
21.5.42 I2C register map 379

22 UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER (UART) 383

22.1 INTRODUCTION 383
22.2 UART MAIN FEATURES 384
22.3 UART SIGNAL DESCRIPTIONS 385
22.4 UART FUNCTIONAL DESCRIPTIONS 385
22.4.1 Register access 385
22.4.2 Calculate baudrate 385
22.4.3 UART Transmit operation in Polling mode 385
22.4.4 UART Transmit operation in Interrupt mode 388
22.4.5 UART Transmit operation in DMA mode 390

22.4.6 UART Receive operation in Polling mode 392
22.4.7 UART Receive operation in Interrupt mode 394
22.4.8 UART Receive operation in DMA mode 396
22.5 UART REGISTERS 398
22.5.1 UART_RBR (Receive Buffer Register) 398
22.5.2 UART_THR (Transmit Holding Register) 398
22.5.3 UART_DLL (Divisor Latch Low Register) 398
22.5.4 UART_IER (Interrupt Enable Register) 399
22.5.5 UART_DLH (Divisor Latch High Register) 400
22.5.6 UART_IIR (Interrupt Identity Register) 400
22.5.7 UART_FCR (FIFO Control Register) 401
22.5.8 UART_LCR (Line Control Register) 402
22.5.9 UART_MCR (Modem Control Register) 403
22.5.10 UART_LSR (Line Status Register) 404
22.5.11 UART_MSR (Modem Status Register) 405
22.5.12 UART_USR (UART Status Register) 406
22.5.13 UART register map 408

23 SERIAL PERIPHERAL INTERFACE (SPI) 409

23.1 INTRODUCTION 409
23.2 SPI MAIN FEATURES 409
23.3 SPI SIGNAL DESCRIPTIONS 409
23.4 SPI FUNCTIONAL DESCRIPTION 410
23.4.1 Register access 410
23.4.2 Clock Ratios 410
23.4.3 Transmit and Receive FIFO buffers 410
23.4.4 Transfer Modes 411
23.4.5 Master transfer programming in polling mode 411
23.4.6 Master transfer programming in Interrupt mode 413
23.4.7 Master transfer programming in DMA mode 415
23.4.8 Slave transfer programming in polling mode 417
23.4.9 Slave transfer programming in Interrupt mode 420
23.4.10 Slave transfer programming in DMA mode 421
23.5 SPI REGISTERS 424
23.5.1 SPI_CTRLR0 (Control Register 0) 424
23.5.2 SPI_CTRLR1 (Control Register 1) 426
23.5.3 SPI_SSIEN (SPI Enable Register) 426
23.5.4 SPI_MWCR (Microwire Control Register) 427
23.5.5 SPI_SER (Slave Enable Register) 427
23.5.6 SPI_BAUDR (Baud Rate Select) 428
23.5.7 SPI_TXFTLR (Transmit FIFO Threshold Level Register) 428

23.5.8 SPI_RXFTLR (Receive FIFO Threshold Level Register) 429
23.5.9 SPI_TXFLR (Transmit FIFO Level Register) 429
23.5.10 SPI_RXFLR (Receive FIFO Level Register) 430
23.5.11 SPI_SR (Status Register) 430
23.5.12 SPI_IMR (Interrupt Mask Register) 431
23.5.13 SPI_ISR (Interrupt Status Register) 432
23.5.14 SPI_RISR (Raw Interrupt Status Register) 433
23.5.15 SPI_TXO_ICR (Transmit FIFO Overflow Interrupt Clear Register) 434
23.5.16 SPI_RXO_ICR (Receive FIFO Overflow Interrupt Clear Register) 435
23.5.17 SPI_RXU_ICR (Receive FIFO Overflow Interrupt Clear Register) 435
23.5.18 SPI_MST_ICR (Multi-Master Interrupt Clear Register) 435
23.5.19 SPI_ICR (Interrupt Clear Register) 436
23.5.20 SPI_DMA_CR (DMA Control Register) 436
23.5.21 SPI_DMA_TDLR (DMA Transmit Data Level Register) 437
23.5.22 SPI_DMA_RDLR (DMA Receive Data Level Register) 437
23.5.23 SPI_DR (Data Register) 438
23.5.24 SPI register map 439

24 LED DRIVER (LED) 441

24.1 INTRODUCTION 441
24.2 LED DRIVER MAIN FEATURES 442
24.3 LED DRIVER SIGNAL DESCRIPTIONS 442
24.4 LED DRIVER FUNCTIONAL DESCRIPTION 444
24.4.1 Run mode 444
24.4.2 Frame format 444
24.4.3 LED initial and turn on/off 446
24.4.4 Display mode 447
24.4.5 Example for work flow 447
24.5 LED REGISTER 451
24.5.1 LED_CTRL 451
24.5.2 LED_CFG 452
24.5.3 LED_MODE 452
24.5.4 LED_BRG0 453
24.5.5 LED_BRG1 453
24.5.6 LED_BRG2 453
24.5.7 LED_EG 454
24.5.8 LED_STATUS 454
24.5.9 LED_DEBUG 455
24.5.10 LED_BUFFER 455
24.5.11 LED register map 456

25 CAPACITANCE TOUCH SENSOR MODULE (CTSM) 457

25.1 INTRODUCTION 457
25.2 CTSM MAIN FEATURES 458
25.3 CTSM SIGNAL DESCRIPTIONS 459
25.4 CTSM FUNCTIONAL DESCRIPTION 460
25.4.1 Scanning 461
25.4.2 Correction 466
25.4.3 Measurement 466
25.4.4 Tuning sensor offset 467
25.4.5 Auto correction table 468
25.4.6 Interrupts 468
25.4.7 Register setting request 469
25.4.8 Measurement data read request 469
25.4.9 Measurement end 469
25.5 CTSM REGISTER 470
25.5.1 CTSM_INT_CTSM2HOST (interrupt request, used by CTSM system) 470
25.5.2 CTSM_INT_HOST2CTSM (interrupt status, used by CTSM system) 470
25.5.3 CTSM_SYSCFG (used by CTSM system) 471
25.5.4 CTSM_PSC (used by CTSM system) 471
25.5.5 CTSM_DCTRL 472
25.5.6 CTSM_TS 473
25.5.7 CTSM_DIV 474
25.5.8 PS 475
25.5.9 DF (no-used) 476
25.5.10 SR 477
25.5.11 CNT (no-used) 477
25.5.12 FRECFG 478
25.5.13 CHEN 478
25.5.14 CHCTRL 478
25.5.15 ACTRL 479
25.5.16 ACFG (no-used) 480
25.5.17 PE (no-used) 480
25.5.18 AD 481
25.5.19 AF (no-used) 482
25.5.20 AFCCTRL 482
25.5.21 AFCCFG 483
25.5.22 SUCLK0 483
25.5.23 SUCLK1 484
25.5.24 SUCLK2 484
25.5.25 SUCLK3 484
25.5.26 SUCLK4 485
25.5.27 TRIM0 485

25.5.28 TRIM1 485
25.5.29 OPT 486
25.5.30 SCANNUM 487
25.5.31 AJCTRL 487
25.5.32 AJTHR 488
25.5.33 MARSLT (no-used) (INTEN for CTSMtoHost interrupts) 488
25.5.34 CTSM_AJRSLT (no-used) 489
25.5.35 CTSM register map 490

List of Figures

Figure 2 XT32H051 functional block....6
Figure 3 XT32H053 functional block....7
Figure 4 Memory mapping diagram....8
Figure 5 power mode state transition diagram....14
Figure 6 Programming flash....17
Figure 7 Flash sector erase....18
Figure 8 Flash block erase....19
Figure 9 Diagram of internal clock generator....25
Figure 10 Clock tree....26
Figure 11 ICG initialization....28
Figure 12 PMU overview....53
Figure 13 GPIO port control....65
Figure 14 Interrupt process....66
Figure 15 Diagram of the external interrupts line multiplexing to ICTL....75
Figure 16 Reset generation block diagram....76
Figure 17 Handshaking between core and accelerator....80
Figure 18 DMA transfer – 1....145
Figure 19 DMA transfer – 2....146
Figure 20 DMA transfer -3....147
Figure 21 DMA abort....148
Figure 22 IRQ generation....176
Figure 23 FIQ generation....177
Figure 24 Architectural representation of CRC-8....187
Figure 25 CRC calculation....188
Figure 26 MDU block....194
Figure 27 ADC channels and groups mapping....196
Figure 28 ADC routing-1....204
Figure 29 ADC routing-2....205
Figure 30 ADC output code calibration and correction....207
Figure 31 ADC calibration....208
Figure 32 Illustration of PGA input and output....209
Figure 33 Advanced timer block diagram....260
Figure 34 General-purpose timer block diagram....298
Figure 35 Diagram of basic timer....322
Figure 36 Basic timer use flow....324
Figure 37 Independent watchdog timer....331
Figure 38 Windows watchdog timer....335

Figure 39 RTC 340
Figure 40 I2C master transfer data -1 348
Figure 41 I2C master transfer data-2 349
Figure 42 I2C salve transfer data -1 351
Figure 43 I2C slave transfer data -2 352
Figure 44 I2C DMA configuration flow 356
Figure 45 RS232 protocol 383
Figure 46 UART block diagram 384
Figure 47 UART transmit in polling mode 387
Figure 48 UART transmit in Interrupt mode 389
Figure 49 UART transmit in DMA mode 391
Figure 50 UART Receive operation in Polling mode 393
Figure 51 UART Receive operation in Interrupt mode 395
Figure 52 UART Receive operation in DMA mode 397
Figure 53 SPI signals 409
Figure 54 SPI Master transmit & receive in polling mode 413
Figure 55 SPI Master transmit & receive in interrupt mode 415
Figure 56 SPI Master transmit & receive in DMA mode 417
Figure 57 SPI Slave transmit & receive in polling mode 419
Figure 58 SPI Slave transmit & receive in interrupt mode 421
Figure 59 SPI Slave transmit & receive in polling mode 423
Figure 60 LED 8x8 Matrix 441
Figure 61 LED driver diagram 442
Figure 62 Display data flow 448
Figure 63 Display process 449
Figure 64 Example for display data flow 450
Figure 65 Example for display data when DMA error occurs 450
Figure 66 CTSM block diagram 457
Figure 67 CTSM analog and digital interface diagram 461
Figure 68 Scanning status 462
Figure 69 Correct process 466
Figure 70 Measurement 467
Figure 71 Tunning sensor offset 468

List of Tables

Table 1 Device summary 1
Table 2 Common Features 2
Table 3 Specific Features 3
Table 4 XT32H0 memory address 9
Table 5 supported boot modes 12
Table 6 Flash area description 15
Table 7 Operation times unit requirement 19
Table 8 Flash controller register map and reset values 24
Table 9 High speed clock sources 27
Table 10 Peripheral clock setting 29
Table 11 PLL Output Clock Frequency Configuration 37
Table 12 ICG register map and reset values 49
Table 13 AON register map and reset values 63
Table 14 GPIO signal descriptions 65
Table 15 GPIO register map and reset values 72
Table 16 External IRQ assignment 87
Table 17 DMA handshaking interface assignment 1. 89
Table 18 DMA handshaking interface assignment 2. 91
Table 19 GPIO AUX port B assignment 96
Table 20 GPIO AUX port C assignment 97
Table 21 GPIO AUX port D assignment 98
Table 22 SYSCFG register map and reset values 117
Table 23 XT32H05x IO assignment (64 Pins) 123
Table 24 XT32H0xxExxx IO assignment (52 Pins) 126
Table 25 XT32H0xxDxxx IO assignment (48 Pins) 128
Table 26 Configure IO functions 131
Table 27 PADI register map and reset values 138
Table 28 DMA registers and reset values 166
Table 29 Vector table 174
Table 30 ICTL register map and reset values 184
Table 31 CRC register map and reset values 191
Table 32 MDU signal descriptions 192
Table 33 ADC channels mapping. 196
Table 34 ADC channel definition 200
Table 35 Trigger sources 225
Table 36 MDU register map and reset values 254
Table 37 TIMA signal descriptions 261
Table 38 Counting direction 264
Table 39 TIM internal trigger connection 274

Table 40 TIMA register map and reset values 295
Table 41 TIMG signal descriptions 299
Table 42 TIMG register map and reset values 320
Table 43 TIMB signal descriptions 323
Table 44 TIMB register map and reset values 329
Table 45 WDTI register map and reset values 334
Table 46 WDTW register map and reset values 339
Table 47 RTC register map and reset values 344
Table 48 I2C signal descriptions 345
Table 49 The minimum time of I2C 353
Table 50 SDA_RX_HOLD maximum values 355
Table 51 SDA_TX_HOLD minimum values 355
Table 52 I2C register map and reset values 379
Table 53 UART signal descriptions 385
Table 54 UART register map and reset values 408
Table 55 SPI signal descriptions 409
Table 56 SPI register map and reset values 439
Table 57 A suggested driving scheme of 8 8-segment digits 442
Table 58 LED signal descriptions 443
Table 59 Display buffer data format in on-off mode 444
Table 60 Display buffer data format in brightness mode 446
Table 61 LED register map and reset values 456
Table 62 Shared memory mapping 458
Table 63 CTSM signal descriptions 459
Table 64 TSU register map and reset values 490

1 Introduction

1.1 General information

The XT32H05x is a powerful and versatile microcontroller platform based on ARM Cortex M0+ cores and frequency up to 96 MHz, which can be widely used in a variety of applications, including IoT, smart home automation, industrial control, and automotive systems.

The XT32H05x MCU with ARM Cortex-M0+ includes a rich set of peripherals, including LED driver, capacitive touch sensor driver, ADCs, timers, GPIOs, and communication interfaces such as UART, SPI, and I2C. These peripherals provide a wide range of functionality and enable the MCU to interface with a variety of sensors, actuators, motors, and other devices.

1.2 Device summary

Table 1 Device summary

Reference	Part number
XT32H051	XT32H051B, XT32H051C, XT32H051D, XT32H051F
XT32H053	XT32H053B, XT32H053C, XT32H053D, XT32H053F

1.3 Features and Peripherals

The below table shows the peripherals, common features and the specific features of each device of XT32H05x series product.

Table 2 Common Features

Core	ARM Cortex-M0+
Max. Frequency	96 MHz
Voltage range	1.8V - 5.5V
Temperature range	-40℃ - 105℃
Debug interface	SWD
Hardware divider	Yes
Systick timer	1
Deep-sleep wakeup timer	1
Window watchdog(WWDT)	1
Independent watchdog(IWDT)	1
Real-time Counter(RTC)	1
DMA channels	8
Data integrity check(CRC)	CRC8/16/32
Power-on Reset(POR)	Yes
Programmable voltage detector(PVD)	Yes
External Interrupt line(EXTI)	16
Oscillators(OSC)	16MHz internal RC oscillator(HSI) 4-48MHz crystal oscillator(HSE) 32kHz internal RC oscillator(LSI) 32kHz crystal oscillator(LSE) 96MHz fractional digital phased locked loop(PLL)
Basic timer	2
General purpose timer	4
PWM channels	16
complementary PWM channels	6
ADC instances	1
DAC instances	1
Comparators	2
Programmable-gain amplifier(PGA)	2
UART	4
I2C	2
SPI	2

Table 3 Specific Features

OPN	Series	Memory			Package		CTSM				TIMA		DCE	
		Code Flash (KB)	Data Flash (KB)	SRAM (KB)	Pins	GPIOs	self-capacitance channels	mutual-capacitance channels	PGA instances	ADC differential channels	Timer instances	Complementary PWM channels	PFC Accelerator	Math Accelerator
XT32H051B8L5	touch control	128	32	16	32	29	16	8XY - 8Y	4	4	-	3	-	-
XT32H051C8L5	touch control	128	32	16	44	41	16	8XY - 8Y	4	4	-	5	-	-
XT32H051D8L5	touch control	128	32	16	48	45	24	12XY - 10Y	4	4	-	7	-	-
XT32H051F8L5	touch control	128	32	16	64	61	32	16XY - 16Y	4	4	-	12	-	-
XT32H053B8L5	motor control	128	32	16	32	29	-	-	4	4	2	3	Yes	Yes
XT32H053C8L5	motor control	128	32	16	44	41	-	-	4	4	2	5	Yes	Yes
XT32H053D8L5	motor control	128	32	16	48	45	-	-	4	4	2	7	Yes	Yes
XT32H053F8L5	motor control	128	32	16	64	61	-	-	4	4	2	12	Yes	Yes

1.4 Glossary

ADC	Analog to Digital Converter
AHB	Advanced High-performance Bus
AON	Always-ON
APB	Advanced Peripheral Bus
Byte	Data of 8-bit length
COM	Common display terminal
COMP	Comparator
CRC	Cyclic Redundancy Check calculation
CTSM	Capacitance touch sensor module
DAC	Digital to Analog Converter
DMA	Direct Memory Access
FSM	Finite-state machine
GPIO	General Propose IO
half word	16 bits
I2C	Inter-Integrated Circuit
ICG	Internal Clock Generator
ICTL	Interrupt Controller
LED	Light-emitting diodes
LSB	Least Significant Bit
MDU	Motor Driver Unit
MSB	Most Significant Bit
NVIC	Nested Vectored Interrupt Controller
OPAMP	Operational Amplifier
PGA	Programmable Gail Amplifier
PSC	Prescaler
RAM	Random Access Memory
ROM	Read-Only Memory
RTC	Real-Timer Clock
SEG	Segment display terminal
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
SYSCFG	System Configure
TSU	Touch Sensor Unit
UART	Universal Asynchronous Receiver/Transmitter
UID	Unique Identifier (92 bits)
WDT	Watchdog Timer
WDTI	Independent watchdog timer
WDTW	Windows watchdog timer
Word	Data of 32-bit length

2 Memory and bus architecture

2.1 System architecture

The main system architecture is based on 2 master 2 layers AHB bus matrix.

Bus	Modules	Cortex-M0+	DMAC
AHB	SRAM	√	√
	ROM	√	-
	eFlash	√	√
	Clock generator	√	-
	Reset generator	√	-
	Touch sensor	√	√
	Interrupt controller	√	-
	HDIV	√	-
	CRC	√	√
	SIM	√	√
APB1	WDTW	√	√
	WDTI	√	√
	Advanced TIMER	√	√
	General Timer	√	√
	Basic Timer	√	√
	UART	√	√
	I2C	√	√
	SPI	√	√
	MDU	√	√
	LED driver	√	√
	RTC	√	√
	PMU	√	√
APB2	GPIO	√	√

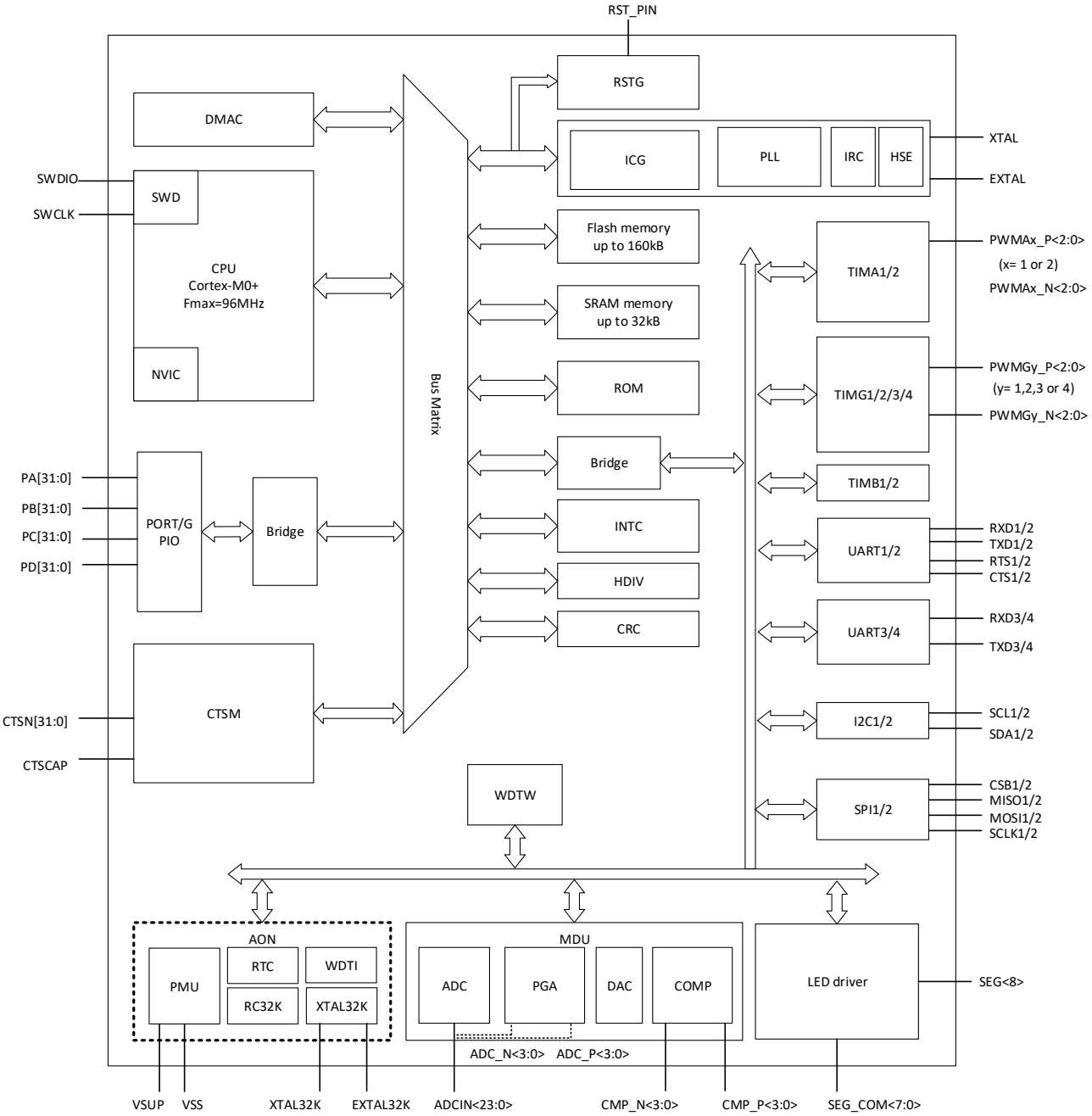


Figure 1 XT32H051 functional block

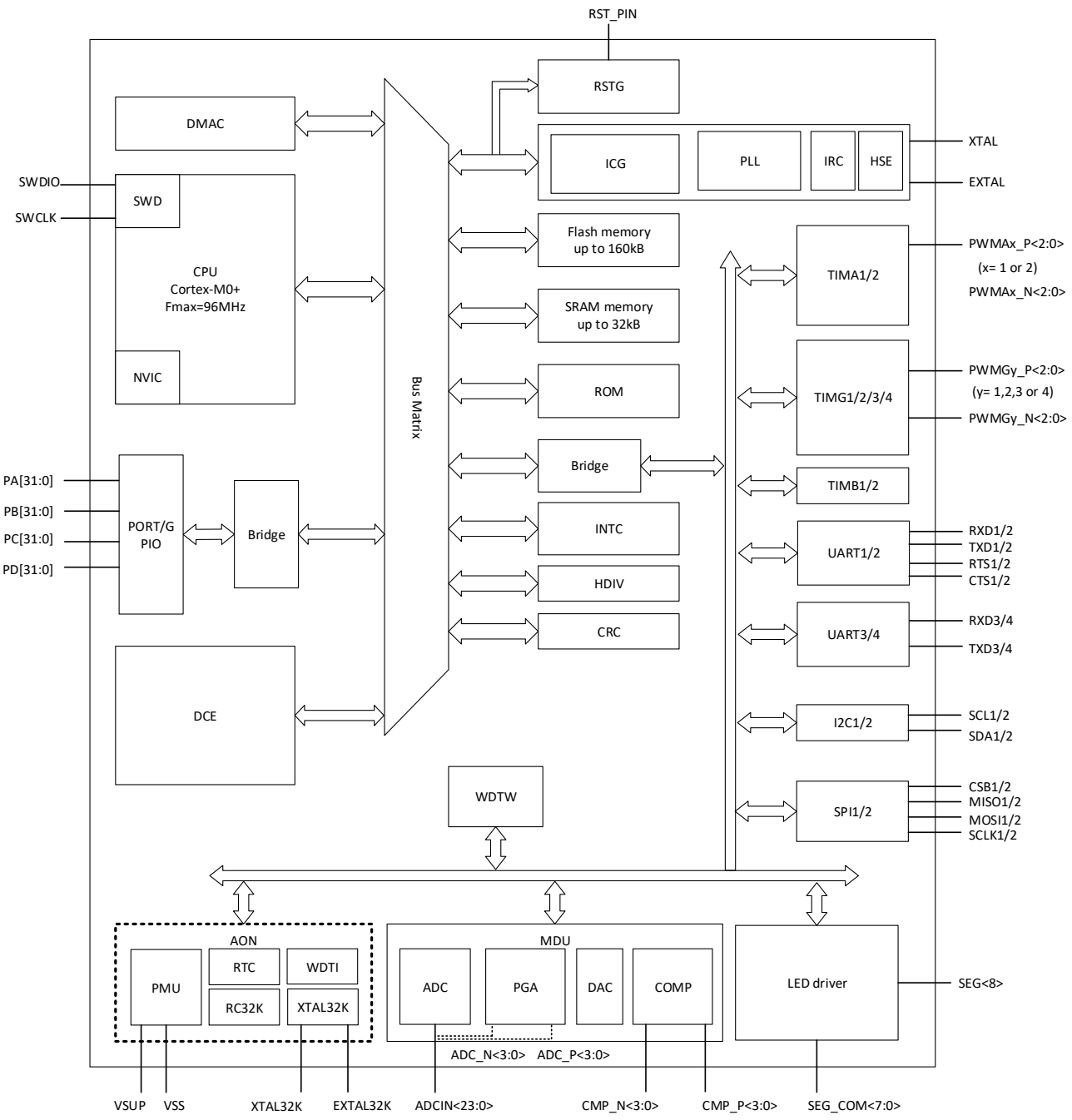


Figure 2 XT32H053 functional block

2.2 Memory

2.2.1 Introduction

The bytes are codes in little endian. The lowest numbered byte in a word is considered the least significant byte of the word.

2.2.2 Memory mapping

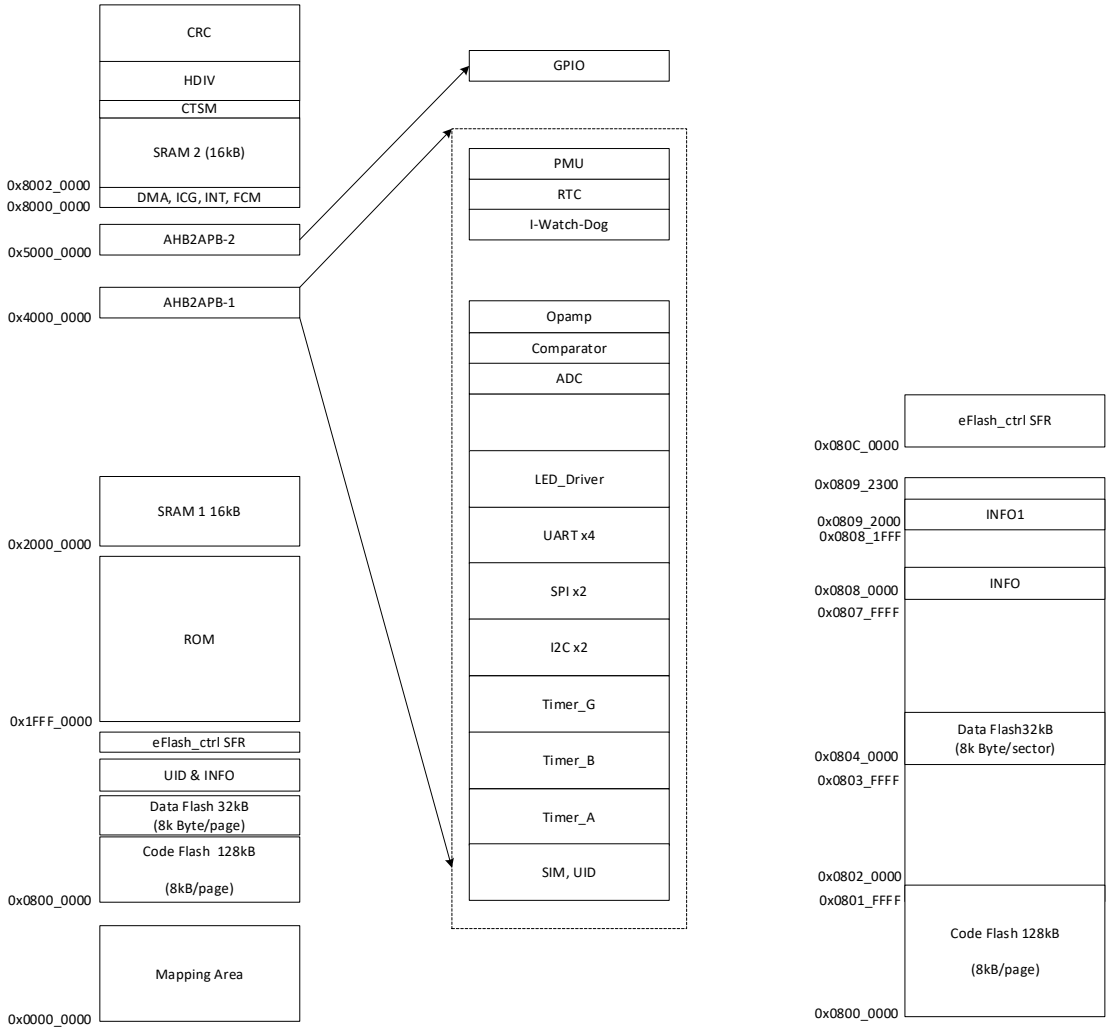


Figure 3 Memory mapping diagram

Table 4 XT32H0 memory address

Type	Start Address	Stop Address	Size	Memory Area Description
AHB	0x8005_0000	0x8005_FFFF	64KB	CRC
	0x8003_0000	0x8003_FFFF	64KB	External interrupt controller
	0x8002_0000	0x8002_FFFF	64KB	Reset generator
	0x8001_0000	0x8001_FFFF	64KB	Clock generator
	0x8000_0000	0x8000_FFFF	64KB	DMAC
APB	0x5000_0C00	0x5000_0FFF	1KB	GPIO
	0x5000_0800	0x5000_0BFF	1KB	PADI
	0x4000_3600	0x4000_36FF	256B	AON
	0x4000_3500	0x4000_35FF	256B	RTC
	0x4000_3400	0x4000_34FF	256B	Independent WDT
	0x4000_3000	0x4000_30FF	256B	LED
	0x4000_2800	0x4000_2BFF	1KB	MDU
	0x4000_2700	0x4000_27FF	256B	SPI2 slave
	0x4000_2600	0x4000_26FF	256B	SPI2 master
	0x4000_2500	0x4000_25FF	256B	SPI1 slave
	0x4000_2400	0x4000_24FF	256B	SPI1 master
	0x4000_2100	0x4000_21FF	256B	I2C2
	0x4000_2000	0x4000_20FF	256B	I2C1
	0x4000_1F00	0x4000_1FFF	256B	UART4
	0x4000_1E00	0x4000_1EFF	256B	UART3
	0x4000_1D00	0x4000_1DFF	256B	UART2
	0x4000_1C00	0x4000_1CFF	256B	UART1
	0x4000_1800	0x4000_1BFF	1KB	General Timer
	0x4000_1500	0x4000_15FF	256B	Basic Timer group 2
	0x4000_1400	0x4000_14FF	256B	Basic Timer group 1
	0x4000_1000	0x4000_13FF	1KB	Advanced Timer
	0x4000_0C00	0x4000_0FFF	1KB	Windows WDT
	0x4000_0800	0x4000_0BFF	1KB	SYSCFG
AHB	0x2000_0000	0x2000_7FFF	32KB	SRAM

Type	Start Address	Stop Address	Size	Memory Area Description
AHB	0x1FFF_0000	0x1FFF_FFFF	64KB	ROM
	0x0808_0000	0x0809_FFFF	128KB	Flash info & info1 area
	0x0804_0000	0x0807_FFFF	256KB	Flash data memory
	0x0800_0000	0x0803_FFFF	256KB	Flash code memory
	0x0000_0000	0x0001_FFFF	128KB	Mapping memory on BOOT configuration

3 Reset and Operation Mode

3.1 Reset

This section discusses basic reset mechanisms and reset sources.

3.1.1 System resets and sources

Resetting the MCU provides a way to start processing from a known set of initial conditions. System reset begins with the on-chip regulator in full regulation and clocking generation from an internal reference clock. After reset, most I/O pins are initially configured as disabled except SWD and boot related pins.

The various system reset sources are discussed in the following sub-sections.

3.1.1.1 Power-on reset (POR)

When power is initially applied to the MCU or when the supply voltage drops below the power-on reset re-arm voltage level, the POR circuit causes a POR reset condition.

3.1.1.2 External pin reset (RSTb)

On this device, RSTb pin has an internal pull-up resistor. Asserting RSTb will reset whole system in normal operation mode. It cannot reset or wake-up chip from external when chip is in deep-sleep mode.

3.1.1.3 Watchdog reset

The watchdog timer monitors the operation of the system by expecting periodic communication from the software, generally known as servicing (or refreshing) the watchdog. If this periodic refreshing does not occur, the watchdog issues a system reset.

3.1.1.4 Software reset

The SYSRESETREQ bit in the NVIC application interrupt and reset control register can be set to force a software reset on the device. Setting SYSRESETREQ generates a software reset request. This reset forces a system reset of all major components except for the debug module.

3.1.1.5 Debug reset (DAP reset)

The chip can be reset while in debug mode by generating a DAP reset. The reset command is sent to

the DAP (Debug Access Port) which when decoded generates a system reset and the chip executes the reset sequence.

3.2 Boot

This section describes the boot modes.

3.2.1 Boot modes

The device supports three different modes, which can be configured by setting BOOT0 and BOOT1 pins to a dedicated status during RSTb pin is asserted, as shown in the following table. This device boots up in functional mode by default using the internal flash as the boot source after any reset or power up.

Table 5 supported boot modes

BOOT1	BOOT0	Boot Mode
0	0	Boot from eFlash (0x0800_0000)
0	1	Boot from ROM (0x1FFF_0000)
1	0	Boot from SRAM (0x2000_0000)
1	1	Reserved, default Boot from eFlash

3.2.2 Boot sequence

3.3 Operation mode

The device support multiple power options to allow the user to optimize power consumption for the level of functionality needed.

The three primary modes of operation are run, wait and stop. The WFI instruction invokes both wait and stop modes for the chip. The primary modes are augmented in a number of ways to provide lower power based on application needs.

For each run mode (Normal Run and Low-Speed Run) there is a corresponding wait and stop mode. Wait modes are similar to ARM sleep modes. Stop modes are similar to ARM sleep deep mode. The low-speed run operating mode can drastically reduce runtime power when the maximum bus frequency is not required to handle the application needs.

Depending on the stop requirements of the user application, a variety of stop modes are available that provide state freeze (by clock gating), partial power down or full power down of certain logic and/or memory.

Chip mode	Description	Core mode	Normal recovery method
Normal Run (RUN)	Allows maximum performance of chip. Default mode out of reset; on-chip voltage regulator is on.	Run	-
Normal Wait (WAIT)	Allows peripherals to function while the core is in sleep mode, thus reducing power. NVIC remains sensitive to interrupts; peripherals continue to be clocked.	Stop	Interrupt
Normal Stop (STOP)	Core and System clocks are gated while Bus clock is active. Allows peripherals to function. Clock generators and regulators are fully functional.	Stop	Interrupt
Low-Speed Run (LSRUN)	Power down PLL and unused HS Clock, use very low speed clock as system clock. All system works same as normal run.	Run	-
Low-Speed Wait (LSWAIT)	Allows peripherals to function while the core is in sleep mode, thus reducing power. NVIC remains sensitive to interrupts; peripherals continue to be clocked.	Stop	Interrupt
Low-Speed Stop (LSSTOP)	Core and System clocks are gated while Bus clock is active. Allows peripherals to function. Clock generators and regulators are fully functional.	Stop	Interrupt
Deep-Sleep	The chip is powered down except for the VBAT supply and AON block.	Off	Dedicated wakeup timer/external

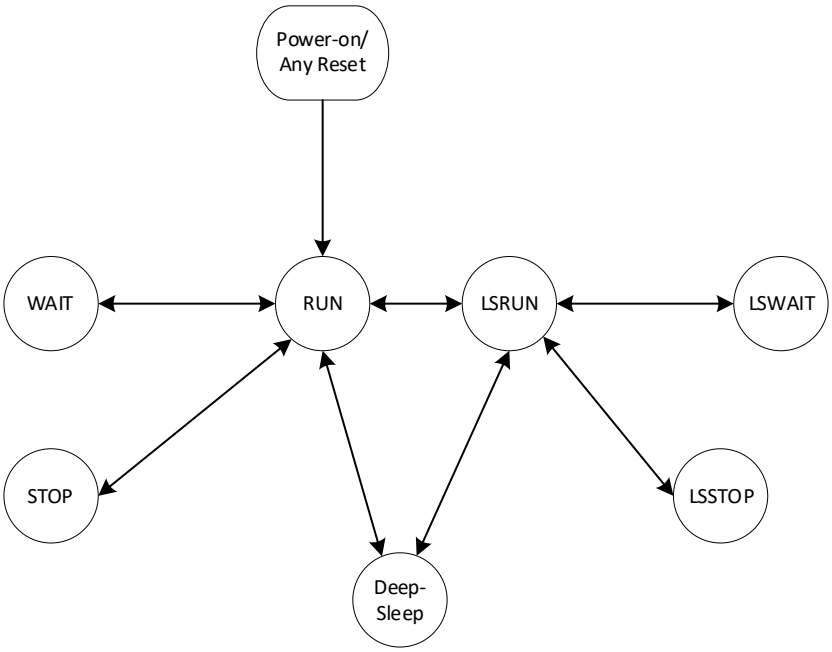


Figure 4 power mode state transition diagram

4 Flash controller (FLASH)

4.1 Introduction

Flash memory includes four areas which are code, data, info and info1 configuration. The follow table shows the address and size of four areas.

Table 6 Flash area description

Area Name	Start Address	Stop Address	Memory Area Description
flash_code	0x0800_0000	0x0801_FFFF	Code Area, 128k Bytes 16 pages
flash_data	0x0804_0000	0x0804_7FFF	Data Area, 32k Bytes 4 pages
flash_info	0x0808_0000	0x0808_1FFF	INFO Area, 8k Bytes 1 pages
flash_info1	0x0809_2000	0x0809_3FFF	INFO1 Area, 8k Bytes 1 page

4.2 Flash main features

- Fast read, adaptive to bus read frequency up to 96MHz
- Two erase mode: page erase and mass erase
- 32bit word programming
- Configure operating time

4.3 Flash Operation description

4.3.1 Read Operation

Flash read is similar to normal memory.

1. Set access mode (FLASH_CTRL.MODE) to read mode.
2. Read from address. For example,

```
unsigned int a = *(unsigned int *)(0x08000004);
```

4.3.2 Programming Operation

Flash programming performs the following steps:

Configure operation time, details in section 3.3.5;
set PROGRAMING mode in FLASH.FLASH_CTRL.MODE;
Write data to target address, for example
(unsigned int)(0x80000040) = 0xf0405067;
Set READ mode in FLASH.FLASH_CTRL.MODE;
Repeat step 2 to step 4 to write multiple data.

Note：Once the flash macro is working in programming mode, it is totally controlled by flash controller, it cannot be accessed by CPU through system bus. So this kinds of Flash programming control codes cannot be fetched from Flash. Therefore, these codes should be located in SRAM/ROM, and it is necessary to disable all interrupts during flash programming modes. Once the flash programming operation is completed and flash operation mode is set back to read mode, the interrupts can be enabled again.

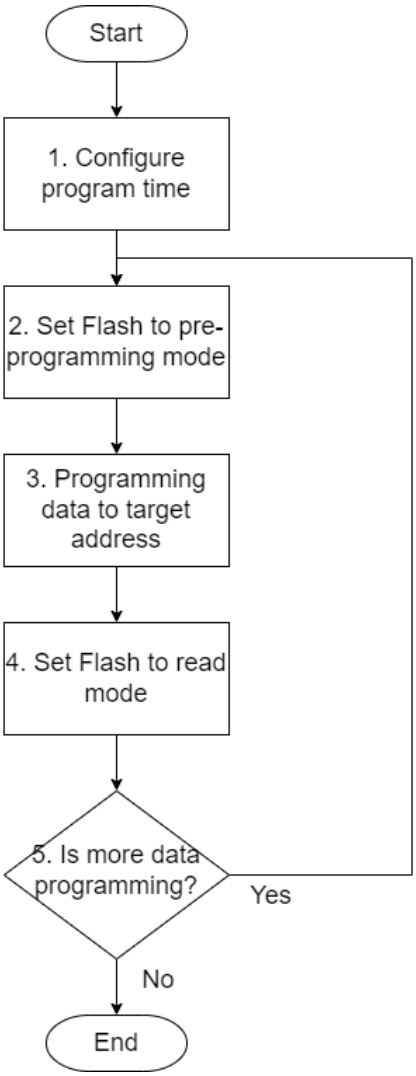


Figure 5 Programming flash

4.3.3 Page Erase Operation

Page erase performs the following steps:

1. Set PAGE ERASE mode in FLASH.FLASH_CTRL.MODE;
2. Write any data to any address in the erasing page;
3. Set READ mode in FLASH.FLASH_CTRL.MODE;
4. Repeat step 2 and step 3 to erase multiple pages.

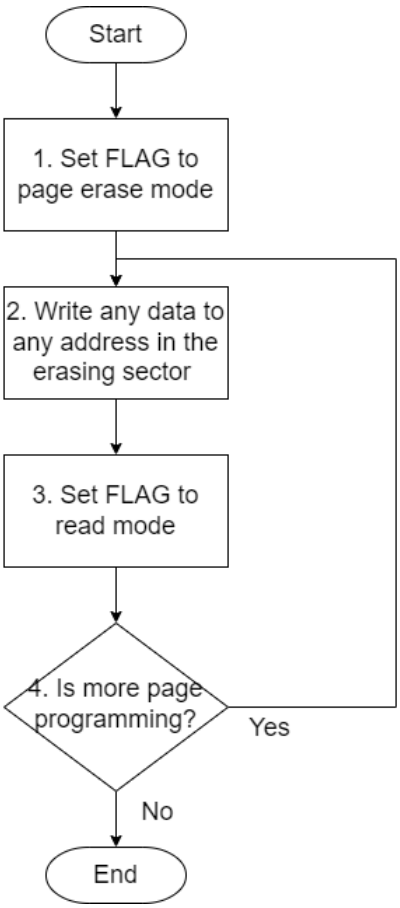


Figure 6 Flash sector erase

4.3.4 Mass/Chip Erase Operation

Chip erase performs the following steps:

1. Set CHIP ERASE mode in FLASH.FLASH_CTRL.MODE;
2. Write any data to any address, if the address is not in info1 area, the area excluding info1 will be erased, otherwise the erasing area would include info1 area;
3. Set READ mode in FLASH.FLASH_CTRL.MODE

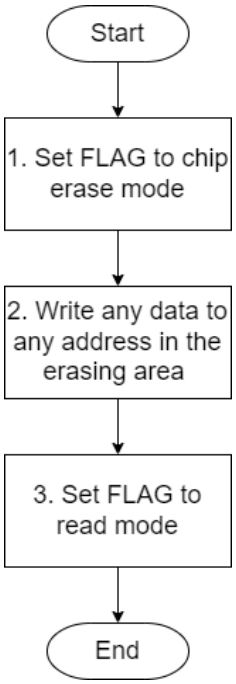


Figure 7 Flash block erase

Note：Once the flash macro is working in erase mode, it is totally controlled by flash controller, it cannot be accessed by CPU through system bus. So this kinds of Flash erase control codes cannot be fetched from Flash. Therefore, these codes should be located in SRAM/ROM, and it is necessary to disable all interrupts during flash erase modes. Once the flash erase operation is completed and flash operation mode is set back to read mode, the interrupts can be enabled again.

4.3.5 Configure operation time

Flash programming and erase is timing critical operation, users need to set correct value in register FLASH_SMW_SET_TIMER to generate a 1us time unit according to the running bus clock frequency.

Table 7 Operation times unit requirement

Time	Time minimus	Value @ 96MHz	Value @ 48MHz
FLASH_SMW_SET_TIMER.TMR_GEN[7:0]	>=1us	96	48

4.4 Flash registers

4.4.1 FLASH_CTRL(Flash control register)

Base address: 0x080C_0000
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0000

<table>
<tr><th>Bit</th><th>Description</th><th>Type</th></tr>
<tr><td><31:30></td><td>CTRL_ACCWAIT[1:0]: eFlash read access waiting cycle number control
2’b00: no waiting
2’b01: wait 1 cycle
2’b1x: waiting cycle is auto controlled by ICG_CLK_CFG<1:0> and ICG_PLL_DIV<3:0>

<table>
<tr><th>ICG_CLK_CFG<1:0></th><th>Clock Source</th><th>HCLK_DIV<3:0></th><th>Waiting cycle</th></tr>
<tr><td>2’b00</td><td>16MHz RC</td><td>-</td><td>0</td></tr>
<tr><td rowspan="2">2’b01</td><td rowspan="2">48MHz XTAL</td><td>0, 1</td><td>1</td></tr>
<tr><td>>1</td><td>0</td></tr>
<tr><td rowspan="3">2’b10</td><td>96~66Mhz PLL CLK</td><td></td><td>2</td></tr>
<tr><td>64~50MHz PLL CLK</td><td></td><td>1</td></tr>
<tr><td>49~33MHz PLL CLK</td><td></td><td>0</td></tr>
</table>
</td><td>R/W</td></tr>
<tr><td><29:28></td><td>Reserved</td><td>R/W</td></tr>
<tr><td><27></td><td>CTRL_LVE: Low voltage enable (to LVE pin of flash macro),
1’b0: VDD_CORE = 1.1V
1’b1: VDD_CORE = 0.9V</td><td>R/W</td></tr>
<tr><td><26:25></td><td>Reserved</td><td>R/W</td></tr>
<tr><td><24></td><td>CTRL_TMR: eFlash TMEN control signal
1’b0: reset the test register in flash macro
1’b1: normal operation</td><td>R/W</td></tr>
<tr><td><23:3></td><td>Reserved</td><td>R/W</td></tr>
<tr><td><2:0></td><td>CTRL_MODE[2:0]: Flash access mode
3’b000: Read, waiting cycles (1,2,3 cycles)
3’b001: Reserved
3’b010: Chip/Mass erase
3’b011: Smart page erase
3’b100: Smart program mode 1
3’b101: Reserved
3’b110: Smart program mode 2
3’b111: Reserved</td><td>R/W</td></tr>
</table>

4.4.2 FLASH_SMW_SET_OPTION

Base address: 0x080C_0000
Address offset: 0x04
Type: Read/Write
Reset value: 0x000F_1680

Bit	Description	Type
<31>	**OPT_PPVRD**: Pre-PV read disable control 1'b0: enable the pre-PV-read before program cycle when run SMP1 and SMP2 command 1'b1: disable the pre-PV-read befroe program cycle when run SMP1 and SMP2 command	R/W
<30>	**OPT_MASK**: mask option 1'b0: successfully programmed bits will not be read and programmed again till extra shot 1'b1: successfully programmed bits will be read and programmed again	R/W
<29>	Reserved.	R/W
<28:24>	**OPT_MAXPRNUM[4:0]**:Max Prog number, default 5'd5	R/W
<23:20>	**OPT_EXPULSE[3:0]**: extra_pulse, extra post shots for Program and Erase default: 4'b0101 <23:22> for Program <21:20> for Erase	R/W
<19:17>	Reserved.	R/W
<16:8>	**OPT_MAXERNUM[8:0]**: Max Erase number, default 9'b0 0101 0110	R/W
<7:6>	Reserved.	R/W
<5:4>	**OPT_TPGM[1:0]**:T_{pgm} control, default 2'b01 2'b00: 1us, 2'b01: 2us, 2'b10 : 4us, 2'b11: 8us	R/W
<3>	Reserved.	R/W
<2:0>	**OPT_TERS[2:0]**:T_{ers} control, default 3'b010 3'b000: 50us, 3'b001: 100us, 3'b010: 200us, 3'b011: 300us, 3'b100: 500us, 3'b101: 1ms, 3'b100: 1.5ms, 3'b111: 2ms	R/W

4.4.3 FLASH_SMW_SET_HEMWHV

Base address: 0x080C_0000
Address offset: 0x08
Type: Read/Write
Reset value: 0x000E_A600

Bit	Description	Type
<31:22>	Reserved.	R/W
<21>	**HEM_CTRL**: smw hem control for HEM-Erase or Non-HEM-Erase in SME command.	R/W
<20>	**HEM_WAIT**: smw wait control provide wait cycle, 1'b0: no wait, 1'b1: insert wait cycle	R/W
<19:17>	Reserved.	R/W
<16:8>	**HEM_MAXERHEM[8:0]**:Max-Erase for HEM-Erase cycle. Default value: 9'b1 1011 1101 For 100k endurance area, max: 445+1+1	R/W
<7:0>	**HEM_HEMWHV[7:0]**: WHV Counter for HEM-Erase cycle, add the extra shots at the 15th shot, use the same WHV as the 15th shot. Default value 8'b0000 0000	R/W

4.4.4 FLASH_SMW_SET_TIMER

Base address: 0x080C_0000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_01E0

Bit	Description	Type
<31:13>	Reserved	R/W
<12:8>	**TMR_RD[4:0]**: Define # of state required for PV/EV read cycle. The total clocks for a PV/EV read cycle will be: clock_cycle * setting > Tmacc and clock_cycle * (setting + 1) > Tmpws clock_cycle * (setting + 1) > Tmnws clock_cycle * 2 * (setting + 1) > Tmcyc Spec: min 100ns for Tmpws, Tmnws, Tmacc Default: 5'b0 1100	R/W
<7:0>	**TMR_GEN[7:0]**: Define # of clock cycles to generate a >= 1us pulse e.g. minimum 10 when clock runs at 100ns (10MHz) Default: 8'b0110 0100 for 100MHz system clock.	R/W

4.4.5 FLASH_NSMW_SET_WHV

Base address: 0x080C_0000
Address offset: 0x10
Type: Read/Write
Reset value: 0x0000_01E0

Bit	Description	Type
<31:12>	Reserved.	R/W
<11:8>	**WHV_WHV[3:0]**: WHV for non-smart write Default: 4’b1100	R/W
<7>	Reserved.	R/W
<6:4>	**WHV_WMV[2:0]**: WMV for non-smart write Default: 3’b101	R/W
<3:2>	Reserved.	R/W
<1:0>	**WHV_WIPGM[1:0]**: WIPGM for non-smart write Default: 2’b00	R/W

4.4.6 Flash register map

The base address of the Flash controller registers is 0x080C_0000.

The following table gives the Flash controller register map and reset values.

Table 8 Flash controller register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FLASH_CTRL		CTRL_ACCWAIT[1:0]	Res	Res	CTRL_LVE	Res	Res	CTRL_TMR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CTRL_MODE[2:0]		
	Reset value	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
0x04	FLASH_SMW_SET_OPTION	OPT_PPVRD	OPT_MASK	Res	OPT_MAXPRNUM[4:0]								OPT_WXPULSE[3:0]				OPT_MAXERNUM[8:0]												OPT_TPGM[1:0]				OPT_TERS[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0
0x08	FLASH_SMW_SET_HEMWHV	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	HEM_CTRL	HEM_WAIT	Res	Res	Res	HEM_MAXERHEM[8:0]									HEM_HEMWHV[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
0x0C	FLASH_SMW_SET_TIMER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TMR_RD[4:0]					TMR_GEN[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
0x10	FLASH_NSMW_SET_WHV	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WHV_WHV[3:0]				Res			WHV_WMV[2:0]		Res	Res	WHV_WIPEN[1:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0

5 Internal clock generator (ICG)

5.1 Introduction

XT32H05x series provides flexible configuration of high-speed clocks and low speed clocks. ICG manages high speed clocks and AON manages low speed clocks (refer to session 6).

The ICG manages 3 clock sources: 16MHz RC oscillator, 4MHz to 48MHz crystal oscillator, and MCU PLL. Secondary clocks are generated by dividing these 3 clock sources.

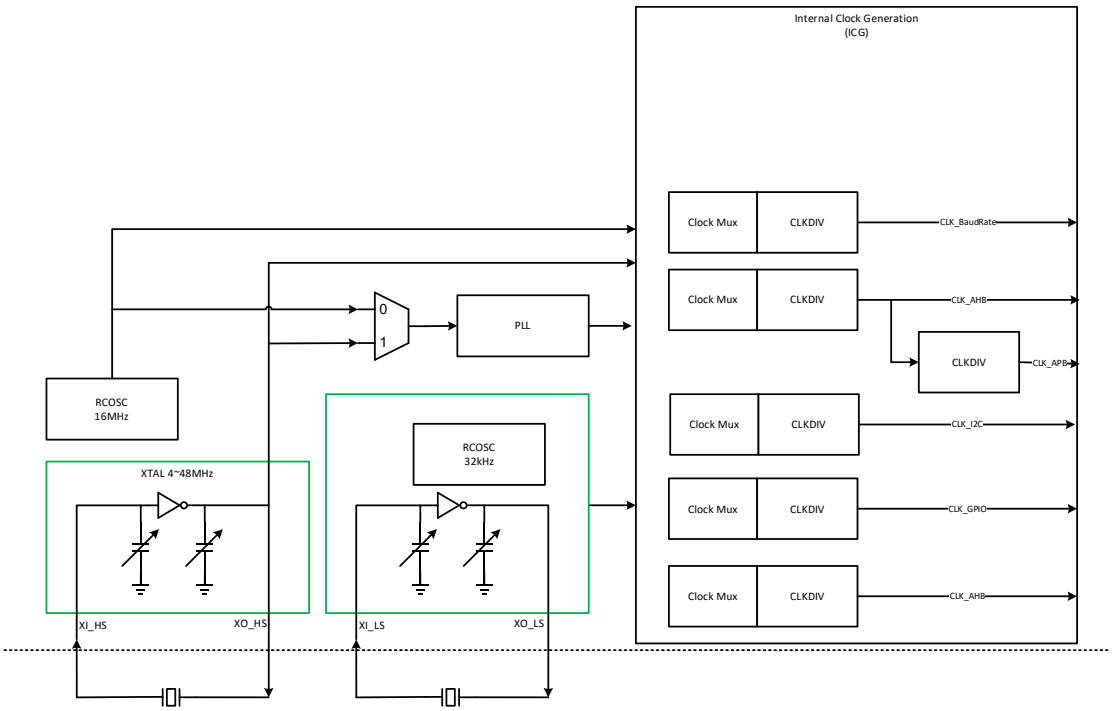


Figure 8 Diagram of internal clock generator

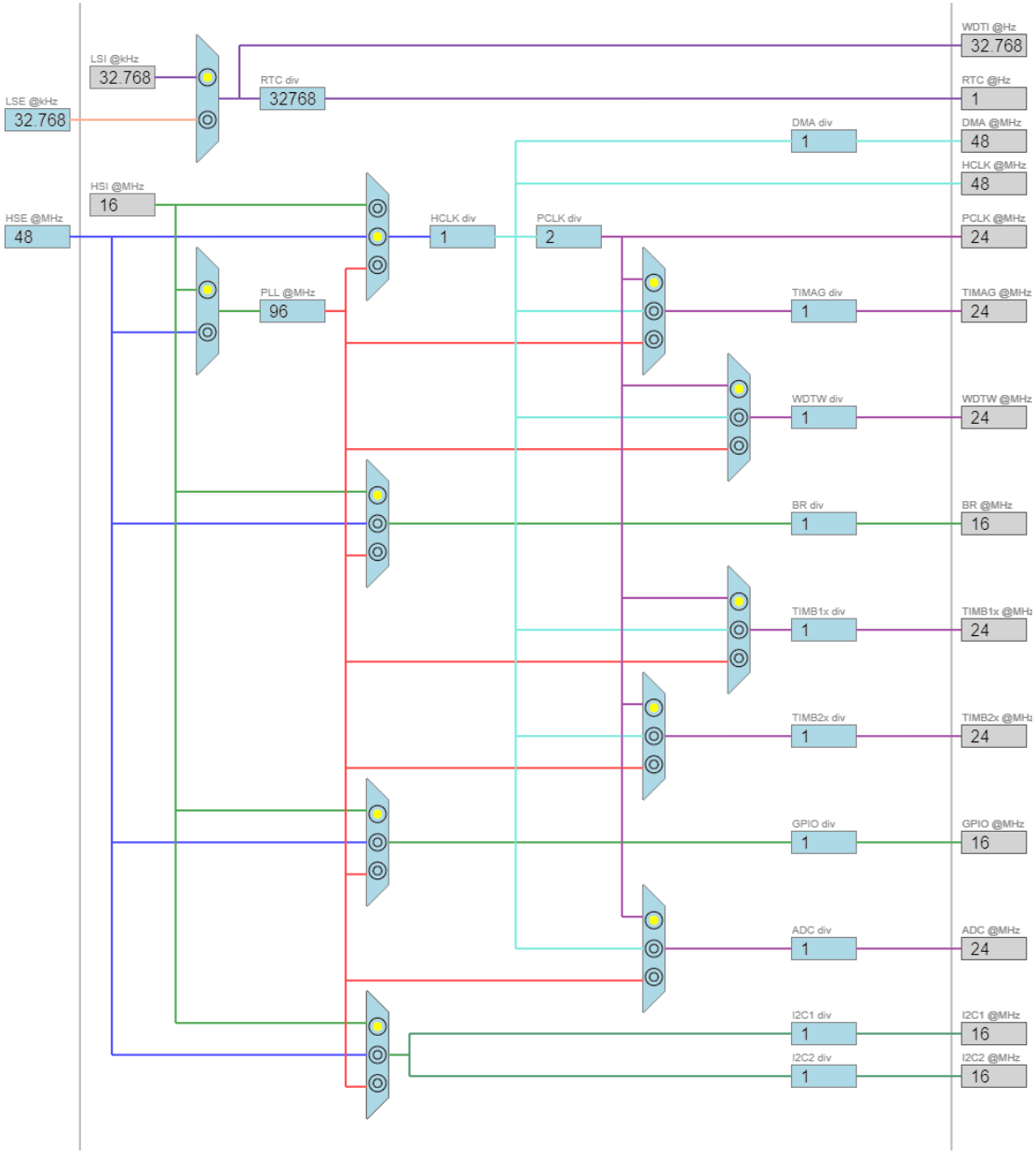


Figure 9 Clock tree

5.2 ICG main features

-Power switches clock sources

-Generate PLL clock

-Generate HCLK and PCLK

-Provide clock for peripherals: ADC, general timer, advanced timer, basic timers, GPIO debounce clock, I2C clock and baud rate clock.

5.3 ICG functional description

5.3.1 Clock Sources

The device provides the following clock sources to produce the primary clocks:

- 16MHz RC - a high-speed fully integrated RC oscillator producing 16MHz clock
- 4 to 48 MHz OSC – a high-speed oscillator with external crystal/ceramic resonator or external clock source
- Up to 96MHz PLL – a high-speed fully-integrated PLL producing up to 96MHz clock
- 32kHz RC – a low-speed fully-integrated RC oscillator producing 32kHz clock

Each oscillator can be switched on or off independently when it is not used for optimal power performance. The sub-sections of this chapter will provide further details of various clock sources. These clocks are divided down through configurable multi-stage clock divider to generate the secondary clocks for the target application. The 32kHz low-speed clock are used in the RTC and watchdog module for system monitoring purpose. It is also utilized during low-power mode to keep the system active.

Figure 9 shows the clock tree diagram in XT32H05x.

Table 9 High speed clock sources

Clock source	Description
High speed RC oscillator	16MHz, may be calibrated
High speed external crystal	Supported from 4 to 48 MHz
PLL using 16M RC oscillator as source	Setting PLL pre-divider to configure reference clock to 4MHz, then setting feedback divider and post divider to configure PLL clock output, refer to Table xx
PLL using external crystal as source	Same as above cell

5.3.2 High speed clock setting

1. Power on clock source,

The XT32H05x series supports external crystal from 4MHz to 48MHz. To use the external crystal, set both ICG_CLK_CFG.CTL_HSXTAL[17] and ICG_CLK_CFG.CTL_HSXTALOUT[18] to 0.

To use the 16MHz RC oscillator as clock source, set ICG_CLK_CFG.CTL_HSR[16] to 0.

There are two clock sources available to the PLL: 16MHz RC oscillator or external crystal. To select 16MHz RC oscillator as the PLL clock source, set PLL_LFCFG_CTRL.PLL_CLKSRC[24] to 0. The external crystal would be used as PLL clock source, if PLL_LFCFG_CTRL.PLL_CLKSRC[24] is set to 1.

The PLL clock is further divided down through the PLL divider.

To configure the PLL divider. Set ICG_PLL_DIV.PLL_PREDIV[19:16] to configure reference clock to 4MHz, pre-divider value is shown in session 5.4.10. Then set ICG_PLL_DIV.PLL_FBDIV[13:8] and ICG_PLL_DIV.PLL_POSTDIV[3:0] to configure PLL frequencies according to the application requirement. The setting details can be found in Table xx.

PLL can be power on by writing 0 to ICG_CLK_CFG.CTL_PLL[19].

XT32H05x can be configured to select different bus clock sources. If external crystal oscillator is used as system clock source, write 1 to ICG_CLK_CFG.CTL_HCLKSRC[0:1]. If internal RC oscillator is used as system clock source, write 0 to ICG_CLK_CFG.CTL_HCLKSRC[0:1]. If PLL clock is used as system clock source, write 2 to ICG_CLK_CFG.CTL_HCLKSRC[0:1].

HCLK clock divider can be set by writing the divider values to ICG_HCLK_DIV.

PCLK clock divider can be set by writing the divider values to ICG_PCLK_DIV.

For the selected peripheral, the users should select the appropriate peripheral clock source and configure the clock source accordingly. The details were shown in Figure 11.

To set the peripheral clock divider, please refer to Table. 10 for the register settings.

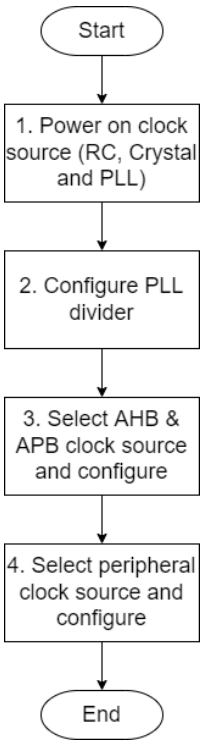


Figure 10 ICG initialization

Table 10 Peripheral clock setting

Peripherals	Clock source	Register	Clock divider	Register
Window Watch Dog Timer	PCLK	ICG_CLK_CFG[3:2] = 0	divider	ICG_WDT_W_CLK_DIV
Window Watch Dog Timer	HCLK	ICG_CLK_CFG[3:2] = 1	divider	ICG_WDT_W_CLK_DIV
Window Watch Dog Timer	PLL	ICG_CLK_CFG[3:2] = 2	divider	ICG_WDT_W_CLK_DIV
Window Advanced Timer	PCLK	ICG_CLK_CFG[15:14] = 0	divider	ICG_TIMER_ADV_CLK_DIV
Window Advanced Timer	HCLK	ICG_CLK_CFG[15:14] = 1	divider	ICG_TIMER_ADV_CLK_DIV
Window Advanced Timer	PLL	ICG_CLK_CFG[15:14] = 2	divider	ICG_TIMER_ADV_CLK_DIV
Window Basic Timer1	PCLK	ICG_CLK_CFG[7:6] = 0	divider	ICG_TIMER_B1_CLK_DIV
Window Basic Timer1	HCLK	ICG_CLK_CFG[7:6] = 1	divider	ICG_TIMER_B1_CLK_DIV
Window Basic Timer1	PLL	ICG_CLK_CFG[7:6] = 2	divider	ICG_TIMER_B1_CLK_DIV
Window Basic Timer2	PCLK	ICG_CLK_CFG[11:10] = 0	divider	ICG_TIMER_B2_CLK_DIV
Window Basic Timer2	HCLK	ICG_CLK_CFG[11:10] = 1	divider	ICG_TIMER_B2_CLK_DIV
Window Basic Timer2	PLL	ICG_CLK_CFG[11:10] = 2	divider	ICG_TIMER_B2_CLK_DIV
Window ADC	PCLK	ICG_CLK_CFG[31:30] = 0	divider	ICG_ADC_CLK_DIV
Window ADC	HCLK	ICG_CLK_CFG[31:30] = 1	divider	ICG_ADC_CLK_DIV
Window ADC	PLL	ICG_CLK_CFG[31:30] = 2	divider	ICG_ADC_CLK_DIV
Window Baud Rate	PCLK	ICG_CLK_CFG[5:4] = 0	divider	ICG_BR_CLK_DIV
Window Baud Rate	HCLK	ICG_CLK_CFG[5:4] = 1	divider	ICG_BR_CLK_DIV
Window Baud Rate	PLL	ICG_CLK_CFG[5:4] = 2	divider	ICG_BR_CLK_DIV
Window I2C1	PCLK	ICG_CLK_CFG[9:8] = 0	divider	ICG_I2C1_CLK_DIV
Window I2C1	HCLK	ICG_CLK_CFG[9:8] = 1	divider	ICG_I2C1_CLK_DIV
Window I2C1	PLL	ICG_CLK_CFG[9:8] = 2	divider	ICG_I2C1_CLK_DIV
Window I2C2	PCLK	ICG_CLK_CFG[9:8] = 0	divider	ICG_I2C2_CLK_DIV
Window I2C2	HCLK	ICG_CLK_CFG[9:8] = 1	divider	ICG_I2C2_CLK_DIV
Window I2C2	PLL	ICG_CLK_CFG[9:8] = 2	divider	ICG_I2C2_CLK_DIV
Window GPIO	RCOSC	ICG_CLK_CFG[13:12] = 0	divider	ICG_GPIO_CLK_DIV
Window GPIO	XTAL	ICG_CLK_CFG[13:12] = 1	divider	ICG_GPIO_CLK_DIV
Window GPIO	PLL	ICG_CLK_CFG[13:12] = 2	divider	ICG_GPIO_CLK_DIV

5.3.3 High speed RC oscillator clock calibration

Refer to session 8.3.7.

5.4 ICG Registers

5.4.1 ICG_CLK_CFG

Base address: 0x8001_0000
Address offset: 0x00
Type: Read/Write
Reset value: 0x04AA_0000

Bit	Description	Type
<31:30>	**CLK_CTL_ADCSRC[1:0]**: ADC clock source selection control 2'b00: PCLK 2'b01: HCLK 2'b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2'b11: Reserved	R/W
<29>	**CLK_CTL_TEST_OUT**: XTAL output clock source selection 1'b0: internal xtal's output 1'b1: external clock from XTAL_O pin	R/W
<28>	**CLK_CTL_HSXTAL_CRES**: High speed crystal cres control default 0, feedback resistor div2 enable crystal shunt resistance control, cres=1, res=36k ohm; cres=0 ,res=72k ohm	R/W
<27:24>	**CLK_CTL_HSXTAL_CTUNE[3:0]**:High speed crystal cap_tune<3:0> default 0100 4 bits crystal capacity loading setting: 12p(fixed)+8p+4p+2p+1p	R/W
<23:20>	**CLK_CTL_HSXTAL_ITUNE[3:0]**:High speed crystal i_tune<3:0> default 1010 4 bits current bias control for osc core: 175uA(fixed)+50uA+100uA+200uA+400uA, default 0100 225uA	R/W
<19>	**CLK_CTL_PLL**: MCU_PLL power down control 1'b0: PLL power on 1'b1: PLL power down	R/W
<18>	**CLK_CTL_HSXTALOUT**: High speed crystal output clock power down control 1'b0: enable 1'b1: disable	R/W
<17>	**CLK_CTL_HSXTAL**: High speed crystal oscillator power down control 1'b0: High speed crystal power on 1'b1: High speed crystal power down	R/W
<16>	**CLK_CTL_HSRC**: High speed RC oscillator power down control 1'b0: High speed RC oscillator power on 1'b1: High speed RC oscillator power down	R/W
<15:14>	**CLK_CTL_TIMAGSRC[1:0]**: GP Timer and Advanced Timer clock source selection 2'b00: PCLK	R/W

Bit	Description	Type
	2’b01: HCLK 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	
<13:12>	**CLK_CTL_GPIOSRC[1:0]**: GPIO debounce clock source selection 2’b00: 16MHz RC Oscillator 2’b01: 48MHz XTAL oscillator 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	R/W
<11:10>	**CLK_CTL_TIMB2SRC[1:0]**:Basic Timer Channel 5 ~ 8 Clock source selection 2’b00: PCLK 2’b01: HCLK 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	R/W
<9:8>	**CLK_CTL_I2CSRC[1:0]**: I2C clk source selection 2’b00: 16MHz RC Oscillator 2’b01: 48MHz XTAL oscillator 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	R/W
<7:6>	**CLK_CTL_TIMB1SRC[1:0]**:Basic Timer Channel 1 ~ 4 Clock source selection 2’b00: PCLK 2’b01: HCLK 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	R/W
<5:4>	**CLK_CTL_BRSRC[1:0]**: Baud Rate Clock source selection 2’b00: 16MHz RC Oscillator 2’b01: 48MHz XTAL oscillator 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	R/W
<3:2>	**CLK_CTL_WDTWSRC[1:0]**: Window watch-dog timer Clock source selection 2’b00: PCLK 2’b01: HCLK 2’b10: Output clock of MCU_PLL (16~96MHz, 2MHz/step) 2’b11: Reserved	R/W
<1:0>	**CLK_CTL_HCLKSRC[1:0]**: HCLK clock source selection 2’b00: 16MHz RC Oscillator 2’b01: 4 - 48MHz XTAL oscillator 2’b10: Output clock of MCU_PLL (16~96MHz, 1MHz/step) 2’b11: 32kHz clock	R/W

Note: The baudrate clock generator is shared by SPI. We suggest the clock source (BR_SRC) is the same as system clock (HCLK_SRC).

5.4.2 ICG_HCLK_DIV

Base address: 0x8001_0000
Address offset: 0x04
Type: Read/Write
Reset value: 0x0000_0001

Bit	Description	Type
<31:0>	**HCLK_DIV_DIV[31:0]**: Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.3 ICG_PCLK_DIV

Base address: 0x8001_0000
Address offset: 0x08
Type: Read/Write
Reset value: 0x0000_0002

Bit	Description	Type
<31:0>	**PCLK_DIV_DIV[31:0]**: Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.4 ICG_DMA_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_0001

Bit	Description	Type
<31:0>	**DMA_DIV_DIV[31:0]**:Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.5 ICG_BR_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x10
Type: Read/Write
Reset value: 0x0000_0002

Bit	Description	Type
<31:0>	**BR_CLK_DIV_DIV[31:0]**: Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

Note: The baudrate clock generator is shared by SPI. We suggest the divider is fixed in default value (1).

5.4.6 ICG_I2C1_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x14
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**I2C1_CLK_DIV_DIV[31:0]**: Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.7 ICG_I2C2_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x18
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**I2C2_CLK_DIV_DIV[31:0]**: Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.8 ICG_GPIO_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x1C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**GPIO_CLK_DIV_DIV[31:0]**: Clock divider value 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.9 ICG_RC16M_CTRL

Base address: 0x8001_0000
Address offset: 0x20
Type: Read/Write
Reset value: 0x0020_0092

Bit	Description	Type
<31>	**RC16M_CTRL_CALI**: calibrate 16MHz RCOSC 1'b0: normal mode; 1'b1: calibrate mode.	R/W
<30>	Reserved	R/W
<29:22>	**TEST ONLY - clkdiv[7:0] to clock test mux**	R/W
<21:20>	**RC16M_CTRL_OPAM_ISEL[1:0]**: default 2'b10	R/W
<19:18>	Reserved	R/W
<17:16>	**RC16M_CTRL_COMP_ISEL[1:0]**: default 2'b00	R/W
<15:13>	Reserved	R/W
<12>	**RC16M_CTRL_CLKOUT_EN**:	R/W
<11>	**RC16M_CTRL_CLK_TRIM_SRC**:	R/W
<10:8>	**RC16M_CTRL_TST_SRC[2:0]**:	R/W
<7:0>	**RC16M_CTRL_CRES[7:0]**: 16MHz RCOSC cap tune control. Default: 8'h92	R/W

5.4.10 ICG_PLL_DIV

Base address: 0x8001_0000
Address offset: 0x24
Type: Read/Write
Reset value: 0xD147_2200

Bit	Description	Type
<31>	**PLL_RBFBDIV**: rb_fbdiv 1'b0: reset 1'b1: normal operation	R/W
<30>	**PLL_RBPOSTDIV**: rb_postdiv 1'b0: reset 1'b1: normal operation	R/W
<29:28>	**PLL_VCOBIAS[1:0]**: vcobias_ctrl<1:0> vco bias current control, default 2'b01	R/W
<27>	Reserved	R/W
<26:20>	**PLL_VCOCAP[6:0]**: vcocap_ctrl<6:0> manual vco cap array setting, default 8'h14	R/W
<19:16>	**PLL_PREDIV[3:0]**: Pre-divider of MCU_PLL ref_div<3:0> 4'b0000: ref_clk/1 4'b0010: ref_clk/2	R/W

Bit	Description	Type
	4’b0001: ref_clk/3 4’b0110: ref_clk/4 4’b0011: ref_clk/6 4’b1110: ref_clk/8 4’b0111: ref_clk/12 (default) 4’b1111: ref_clk/24	
<15:14>	Reserved	R/W
<13:8>	**PLL_FBDIV[5:0]**: Feedback divider of MCU_PLL fb_div<5:0> default: 6’b100010	R/W
<7:4>	Reserved	R/W
<3:0>	**PLL_POSTDIV[3:0]**: Post divider of MCU_PLL post_div<3:0> 4’b0000: vco_clk/4 (default) 4’b0001: vco_clk/6 4’b0110: vco_clk/8 4’b0011: vco_clk/12 4’b1110: vco_clk/16 4’b0111: vco_clk/24 4’b1111: vco_clk/48	R/W

PLL divider setting shows in the follow table.

Table 11 PLL Output Clock Frequency Configuration

ref clk (MHz)	fb_div<5:0>	post_div<3:0>	MCU clock Freq (MHz)
4	101010	0000	106
4	101001	0000	105
4	101000	0000	104
4	100111	0000	103
4	100110	0000	102
4	100101	0000	101
4	100100	0000	100
4	100011	0000	99
4	100010	0000	98
4	100001	0000	97
4	100000	0000	96
4	011111	0000	95
4	011110	0000	94
4	011101	0000	93
4	011100	0000	92
4	011011	0000	91
4	011010	0000	90
4	011001	0000	89
4	011000	0000	88
4	010111	0000	87
4	010110	0000	86
4	010101	0000	85
4	010100	0000	84
4	010011	0000	83
4	010010	0000	82
4	010001	0000	81
4	010000	0000	80
4	001111	0000	79
4	001110	0000	78

ref clk (MHz)	fb_div<5:0>	post_div<3:0>	MCU clock Freq (MHz)
4	001101	0000	77
4	001100	0000	76
4	001011	0000	75
4	001010	0000	74
4	001001	0000	73
4	001000	0000	72
4	000111	0000	71
4	000110	0000	70
4	000101	0000	69
4	000100	0000	68
4	000011	0000	67
4	000010	0000	66
4	000001	0000	65
4	100000	0001	64
4	011101	0001	62
4	011010	0001	60
4	010111	0001	58
4	010100	0001	56
4	010001	0001	54
4	001110	0001	52
4	001011	0001	50
4	100010	0110	49
4	100000	0110	48
4	011110	0110	47
4	011100	0110	46
4	011010	0110	45
4	011000	0110	44
4	010110	0110	43
4	010100	0110	42
4	010010	0110	41
4	010000	0110	40
4	001110	0110	39

ref clk (MHz)	fb_div<5:0>	post_div<3:0>	MCU clock Freq (MHz)
4	001100	0110	38
4	001010	0110	37
4	001000	0110	36
4	000110	0110	35
4	000100	0110	34
4	000010	0110	33
4	100000	0011	32
4	011101	0011	31
4	011010	0011	30
4	010111	0011	29
4	010100	0011	28
4	010001	0011	27
4	001110	0011	26
4	001011	0011	25
4	100000	1110	24
4	011100	1110	23
4	011000	1110	22
4	010100	1110	21
4	010000	1110	20
4	001100	1110	19
4	001000	1110	18
4	000100	1110	17
4	100000	0111	16
4	011010	0111	15
4	010100	0111	14
4	001110	0111	13
4	001000	0111	12
4	000010	0111	11

5.4.11 ICG_PLL_PD_CTRL

Base address: 0x8001_0000
Address offset: 0x28
Type: Read/Write
Reset value: 0x0000_0080

Bit	Description	Type
<31:14>	Reserved	R/W
<13>	**PLL_PLLCLKTST_EN**: en_pllclk_test enable the test buffer of PLL clock output	R/W
<12>	**PLL_VCTST_EN**: en_vctest enable the test buffer of VCO input voltage for testing	R/W
<11:9>	Reserved	R/W
<8>	**PLL_VCOCAL_AUTOB**: vcocal_autob 1'b0: VCO cap array setting from VCO calibration circuit 1'b1: VCO cap array setting from SFR Register vcocap_ctrl_in<3:0>	R/W
<7>	**PLL_VCOCAL_EN**: vcocal_en default: 1 rising edge to enable the VCO calibration	R/W
<6>	**PLL_PD_VCOCOMP**: pd_vcocal_comparator power down Comparator in analog part of VCO calibration	R/W
<5>	**PLL_PD_VCOBUF**: pd_vcocal_vco_buffer power down VCO buffer in analog part of VCO calibration	R/W
<4>	**PLL_PD_VCOCAL**: pd_vcocal power down VCO calibration, both analog and digital circuit	R/W
<3>	Reserved	R/W
<2>	**PLL_PD_PFD**: pd_pfd power down of Phase detector	R/W
<1>	**PLL_PD_VCO**: pd_vco power down of VCO	R/W
<0>	**PLL_PD_CP**: pd_cp power down of charge pump	R/W

5.4.12 ICG_PLL_LFCFG_CTRL

Base address: 0x8001_0000
Address offset: 0x2C
Type: Read/Write
Reset value: 0x0900_A9D3

Bit	Description	Type
<31:29>	Reserved	R/W
<28>	**PLL_SW_VAR_CTRL**: sw_var_ctrl default 0 option to increase Kvco	R/W
<27:25>	**PLL_PFD_SLCT**: pfd_sel<2:0> the min time control of both CP UP & DN is on 3'b100	R/W
<24>	**PLL_CLKSRC**: ref_sel, reference clock source selection default 1 1'b0: 16MHz RC 1'b1: XTAL	R/W
<23:0>	**PLL_IF_CTRL[23:0]**: lf_ctrl<23:0> loop filter control, default 0x00A9D3	R/W

5.4.13 ICG_PLL_VCOCAL_CTRL

Base address: 0x8001_0000
Address offset: 0x30
Type: Read/Write
Reset value: 0x0511_4160

Bit	Description	Type
<31:27>	Reserved	R/W
<26:24>	**VCOCAL_CAL_ITVL[2:0]**: vcocal_cal_interval<2:0> the interval between each VCO calibration 3'b000: run calibration continuously, 3'b001: run calibration one time 3'b010: 10ms, 3'b011: 100ms, 3'b100: 500ms, 3'b101: 1000ms (default) 3'b110: 5000ms, 3'b111: 10000ms	R/W
<23>	Reserved	R/W
<22:20>	**VCOCAL_STB_TM[2:0]**: vcocal_pll_stable_time<2:0> 3'b000: 32us, 3'b001: 48us (default), 3'b010: 64us, 3'b011: 96us 3'b100: 128us, 3'b101: 160us, 3'b110: 192us, 3'b111: 256us	R/W
<19>	Reserved	R/W
<18:12>	**VCOCAL_INI_VAL[6:0]**: vcocal_init_vcocal_value<6:0> default: 7'b0010100	R/W

Bit	Description	Type
<11>	Reserved	R/W
<10:8>	**VCOCAL_IN_H_RNG[2:0]**: vcocal_vcoin_high_range<2:0> default 001 the expected upper limit of the VCO input during locking (000=0.66*vdd,001=0.68*vdd,….111=0.8*vdd)	R/W
<7>	Reserved	R/W
<6:4>	**VCOCAL_IN_L_RNG[2:0]**: vcocal_vcoin_low_range<2:0> default 110 the expected lower limit of the VCO input during locking (000=0.2*vdd, 001=0.22*vdd,…,111=0.34*vdd)	R/W
<3:2>	**VCOCAL_COMP_VH[1:0]**: vcocal_comp_vh_ctrl<1:0> default 00 2'b00: no hysteresis, 2'b01: smallest hysteresis 2'b10: 2'b11: largest hysteresis	R/W
<1:0>	**VCOCAL_COMP_VL[1:0]**: vcocal_comp_vl_ctrl<1:0> default 00 2'b00: no hysteresis , 2'b01: smallest hysteresis 2'b10: 2'b11: largest hysteresis	R/W

5.4.14 ICG_PLL_IREF_CTRL

Base address: 0x8001_0000
Address offset: 0x34
Type: Read/Write
Reset value: 0x0000_0043

Bit	Description	Type
<31>	**IREF_PLL_STAT**: Pll lock status. 1'b0: unlocked 1'b1: locked	R
<30:24>	**IREF_VCOCAP_WR[6:0]**: write vcocap_ctrl_out2spi<6:0> to sfr register	R/W
<23>	Reserved	R/W
<22:16>	**IREF_VCOCAP_RD[6:0]**: direct read out vcocap_ctrl_out2spi<6:0> read-only	R/W
<15:8>	Reserved	R/W
<7>	Reserved	R/W
<6:4>	**IREF_CTRL[2:0]**: cp_iref_ctrl<2:0> 3'b000: 2.5uA (max) … 3'b100: 2uA (default) … 3'b111: 1.74uA (min)	R/W
<3>	Reserved	R/W
<2:0>	**IREF_OCCTRL[2:0]**: cp_outcurrent_ctrl<2:0> 3'b000: 10uA, 3'b001: 25uA, 3'b010: 40uA ….. 3'b111: 115uA (default 3'b011)	R/W

5.4.15 ICG_TIMER_ADV_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x40
Type: Read/Write
Reset value: 0x0000_000F

Bit	Description	Type
<31:0>	**TIMACLK_DIV[31:0]**: Divider 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.16 ICG_TIMER_B1_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x44
Type: Read/Write
Reset value: 0x0000_000F

Bit	Description	Type
<31:0>	**TIMB1_DIV[31:0]**: Divider 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.17 ICG_TIMER_B2_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x48
Type: Read/Write
Reset value: 0x0000_000F

Bit	Description	Type
<31:0>	**TIMB2_DIV[31:0]**: Divider 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.18 ICG_WDT_W_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x4C
Type: Read/Write
Reset value: 0x0000_000F

Bit	Description	Type
<31:0>	**WDTW_DIV[31:0]**: Divider 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.19 ICG_TIMER_EXCLK_ENABLE_1

Base address: 0x8001_0000
Address offset: 0x50
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:18>	Reserved	R/W
<17>	**TMXC_ATM2SRC_ETR**: Advanced timer ETR external input selection 1'b0: internal clock 1'b1: external input	R/W
<16>	**TMXC_ATM2SRC_CH4**: Advanced timer channel 4 external input selection 1'b0: internal clock 1'b1: external input	R/W
<15>	**TMXC_ATM2SRC_CH3**: Advanced timer channel 3 external input selection 1'b0: internal clock 1'b1: external input	R/W
<14>	**TMXC_ATM2SRC_CH2**: Advanced timer channel 2 external input selection 1'b0: internal clock 1'b1: external input	R/W
<13>	**TMXC_ATM2SRC_CH1**: Advanced timer channel 1 external input selection 1'b0: internal clock 1'b1: external input	R/W
<12>	**TMXC_ATM1SRC_ETR**: Advanced timer ETR external input selection 1'b0: internal clock 1'b1: external input	R/W
<11>	**TMXC_ATM1SRC_CH4**: Advanced timer channel 4 external input selection 1'b0: internal clock	R/W

Bit	Description	Type
	1'b1: external input	
<10>	**TMXC_ATM1SRC_CH3**: Advanced timer channel 3 external input selection 1'b0: internal clock 1'b1: external input	R/W
<9>	**TMXC_ATM1SRC_CH2**: Advanced timer channel 2 external input selection 1'b0: internal clock 1'b1: external input	R/W
<8>	**TMXC_ATM1SRC_CH1**: Advanced timer channel 1 external input selection 1'b0: internal clock 1'b1: external input	R/W
<7>	**TMXC_BTMSRC_24**: Basic timer 24, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<6>	**TMXC_BTMSRC_23**: Basic timer 23, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<5>	**TMXC_BTMSRC_22**: Basic timer 22, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<4>	**TMXC_BTMSRC_21**: Basic timer 21, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<3>	**TMXC_BTMSRC_14**: Basic timer 14, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<2>	**TMXC_BTMSRC_13**: Basic timer 13, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<1>	**TMXC_BTMSRC_12**: Basic timer 12, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W
<0>	**TMXC_BTMSRC_11**: Basic timer 11, external input clock selection. 1'b0: internal clock 1'b1: external clock	R/W

5.4.20 ICG_TIMER_EXCLK_ENABLE_2

Base address: 0x8001_0000
Address offset: 0x54
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:20>	Reserved	R/W
<19>	**TMXC_GTM4SRC_ETR**: General purpose timer ETR external input selection 1'b0: internal clock 1'b1: external input	R/W
<18>	**TMXC_GTM4SRC_CH4**: General purpose timer channel 4 external input selection 1'b0: internal clock 1'b1: external input	R/W
<17>	**TMXC_GTM4SRC_CH3**: General purpose timer channel 3 external input selection 1'b0: internal clock 1'b1: external input	R/W
<16>	**TMXC_GTM4SRC_CH2**: General purpose timer channel 2 external input selection 1'b0: internal clock 1'b1: external input	R/W
<15>	**TMXC_GTM4SRC_CH1**: General purpose timer channel 1 external input selection 1'b0: internal clock 1'b1: external input	R/W
<14>	**TMXC_GTM3SRC_ETR**: General purpose timer ETR external input selection 1'b0: internal clock 1'b1: external input	
<13>	**TMXC_GTM3SRC_CH4**: General purpose timer channel 4 external input selection 1'b0: internal clock 1'b1: external input	
<12>	**TMXC_GTM3SRC_CH3**: General purpose timer channel 3 external input selection 1'b0: internal clock 1'b1: external input	R/W
<11>	**TMXC_GTM3SRC_CH2**: General purpose timer channel 2 external input selection 1'b0: internal clock	R/W

Bit	Description	Type
	1’b1: external input	
<10>	**TMXC_GTM3SRC_CH1**: General purpose timer channel 1 external input selection 1’b0: internal clock 1’b1: external input	R/W
<9>	**TMXC_GTM2SRC_ETR**: General purpose timer ETR external input selection 1’b0: internal clock 1’b1: external input	R/W
<8>	**TMXC_GTM2SRC_CH4**: General purpose timer channel 4 external input selection 1’b0: internal clock 1’b1: external input	R/W
<7>	**TMXC_GTM2SRC_CH3**: General purpose timer channel 3 external input selection 1’b0: internal clock 1’b1: external input	R/W
<6>	**TMXC_GTM2SRC_CH2**: General purpose timer channel 2 external input selection 1’b0: internal clock 1’b1: external input	R/W
<5>	**TMXC_GTM2SRC_CH1**: General purpose timer channel 1 external input selection 1’b0: internal clock 1’b1: external input	R/W
<4>	**TMXC_GTM1SRC_ETR**: General purpose timer ETR external input selection 1’b0: internal clock 1’b1: external input	R/W
<3>	**TMXC_GTM1SRC_CH4**: General purpose timer channel 4 external input selection 1’b0: internal clock 1’b1: external input	R/W
<2>	**TMXC_GTM1SRC_CH3**: General purpose timer channel 3 external input selection 1’b0: internal clock 1’b1: external input	R/W
<1>	**TMXC_GTM1SRC_CH2**: General purpose timer channel 2 external input selection 1’b0: internal clock 1’b1: external input	R/W
<0>	**TMXC_GTM1SRC_CH1**: General purpose timer channel 1 external input selection	R/W

Bit	Description	Type
	1’b0: internal clock 1’b1: external input	

5.4.21 ICG_ADC_CLK_DIV

Base address: 0x8001_0000
Address offset: 0x58
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ADCCLK_DIV[31:0]**:Divider 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

5.4.22 ICG register map

The base address of ICG registers is 0x8001_0000.
The following table gives the ICG register map and reset values.

Table 12 ICG register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CLK_CFG		CLK_CTL_ADCSRC[1:0]	CLK_CTL_TEST_OUT	CLK_CTL_HSXTAL_CRES				CLK_CTL_HSXTAL_CTUNE[3:0]				CLK_CTL_HSXTAL_ITUNE[3:0]	CLK_CTL_PLL	HS_XTAL_OUT_PSPHS_XTACLK_CTL_HSXTALOUT	CLK_CTL_HSXTAL	CLK_CTL_HSRC		CLK_CTL_TIMAGSRC[1:0]		CLK_CTL_GPIOSRC[1:0]			CLK_CTL_TIMB2SRC[1:0]		CLK_CTL_I2CSRC[1:0]		CLK_CTL_TIMB1SRC[1:0]		CLK_CTL_BRSRC[1:0]		CLK_CTL_WDTWSRC[1:0]		CLK_CTL_HCLKSRC[1:0]
	Reset value	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	HCLK_DIV	HCLK_DIV_DIV[31:0]																															
	Reset value	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	1
0x08	PCLK_DIV	PCLK_DIV_DIV[31:0]																															
	Reset value	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	1	0
0x0C	DMA_CLK_DIV	DMA_DIV_DIV[31:0]																															
	Reset value	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	1
0x10	BR_DIV	BR_CLK_DIV_DIV[31:0]																															
	Reset value	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	1	0
0x14	I2C1_DIV	I2C1_CLK_DIV_DIV[31:0]																															
	Reset value	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
0x18	I2C2_DIV	I2C2_CLK_DIV_DIV[31:0]																															
	Reset value	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
0x1C	GPIO_DIV	GPIO_CLK_DIV_DIV[31:0]																															
	Reset value	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
0x20	RC16M_CTRL	RC16M_CTRL_CALI	Res								RC16M_CTRL_TST_DIV[7:0]		RC16M_CTRL_OPAM_ISEL[1:0]	Res	Res		RC16M_CTRL_COMP_ISEL[1:0]	Res	Res	Res	RC16M_CTRL_CLKOUT_EN	RC16M_CTRL_CLK_TRIM_SRC			RC16M_CTRL_TST_SRC[2:0]									RC16M_CTRL_CRES[7:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
0x24	PLL_DIV	PLL_RBFBDIV	PLL_RBPOSTDIV		PLL_VCOBIAS[1:0]	Res							PLL_VCOCAP[6:0]				PLL_PREDIV[3:0]	Res	Res						PLL_FBDIV[5:0]	Res	Res	Res	Res					PLL_POSTDIV[3:0]
	Reset value	1	1	0	1	0	0	0	1	0	1	0	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	PLL_PD_CTRL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PLL_PLLCLKTST_EN	PLL_VCTST_EN	Res	Res	Res	PLL_VCOCAL_AUTOB	PLL_VCOCAL_EN	PLL_PD_VCOCOMP	PLL_PD_VCOBUF	PLL_PD_VCOCAL	Res	PLL_PD_PFD	PLL_PD_VCO	PLL_PD_CP
	Reset value	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	0	0	0	0	0
0x2C	PLL_LFCFG_CTRL	Res	Res	Res	PLL_SW_VAR_CTRL			PLL_PFD_SLCT[2:0]	PLL_CLKSRC																								PLL_IF_CTRL[23:0]
	Reset value	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	1
0x30	PLL_VCOCAL_CTRL	Res	Res	Res	Res	Res			VCOCAL_CAL_ITVL[2:0]	Res			VCOCAL_STB_TM[2:0]	Res							VCOCAL_INI_VAL[6:0]	Res			VCOCAL_IN_H_RNG[2:0]	Res			VAOCAL_IN_L_RNG[2:0]		VCOCAL_COMPVH[1:0]		VCOCAL_COMPVL[1:0]
	Reset value	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0
0x34	PLL_IREF_CTRL	IREF_PLL_STAT							IREF_VCOCAP_WR[6:0]	Res							IREF_VCOCAP_RD[6:0]	Res	Res	Res	Res	Res	Res	Res	Res	Res			IREF_CTRL[2:0]	Res			IREF_OCCTRL[2:0]
	Reset value	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	0	0	1	1
0x40	TIMER_ADV_CLK_DIV	TIMACLK_DIV[31:0]																															
	Reset value	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	1	1	1	1
0x44	TIMER_B1_CLK_DIV	TIMB1_DIV[31:0]																															
	Reset value	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	1	1	1	1
0x48	TIMER_B2_CLK_DIV	TIMB2_DIV[31:0]																															
	Reset value	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	1	1	1	1
0x4C	WDT_W_CLK_DIV	WDTW_DIV[31:0]																															
	Reset value	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	1	1	1	1
0x50	TIMER_EXCLK_EN_1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TMXC_ATM2SRC_ETR	TMXC_ATM2SRC_CH4	TMXC_ATM2SRC_CH3	TMXC_ATM2SRC_CH2	TMXC_ATM2SRC_CH1	TMXC_ATM1SRC_ETR	TMXC_ATM1SRC_CH4	TMXC_ATM1SRC_CH3	TMXC_ATM1SRC_CH2	TMXC_ATM1SRC_CH1	TMXC_BTMSRC_24	TMXC_BTMSRC_23	TMXC_BTMSRC_22	TMXC_BTMSRC_21	TMXC_BTMSRC_14	TMXC_BTMSRC_13	TMXC_BTMSRC_12	TMXC_BTMSRC_11
	Reset value	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
0x54	TIMER_EXCLK_EN_1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TMXC_GTM4SRC_ETR	TMXC_GTM4SRC_CH4	TMXC_GTM4SRC_CH3	TMXC_GTM4SRC_CH2	TMXC_GTM4SRC_CH1	TMXC_GTM3SRC_ETR	TMXC_GTM3SRC_CH4	TMXC_GTM3SRC_CH3	TMXC_GTM3SRC_CH2	TMXC_GTM3SRC_CH1	TMXC_GTM2SRC_ETR	TMXC_GTM2SRC_CH4	TMXC_GTM2SRC_CH3	TMXC_GTM2SRC_CH2	TMXC_GTM2SRC_CH1	TMXC_GTM1SRC_ETR	TMXC_GTM1SRC_CH4	TMXC_GTM1SRC_CH3	TMXC_GTM1SRC_CH2	TMXC_GTM1SRC_CH1
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x58	ADC_CLK_DIV	ADCCLK_DIV[31:0]																															
	Reset value	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

6 Always on block (AON)

6.1 Introduction

The always on module (AON) contains 32KHz RC oscillator, 32KHz crystal oscillator, independent WDT, RTC .

6.2 AON main features

6.3 PMU

the internal voltage regulators, power on reset, programmable voltage detector, and deep sleep mode related control logic

6.3.1 Programmable Voltage Detector (PVD)

6.3.2 PMU

6.3.2.1 Introduction

The XT32H053 devices require a 1.8 V to 5.5 V external power supply as operating supply source, and the PMU is combined with LDOs, IVREF, PVD, POR modules.
Four embedded linear voltage regulators (LDOs) are used to supply internal core circuits, and IVREF module offers voltage and current references for internal analog circuits (ADC, CTSM, LDOs etc.). An embedded programmable voltage detector (PVD) monitors the VSUP power supply and compares it to V_{PVD} threshold, and an integrated core voltage power-on/power-down (POR/PDR) is available for ensuring proper operation upon power-on and power-down.

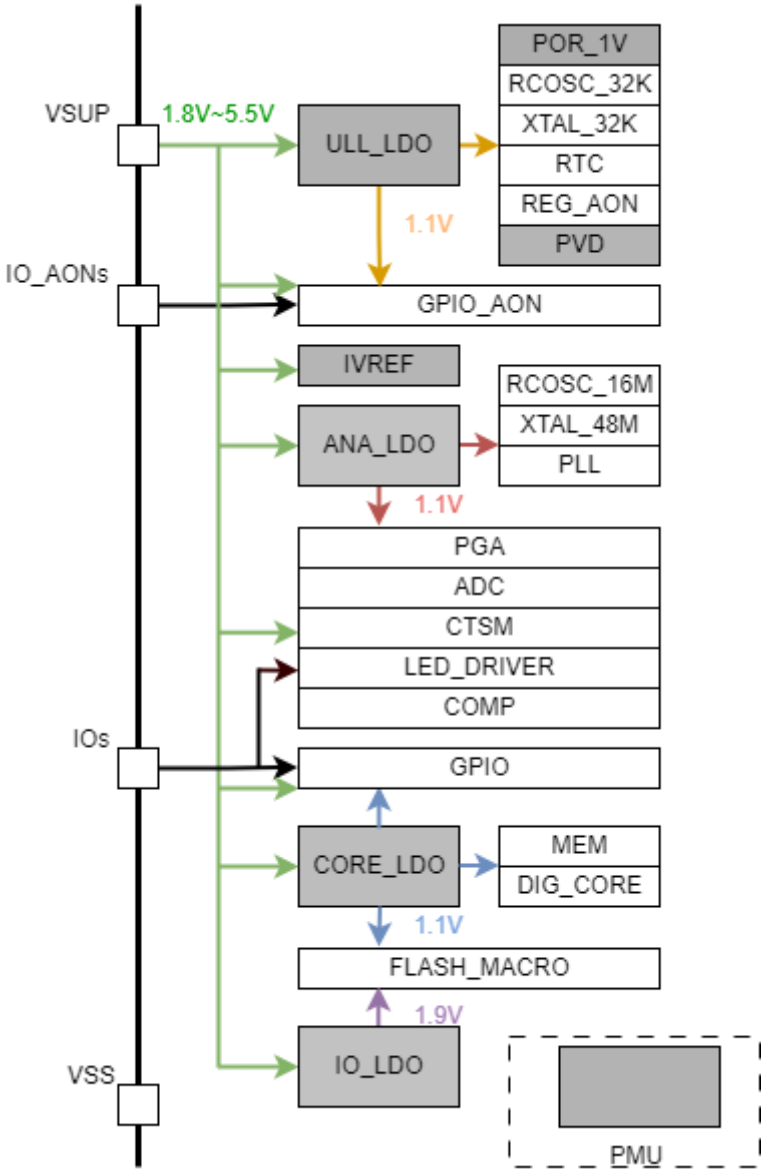


Figure 11 PMU overview

6.3.2.2 PMU main features

-Only one external supply input;
-Four embedded LDOs;
-Internal current and voltage references are generated;
-Embedded programmable voltage detector (PVD) monitor is available.

6.3.2.3 PMU functions

6.3.2.3.1 IVREF tune

1. The voltage references for LED driver, PVD, 16MHz RC OSC, touch sensor, ADC, LDO can be tunning with 3 bits control.
2. The control bits default value is b0011, the tunning step is 10mV, from b0000 to b0111, 8 steps are available and increasing.

6.3.2.3.2 LDO tune

1. The core_LDO and ana_LDO can be tunning with 3 bits control.
2. The control bits default value is b101, and output voltage is 1.1V. The tunning step is 50mV, from b001 to b0111, 7 steps are available and increasing, b001 is to 0.9V and b111 is 1.2V.

6.3.2.3.3 PVD trigger point setting

1. The power voltage detector trigger point can be setting with 5 bits control.
2. The lowest trigger point voltage is 1.7V, the setting value is b00000; the maximum trigger voltage is 4.8V, the setting value is b11111. The tunning step is 100mV, from b00000 to b11111, each step also has hysteresis voltage 50mV for falling and rising change.

6.3.3 Low speed clock

1. The 32K crystal oscillator has 4 bits capacitors tunning control and 4 bits current bias tunning control.
2. The 32K RC oscillator has 6 bits frequency tuning control and its output frequency can be calibrated to 32KHz.

6.4 AON functional description

6.4.1 Generate clock

3. Power on low speed clock source
4. Set RTC divider

6.4.2 Deep sleep mode

XT32H05x serial can run in deep sleep mode. The system can be waked up by timer or external signal from port A.9.

System entering deep sleep mode performs the following steps:

1. Config DEEPSLEEP_TIME to write deep sleep time.
2. Set DEEPSLEEP_CTRL.BIT[0] to enter deep sleep mode and enable or disable external wakeup by set DEEPSLEEP_CTRL.BIT[4].
3. When reach to setting deep sleep time or received valid external wake up signal, system will be waked up and enter normal mode.

6.4.3 Low speed RC oscillator calibration

Refer to session 8.3.7.

6.5 AON registers

6.5.1 AON_CFG

Base address: 0x4000_3600
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	**CFG_RTCRST**: Soft reset of RTC module 1'b0: reset 1'b1: normal operation	R/W
<30>	**CFG_WDTI_EN**: Independent Watch-dog enable control 1'b0: disable 1'b1: enable	R/W
<29>	Reserved	R/W
<28>	**CFG_WDTI_SPD**: Independent Watch-dog speed up control 1'b0: disable 1'b1: enable	R/W
<27:2>	Reserved	R/W
<1:0>	**CFG_32KCLKSRC[1:0]**: Root 32k clock source selection 2'b00: 32k RC clock 2'b01: 32k XTAL clock 2'b10: Reserved 2'b11: Reserved	R/W

6.5.2 AON_RTC_REFCLK_DIV

Base address: 0x4000_3600
Address offset: 0x04
Type: Read/Write
Reset value: 0x0008_0000

Bit	Description	Type
<31:0>	**RTCREFCLK_DIV_DIV[31:0]**: Divide the 32k reference clock by 32768, generate 1Hz clock for RTC 0: divided by 1 1: divided by 1 2: divided by 2 … n: divided by n	R/W

6.5.3 AON_CLK_CTRL

Base address: 0x4000_3600
Address offset: 0x08
Type: Read/Write
Reset value: 0x0020_8302

Bit	Description	Type
<31:22>	Reserved	R/W
<21:16>	**CLKCTRL_32KRC_RESTUNE[5:0]**: res_tune<5:0> of 32k RC Oscillator default 100000 6 bits charge current control by resisitor series: 111111=4.3M ohm, 000000=10.5M ohm, it is to tunning the 32k clock output frequency	R/W
<15:12>	**CLKCTRL_32KXTAL_CAPTUNE[3:0]**: cap_tune<3:0> of 32k XTAL oscillator default 1000 7.5p loading 4 bits crystal capacity loading setting: 12p(fixed)+8p+4p+2p+1p	R/W
<11:8>	**CLKCTRL_32KXTAL_ITUNE[3:0]**: i_tune<3:0> of 32k XTAL Oscillator default 0011 4 bits current bias control for osc core: 100nA+200nA+400nA+800nA, default 200nA	R/W
<7:5>	Reserved	R/W
<4>	**CLKCTRL_32K_CAL**: 32K RCOSC calibration mode 1'b0: normal operation 1'b1: calibration mode	R/W
<3:2>	Reserved	R/W
<1>	**CLKCTRL_32KXTAL_PD**: Power down control of 32k XTAL	R/W
<0>	**CLKCTRL_32KRC_PD:** Power down control of 32k RCOSC	R/W

6.5.4 AON_LDO_CFG

Base address: 0x4000_3600
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0355_0000

Bit	Description	Type
<31:30>	Reserved	R/W
<29>	**LDO_CFG_EXTLDO_EN**: extra enable LDO IO 1'b0: normal operation 1'b1: more current for error amplifier	R/W
<28>	**LDO_CFG_LDOIO_BP**: Bypass LDO IO	R/W
<27:24>	**LDO_CFG_IO_OTUNE[3:0]**: Output voltage tune control of LDO IO default 0011 4'b0000: 1.73V,	R/W

Bit	Description	Type
	4’b0001: 1.78V, 4’b0010: 1.84V, 4’b0011: 1.92V output voltage, 4’b0100: 1.96V, 4’b0101: 2.00V, 4’b0110: 2.06V, 4’b0111: 2.11V 4’b1000: 2.17V, 4’b1001: 2.23V, 4’b1010: 2.29V, 4’b1011: 2.34V, 4’b1100: 2.40V, 4’b1101: 2.44V, 4’b1110: 2.49V, 4’b1111: 2.55V	
<23>	Reserved	R/W
<22:20>	**LDO_CFG_ANALOG_OTUNE[2:0]**: Output voltage tune control of LDO Analog 3’b001: 0.9V, 0.05V per step, 3’b010: 0.95V 3’b011: 1.00V 3’b100: 1.05V 3’b101: 1.10V default output voltage 3’b110: 1.15V 3’b111: 1.20V	R/W
<19>	Reserved	R/W
<18:16>	**LDO_CFG_CORE_OTUNE[2:0]**: Output voltage tune control of LDO core 3’b001: 0.9V, 0.05V per step, 3’b010: 0.95V 3’b011: 1.00V 3’b100: 1.05V 3’b101: 1.10V default output voltage 3’b110: 1.15V 3’b111: 1.20V	R/W
<15:13>	Reserved	R/W
<12:8>	**LDO_CFG_VOLDET_TRG[4:0]**: Voltage Detector trigger point setting control 5’b00000: 1.7V, 0.1V per step, default trigger point voltage 5’b00001: 1.8V 5’b00010: 1.9V 5’b00011: 2.0V 5’b00100: 2.1V	R/W

Bit	Description	Type
	5’b00101: 2.2V 5’b00110: 2.3V 5’b00111: 2.4V 5’b01000: 2.5V 5’b01001: 2.6V 5’b01010: 2.7V 5’b01011: 2.8V 5’b01100: 2.9V 5’b01101: 3.0V 5’b01110: 3.1V 5’b01111: 3.2V 5’b10000: 3.3V 5’b10001: 3.4V 5’b10010: 3.5V 5’b10011: 3.6V 5’b10100: 3.7V 5’b10101: 3.8V 5’b10110: 3.9V 5’b10111: 4.0V 5’b11000: 4.1V 5’b11001: 4.2V 5’b11010: 4.3V 5’b11011: 4.4V 5’b11100: 4.5V 5’b11101: 4.6V 5’b11110: 4.7V 5’b11111: 4.8V	
<5>	**LDO_CFG_VOLDET_PD**: Power down control of voltage detector	R/W
<4>	Reserved	R/W
<3>	**LDO_CFG_IO_PD**: Power down control of LDO IO	R/W
<2>	**LDO_CFG_ANALOG_PD**: Power down control of LDO Analog	R/W
<1>	**LDO_CFG_CORE_PD**: Power down control of LDO Core	R/W
<0>	**LDO_CFG_IVREF_PD**: Power down control of IVREF	R/W

6.5.5 AON_IVREF_CFG_1

Base address: 0x4000_3600
Address offset: 0x10
Type: Read/Write
Reset value: 0x3333_3333

Bit	Description	Type
<31:28>	**IVREF_CFG_1_LED_REF[3:0]**: 0.8v reference voltage for LED driver 4’b0000: 0.57V, 10mV steps 4’b0011: 0.6V (default) 4’b0111: 0.64V	R/W
<27:24>	**IVREF_CFG_1_PVD_REF[3:0]**: Tune control of 0.36v reference voltage for PVD 4’b0000: 0.33V, 10mV steps 4’b0001: 0.34V 4’b0010: 0.35V 4’b0011: 0.36V 4’b0100: 0.37V 4’b0101: 0.38V 4’b0110: 0.39V 4’b0111: 0.40V (default) 4’b1000: 0.41V 4’b1001: 0.42V 4’b1010: 0.43V 4’b1011: 0.44V 4’b1100: 0.45V 4’b1101: 0.46V 4’b1110: 0.47V 4’b1111: 0.48V	R/W
<23:0>	Reserved	R/W

6.5.6 AON_IVREF_CFG_2

Base address: 0x4000_3600
Address offset: 0x14
Type: Read/Write
Reset value: 0x3333_3333

Bit	Description	Type
<31:0>	Reserved	R/W

6.5.7 AON_IVREF_CFG_3

Base address: 0x4000_3600
Address offset: 0x18
Type: Read/Write
Reset value: 0x3333_3333

Bit	Description	Type
<31:16>	Reserved	R/W
<15:12>	**IVREF_CFG_3_HSI_REF[3:0]**: Tune control of 0.8v reference voltage for 16MHz (RCOSC) 4'b0000: 0.67V, 10mV steps 4'b0001: 0.68V 4'b0010: 0.69V 4'b0011: 0.70V (default) 4'b0111: 0.74V	R/W
<11:8>	**IVREF_CFG_3_TSU_REF[3:0]**: Tune control of 0.8v reference voltage for touch sensor 4'b0000: 0.77V, 10mV steps 4'b0111: 0.84V 4'b0011: 0.8V (default)	R/W
<7:4>	Reserved	R/W
<3:0>	**IVREF_CFG_3_DAC_REF[3:0]**: Tune control of 1.22v reference voltage (DAC) 4'b0000: 0.57V, 10mV steps 4'b0111: 0.64V 4'b0011: 0.6V (default)	R/W

6.5.8 AON_DEEPSLEEP_TIME

Base address: 0x4000_3600
Address offset: 0x20
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**DEEPSLEEP_TIME[31:0]**: Deep sleep time is N * $T_{32k\ clk}$	R/W

6.5.9 AON_DEEPSLEEP_CTRL

Base address: 0x4000_3600
Address offset: 0x24
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:5>	Reserved	R/W
<4>	**DEEPSLEEP_EXT_WAKEUP**: external wakeup enable 1'b0: disable 1'b1: enable	R/W
<3:1>	Reserved	R/W
<0>	**DEEPSLEEP_MODE**: Deep sleep mode 1'b0: normal mode 1'b1: deep sleep mode	R/W

6.5.10 AON register map

The base address of the AON registers is 0x4000_3600.
The following table gives the AON register map and reset values.

Table 13 AON register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	AON_CFG	CFG_RTCRST	CFG_WDTI_EN	Res	CFG_WDTI_SPD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CFG_32KCLKSRC[1:0]	
	Reset value	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
0x04	RTC_REFCLK_DIV	RTC_REFCLK_DIV_DIV[31:0]																															
	Reset value	0	0	0	0	0	0	0	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
0x08	AON_CLK_CTRL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLKCTRL_32KRC_RESTUNE[5:0]						CLKCTRL_32KXTAL_CAPTUNE[3:0]							CLKCTRL_32KXTAL_ITUNE[3:0]					CLKCTRL_32K_CAL				CLKCTRL_32KXTAL_PD	CLKCTRL_32KRC_PD
	Reset value	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0
0x0C	LDO_CFG	Res	Res	LDO_CFG_EXTLDO_EN	LDO_CFG_LDOIO_BP				LDO_CFG_IO_OTUNE[3:0]	Res			LDO_CFG_ANALOG_OTUNE[2:0]	Res			LDO_CFG_CORE_OTUNE[2:0]	Res	Res	Res					LDO_CFG_VOLDET_TRG[4:0]				LDO_CFG_VOLDET_PD	Res	LDO_CFG_IO_PD	LDO_CFG_ANALOG_PD	LDO_CFG_CORE_PD	LDO_CFG_IVREF_PD
	Reset value	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	IVREF_CFG_1	IVREF_CFG_1_LED_REF[3:0]							IVREF_CFG_1_PVD_REF [3:0]	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0x14	IVREF_CFG_2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18	IVREF_CFG_3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IVREF_CFG_3_HSI_REF[3:0]				IVREF_CFG_3_TSU_REF[3:0]				Res	Res	Res	Res	IVREF_CFG_3_DAC_REF[3:0]			
	Reset value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0x20	DEEPSLEEP_TIME	DEEPSLEEP_TIME[31:0]																															
	Reset value	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
0x24	DEEPSLEEP_CTRL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DEEPSLEEP_CTRL_EXT_WAKEUP	Res	Res	Res	DEEPSLEEP_CTRL_MODE
	Reset value	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

7 General-purpose IOs (GPIO)

7.1 Introduction

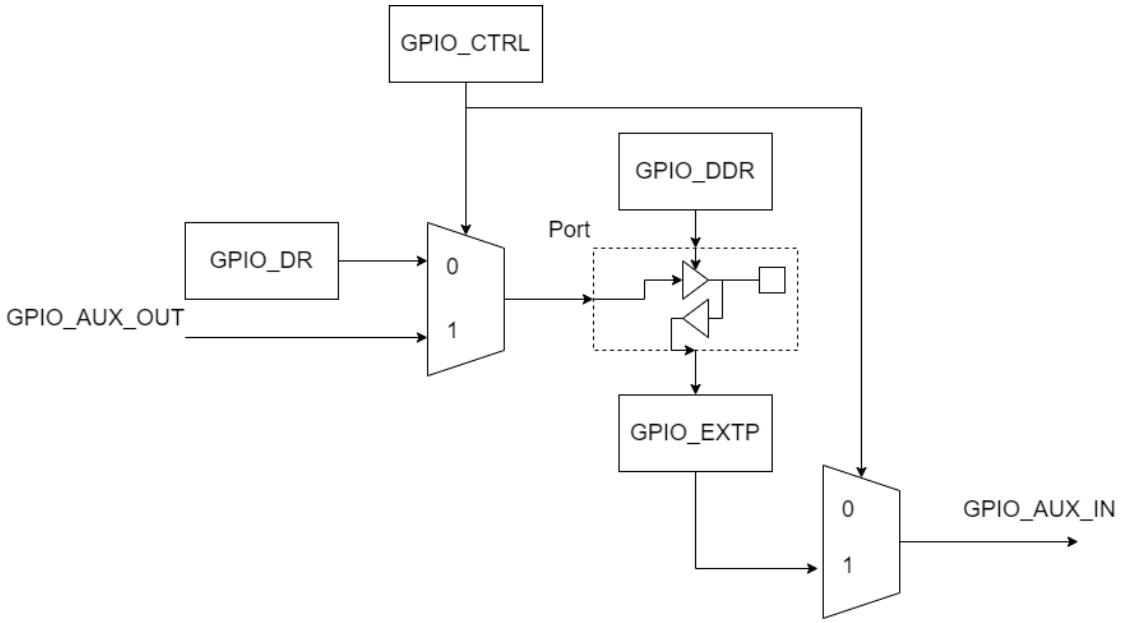


Figure 12 GPIO port control

7.2 GPIO signal descriptions

The GPIO signals are shown in the following table.

Table 14 GPIO signal descriptions

Signal	Description	I/O
Portx[31:0]	IO data state (External interrupt input only for Port A)	I/O

7.3 GPIO main features

XT32H0 supports four GPIO ports which are A, B, C and D. The width of GPIO ports is 32 bits. The data flow which are data and direction can be controlled by software and hardware.

GPIOA can accept external signals as interrupt sources. It supports interrupt types described in the following lines:

- Active-high and level

- Active-low and level
- Riding edge
- Falling edge

7.4 GPIO functional description

7.4.1 Software source outputs

Programming with the following settings:

- Set GPIOx_CTRL to software source;
- Set GPIOx_DDR to output mode;
- Write data to GPIOx_DR.

7.4.2 Software source inputs

Programming with the following settings:

- Set GPIOx_CTRL to software source;
- Set GPIOx_DDR to input mode;
- Read data from GPIOx_EXTP.

7.4.3 Hardware source input and outputs

Programming with the following setting:

- Set GPIOx_CTRL to hardware source;
- Read data from GPIOx_EXTP.

Note: the details of signal mapping refer to system config (GPIO AUX).

7.4.4 Interrupt sources

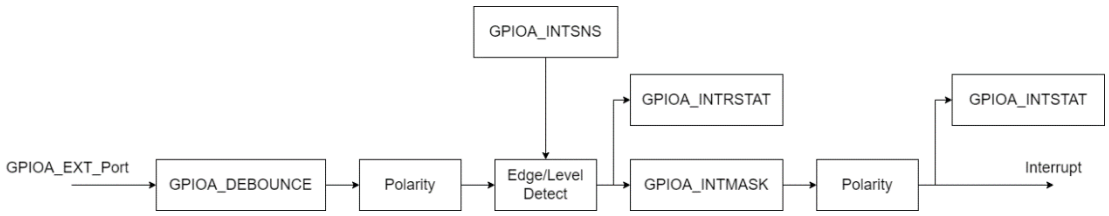


Figure 13 Interrupt process

Programming with the following setting:

- Set GPIOA_CTRL to software source;
- Set GPIOA_DDR to input mode;
- Configure interrupt type (GPIOA_INISNS), signal polarity (GPIOA_INTPOL), debounce (GPIOA_DEBOUNCE) and synchronization to system clock (GPIO_SYNC);
- Unmask interrupt (GPIOA_INTMASK);
- Enable interrupt (GPIOA_INTEN).

Check interrupt status by the following register:

- GPIOA_INTSTAT read the masked interrupt status for port A;
- GPIOA_INTRSTAT read the raw interrupt status for port A.

Clear interrupt to write 1 to GPIOA_EOI.

7.5 GPIO registers

7.5.1 GPIOx_DR

Base address: 0x5000_0C00

Address offset:

GPIOA_DR 0x00
GPIOB_DR 0x0C
GPIOC_DR 0x18
GPIOD_DR 0x24

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:0>	**DRx_B_y**: Port x (x is A, B,C and D) line y (y==31 to 0) data	R/W

7.5.2 GPIOx_DDR

Base address: 0x5000_0C00

Address offset:

GPIOA_DDR 0x04
GPIOB_DDR 0x10
GPIOC_DDR 0x1C
GPIOD_DDR 0x28

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:0>	**DDRx_B_y**: Port x (x is A, B,C and D) line y (y==31 to 0) data direction 1'b0: Input (default) 1'b1: Output	R/W

7.5.3 GPIOx_CTRL

Base address: 0x5000_0C00
Address offset:

GPIOA_CTRL 0x08
GPIOB_CTRL 0x14
GPIOC_CTRL 0x20
GPIOD_CTRL 0x2C

Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CTLx_B_y**: Port x (x is A, B,C and D) line y (y==31 to 0) data source 1'b0: Software mode (default) 1'b1: Hardware mode	R/W

7.5.4 GPIOA_INTEN

Base address: 0x5000_0C00
Address offset: 0x30
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**INTEN_B_y**: Port A line y (y=31 to 0) interrupt enable 1'b0: Normal GPIO signal (default) 1'b1: Interrupt	R/W

7.5.5 GPIOA_INTMASK

Base address: 0x5000_0C00
Address offset: 0x34
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**INTMSK_B_y**: Port A line y (y=31 to 0) interrupt mask 1'b0: Unmask interrupt (default) 1'b1: Mask interrupt	R/W

7.5.6 GPIOA_INTSNS

Base address: 0x5000_0C00
Address offset: 0x38
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**INTLVL_B_y**: Port A line y (y=31 to 0) interrupt sensitive 1'b0: Level sensitive (default) 1'b1: Edge sensitive	R/W

7.5.7 GPIOA_INTPOL

Base address: 0x5000_0C00
Address offset: 0x3C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**INTPOL_B_y**: Port A line y (y=31 to 0) interrupt polarity 1'b0: Active-low (default) 1'b1: Active-high	R/W

7.5.8 GPIOA_INTSTAT

Base address: 0x5000_0C00
Address offset: 0x40
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**INTSTA_B_y**: Port A line y (y=31 to 0) interrupt status	R/W

7.5.9 GPIOA_INTRSTAT

Base address: 0x5000_0C00
Address offset: 0x44
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**INTRAWSTA_B_y**: Port A line y (y=31 to 0) raw interrupt status	R/W

7.5.10 GPIOA_DEBOUNCE

Base address: 0x5000_0C00
Address offset: 0x48
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**DEBOU_B_y**: Port A line y (y=31 to 0) interrupt debounce enable 1'b0: No debounce (default) 1'b1: Enable debounce	R/W

7.5.11 GPIOA_EOI

Base address: 0x5000_0C00
Address offset: 0x4C
Type: Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**EOIA_B_y**: Port A line y (y=31 to 0) clear interrupt 1'b0: No interrupt clear (default) 1'b1: Clear interrupt	R/W

7.5.12 GPIOx_EXTP

Base address: 0x5000_0C00
Address offset:

GPIOA_EXTP 0x50
GPIOB_EXTP 0x54
GPIOC_EXTP 0x58
GPIOD_EXTP 0x5C

Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**EXTx_B_y**: External port x (x is A, B, C and D) line y (y=31 to 0) data	R/W

7.5.13 GPIO_SYNC

Base address: 0x5000_0C00
Address offset: 0x60
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**LSSYNC_SYNC**: Port A interrupt synchronization 1'b0: No synchronization to PCLK (default) 1'b1: Synchronization to PCLK	R/W

7.5.14 GPIO register map

The base address of the GPIO registers is 0x5000_0C00.
The following table gives the GPIO register map and reset values.

Table 15 GPIO register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIO_DRA	DRA_B_31	DRA_B_30	DRA_B_29	DRA_B_28	DRA_B_27	DRA_B_26	DRA_B_25	DRA_B_24	DRA_B_23	DRA_B_22	DRA_B_21	DRA_B_20	DRA_B_19	DRA_B_18	DRA_B_17	DRA_B_16	DRA_B_15	DRA_B_14	DRA_B_13	DRA_B_12	DRA_B_11	DRA_B_10	DRA_B_09	DRA_B_08	DRA_B_07	DRA_B_06	DRA_B_05	DRA_B_04	DRA_B_03	DRA_B_02	DRA_B_01	DRA_B_00
	Reset value	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
0x04	GPIO_DDRA	DDRA_B_31	DDRA_B_30	DDRA_B_29	DDRA_B_28	DDRA_B_27	DDRA_B_26	DDRA_B_25	DDRA_B_24	DDRA_B_23	DDRA_B_22	DDRA_B_21	DDRA_B_20	DDRA_B_19	DDRA_B_18	DDRA_B_17	DDRA_B_16	DDRA_B_15	DDRA_B_14	DDRA_B_13	DDRA_B_12	DDRA_B_11	DDRA_B_10	DDRA_B_09	DDRA_B_08	DDRA_B_07	DDRA_B_06	DDRA_B_05	DDRA_B_04	DDRA_B_03	DDRA_B_02	DDRA_B_01	DDRA_B_00
	Reset value	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
0x08	GPIO_CTLA	CTLA_B_31	CTLA_B_30	CTLA_B_29	CTLA_B_28	CTLA_B_27	CTLA_B_26	CTLA_B_25	CTLA_B_24	CTLA_B_23	CTLA_B_22	CTLA_B_21	CTLA_B_20	CTLA_B_19	CTLA_B_18	CTLA_B_17	CTLA_B_16	CTLA_B_15	CTLA_B_14	CTLA_B_13	CTLA_B_12	CTLA_B_11	CTLA_B_10	CTLA_B_09	CTLA_B_08	CTLA_B_07	CTLA_B_06	CTLA_B_05	CTLA_B_04	CTLA_B_03	CTLA_B_02	CTLA_B_01	CTLA_B_00
	Reset value	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
0x0C	GPIO_DRB	DRB_B_31	DRB_B_30	DRB_B_29	DRB_B_28	DRB_B_27	DRB_B_26	DRB_B_25	DRB_B_24	DRB_B_23	DRB_B_22	DRB_B_21	DRB_B_20	DRB_B_19	DRB_B_18	DRB_B_17	DRB_B_16	DRB_B_15	DRB_B_14	DRB_B_13	DRB_B_12	DRB_B_11	DRB_B_10	DRB_B_09	DRB_B_08	DRB_B_07	DRB_B_06	DRB_B_05	DRB_B_04	DRB_B_03	DRB_B_02	DRB_B_01	DRB_B_00
	Reset value	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
0x10	GPIO_DDRB	DDRB_B_31	DDRB_B_30	DDRB_B_29	DDRB_B_28	DDRB_B_27	DDRB_B_26	DDRB_B_25	DDRB_B_24	DDRB_B_23	DDRB_B_22	DDRB_B_21	DDRB_B_20	DDRB_B_19	DDRB_B_18	DDRB_B_17	DDRB_B_16	DDRB_B_15	DDRB_B_14	DDRB_B_13	DDRB_B_12	DDRB_B_11	DDRB_B_10	DDRB_B_09	DDRB_B_08	DDRB_B_07	DDRB_B_06	DDRB_B_05	DDRB_B_04	DDRB_B_03	DDRB_B_02	DDRB_B_01	DDRB_B_00
	Reset value	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
0x14	GPIO_CTLB	CTLB_B_31	CTLB_B_30	CTLB_B_29	CTLB_B_28	CTLB_B_27	CTLB_B_26	CTLB_B_25	CTLB_B_24	CTLB_B_23	CTLB_B_22	CTLB_B_21	CTLB_B_20	CTLB_B_19	CTLB_B_18	CTLB_B_17	CTLB_B_16	CTLB_B_15	CTLB_B_14	CTLB_B_13	CTLB_B_12	CTLB_B_11	CTLB_B_10	CTLB_B_09	CTLB_B_08	CTLB_B_07	CTLB_B_06	CTLB_B_05	CTLB_B_04	CTLB_B_03	CTLB_B_02	CTLB_B_01	CTLB_B_00
	Reset value	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
0x18	GPIO_DRC	DRC_B_31	DRC_B_30	DRC_B_29	DRC_B_28	DRC_B_27	DRC_B_26	DRC_B_25	DRC_B_24	DRC_B_23	DRC_B_22	DRC_B_21	DRC_B_20	DRC_B_19	DRC_B_18	DRC_B_17	DRC_B_16	DRC_B_15	DRC_B_14	DRC_B_13	DRC_B_12	DRC_B_11	DRC_B_10	DRC_B_09	DRC_B_08	DRC_B_07	DRC_B_06	DRC_B_05	DRC_B_04	DRC_B_03	DRC_B_02	DRC_B_01	DRC_B_00
	Reset value	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
0x1C	GPIO_DDRC	DDRC_B_31	DDRC_B_30	DDRC_B_29	DDRC_B_28	DDRC_B_27	DDRC_B_26	DDRC_B_25	DDRC_B_24	DDRC_B_23	DDRC_B_22	DDRC_B_21	DDRC_B_20	DDRC_B_19	DDRC_B_18	DDRC_B_17	DDRC_B_16	DDRC_B_15	DDRC_B_14	DDRC_B_13	DDRC_B_12	DDRC_B_11	DDRC_B_10	DDRC_B_09	DDRC_B_08	DDRC_B_07	DDRC_B_06	DDRC_B_05	DDRC_B_04	DDRC_B_03	DDRC_B_02	DDRC_B_01	DDRC_B_00
	Reset value	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
0x20	GPIO_CTLC	CTLC_B_31	CTLC_B_30	CTLC_B_29	CTLC_B_28	CTLC_B_27	CTLC_B_26	CTLC_B_25	CTLC_B_24	CTLC_B_23	CTLC_B_22	CTLC_B_21	CTLC_B_20	CTLC_B_19	CTLC_B_18	CTLC_B_17	CTLC_B_16	CTLC_B_15	CTLC_B_14	CTLC_B_13	CTLC_B_12	CTLC_B_11	CTLC_B_10	CTLC_B_09	CTLC_B_08	CTLC_B_07	CTLC_B_06	CTLC_B_05	CTLC_B_04	CTLC_B_03	CTLC_B_02	CTLC_B_01	CTLC_B_00
	Reset value	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
0x24	GPIO_DRD	DRD_B_31	DRD_B_30	DRD_B_29	DRD_B_28	DRD_B_27	DRD_B_26	DRD_B_25	DRD_B_24	DRD_B_23	DRD_B_22	DRD_B_21	DRD_B_20	DRD_B_19	DRD_B_18	DRD_B_17	DRD_B_16	DRD_B_15	DRD_B_14	DRD_B_13	DRD_B_12	DRD_B_11	DRD_B_10	DRD_B_09	DRD_B_08	DRD_B_07	DRD_B_06	DRD_B_05	DRD_B_04	DRD_B_03	DRD_B_02	DRD_B_01	DRD_B_00

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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
0x28	GPIO_DDRD	DDRD_B_31	DDRD_B_30	DDRD_B_29	DDRD_B_28	DDRD_B_27	DDRD_B_26	DDRD_B_25	DDRD_B_24	DDRD_B_23	DDRD_B_22	DDRD_B_21	DDRD_B_20	DDRD_B_19	DDRD_B_18	DDRD_B_17	DDRD_B_16	DDRD_B_15	DDRD_B_14	DDRD_B_13	DDRD_B_12	DDRD_B_11	DDRD_B_10	DDRD_B_09	DDRD_B_08	DDRD_B_07	DDRD_B_06	DDRD_B_05	DDRD_B_04	DDRD_B_03	DDRD_B_02	DDRD_B_01	DDRD_B_00
	Reset value	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
0x2c	GPIO_CTLD	CTLD_B_31	CTLD_B_30	CTLD_B_29	CTLD_B_28	CTLD_B_27	CTLD_B_26	CTLD_B_25	CTLD_B_24	CTLD_B_23	CTLD_B_22	CTLD_B_21	CTLD_B_20	CTLD_B_19	CTLD_B_18	CTLD_B_17	CTLD_B_16	CTLD_B_15	CTLD_B_14	CTLD_B_13	CTLD_B_12	CTLD_B_11	CTLD_B_10	CTLD_B_09	CTLD_B_08	CTLD_B_07	CTLD_B_06	CTLD_B_05	CTLD_B_04	CTLD_B_03	CTLD_B_02	CTLD_B_01	CTLD_B_00
	Reset value	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
0x30	GPIO_INTEN	INTEN_B_31	INTEN_B_30	INTEN_B_29	INTEN_B_28	INTEN_B_27	INTEN_B_26	INTEN_B_25	INTEN_B_24	INTEN_B_23	INTEN_B_22	INTEN_B_21	INTEN_B_20	INTEN_B_19	INTEN_B_18	INTEN_B_17	INTEN_B_16	INTEN_B_15	INTEN_B_14	INTEN_B_13	INTEN_B_12	INTEN_B_11	INTEN_B_10	INTEN_B_09	INTEN_B_08	INTEN_B_07	INTEN_B_06	INTEN_B_05	INTEN_B_04	INTEN_B_03	INTEN_B_02	INTEN_B_01	INTEN_B_00
	Reset value	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
0x34	GPIO_INTMSK	INTMSK_B_31	INTMSK_B_30	INTMSK_B_29	INTMSK_B_28	INTMSK_B_27	INTMSK_B_26	INTMSK_B_25	INTMSK_B_24	INTMSK_B_23	INTMSK_B_22	INTMSK_B_21	INTMSK_B_20	INTMSK_B_19	INTMSK_B_18	INTMSK_B_17	INTMSK_B_16	INTMSK_B_15	INTMSK_B_14	INTMSK_B_13	INTMSK_B_12	INTMSK_B_11	INTMSK_B_10	INTMSK_B_09	INTMSK_B_08	INTMSK_B_07	INTMSK_B_06	INTMSK_B_05	INTMSK_B_04	INTMSK_B_03	INTMSK_B_02	INTMSK_B_01	INTMSK_B_00
	Reset value	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
0x38	GPIO_INTLVL	INTLVL_B_31	INTLVL_B_30	INTLVL_B_29	INTLVL_B_28	INTLVL_B_27	INTLVL_B_26	INTLVL_B_25	INTLVL_B_24	INTLVL_B_23	INTLVL_B_22	INTLVL_B_21	INTLVL_B_20	INTLVL_B_19	INTLVL_B_18	INTLVL_B_17	INTLVL_B_16	INTLVL_B_15	INTLVL_B_14	INTLVL_B_13	INTLVL_B_12	INTLVL_B_11	INTLVL_B_10	INTLVL_B_09	INTLVL_B_08	INTLVL_B_07	INTLVL_B_06	INTLVL_B_05	INTLVL_B_04	INTLVL_B_03	INTLVL_B_02	INTLVL_B_01	INTLVL_B_00
	Reset value	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
0x3C	GPIO_INTPOL	INTPOL_B_31	INTPOL_B_30	INTPOL_B_29	INTPOL_B_28	INTPOL_B_27	INTPOL_B_26	INTPOL_B_25	INTPOL_B_24	INTPOL_B_23	INTPOL_B_22	INTPOL_B_21	INTPOL_B_20	INTPOL_B_19	INTPOL_B_18	INTPOL_B_17	INTPOL_B_16	INTPOL_B_15	INTPOL_B_14	INTPOL_B_13	INTPOL_B_12	INTPOL_B_11	INTPOL_B_10	INTPOL_B_09	INTPOL_B_08	INTPOL_B_07	INTPOL_B_06	INTPOL_B_05	INTPOL_B_04	INTPOL_B_03	INTPOL_B_02	INTPOL_B_01	INTPOL_B_00
	Reset value	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
0x40	GPIO_INTSTA	INTSTA_B_31	INTSTA_B_30	INTSTA_B_29	INTSTA_B_28	INTSTA_B_27	INTSTA_B_26	INTSTA_B_25	INTSTA_B_24	INTSTA_B_23	INTSTA_B_22	INTSTA_B_21	INTSTA_B_20	INTSTA_B_19	INTSTA_B_18	INTSTA_B_17	INTSTA_B_16	INTSTA_B_15	INTSTA_B_14	INTSTA_B_13	INTSTA_B_12	INTSTA_B_11	INTSTA_B_10	INTSTA_B_09	INTSTA_B_08	INTSTA_B_07	INTSTA_B_06	INTSTA_B_05	INTSTA_B_04	INTSTA_B_03	INTSTA_B_02	INTSTA_B_01	INTSTA_B_00
	Reset value	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
0x44	GPIO_INTRAWSTA	INTRAWSTA_B_31	INTRAWSTA_B_30	INTRAWSTA_B_29	INTRAWSTA_B_28	INTRAWSTA_B_27	INTRAWSTA_B_26	INTRAWSTA_B_25	INTRAWSTA_B_24	INTRAWSTA_B_23	INTRAWSTA_B_22	INTRAWSTA_B_21	INTRAWSTA_B_20	INTRAWSTA_B_19	INTRAWSTA_B_18	INTRAWSTA_B_17	INTRAWSTA_B_16	INTRAWSTA_B_15	INTRAWSTA_B_14	INTRAWSTA_B_13	INTRAWSTA_B_12	INTRAWSTA_B_11	INTRAWSTA_B_10	INTRAWSTA_B_09	INTRAWSTA_B_08	INTRAWSTA_B_07	INTRAWSTA_B_06	INTRAWSTA_B_05	INTRAWSTA_B_04	INTRAWSTA_B_03	INTRAWSTA_B_02	INTRAWSTA_B_01	INTRAWSTA_B_00
	Reset value	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
0x48	GPIO_DEBOU	DEBOU_B_31	DEBOU_B_30	DEBOU_B_29	DEBOU_B_28	DEBOU_B_27	DEBOU_B_26	DEBOU_B_25	DEBOU_B_24	DEBOU_B_23	DEBOU_B_22	DEBOU_B_21	DEBOU_B_20	DEBOU_B_19	DEBOU_B_18	DEBOU_B_17	DEBOU_B_16	DEBOU_B_15	DEBOU_B_14	DEBOU_B_13	DEBOU_B_12	DEBOU_B_11	DEBOU_B_10	DEBOU_B_09	DEBOU_B_08	DEBOU_B_07	DEBOU_B_06	DEBOU_B_05	DEBOU_B_04	DEBOU_B_03	DEBOU_B_02	DEBOU_B_01	DEBOU_B_00
	Reset value	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
0x4C	GPIO_EOIA	EOIA_B_31	EOIA_B_30	EOIA_B_29	EOIA_B_28	EOIA_B_27	EOIA_B_26	EOIA_B_25	EOIA_B_24	EOIA_B_23	EOIA_B_22	EOIA_B_21	EOIA_B_20	EOIA_B_19	EOIA_B_18	EOIA_B_17	EOIA_B_16	EOIA_B_15	EOIA_B_14	EOIA_B_13	EOIA_B_12	EOIA_B_11	EOIA_B_10	EOIA_B_09	EOIA_B_08	EOIA_B_07	EOIA_B_06	EOIA_B_05	EOIA_B_04	EOIA_B_03	EOIA_B_02	EOIA_B_01	EOIA_B_00
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	GPIO_EXTA	EXTA_B_31	EXTA_B_30	EXTA_B_29	EXTA_B_28	EXTA_B_27	EXTA_B_26	EXTA_B_25	EXTA_B_24	EXTA_B_23	EXTA_B_22	EXTA_B_21	EXTA_B_20	EXTA_B_19	EXTA_B_18	EXTA_B_17	EXTA_B_16	EXTA_B_15	EXTA_B_14	EXTA_B_13	EXTA_B_12	EXTA_B_11	EXTA_B_10	EXTA_B_09	EXTA_B_08	EXTA_B_07	EXTA_B_06	EXTA_B_05	EXTA_B_04	EXTA_B_03	EXTA_B_02	EXTA_B_01	EXTA_B_00
	Reset value	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
0x54	GPIO_EXTB	EXTB_B_31	EXTB_B_30	EXTB_B_29	EXTB_B_28	EXTB_B_27	EXTB_B_26	EXTB_B_25	EXTB_B_24	EXTB_B_23	EXTB_B_22	EXTB_B_21	EXTB_B_20	EXTB_B_19	EXTB_B_18	EXTB_B_17	EXTB_B_16	EXTB_B_15	EXTB_B_14	EXTB_B_13	EXTB_B_12	EXTB_B_11	EXTB_B_10	EXTB_B_09	EXTB_B_08	EXTB_B_07	EXTB_B_06	EXTB_B_05	EXTB_B_04	EXTB_B_03	EXTB_B_02	EXTB_B_01	EXTB_B_00
	Reset value	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
0x58	GPIO_EXTC	EXTC_B_31	EXTC_B_30	EXTC_B_29	EXTC_B_28	EXTC_B_27	EXTC_B_26	EXTC_B_25	EXTC_B_24	EXTC_B_23	EXTC_B_22	EXTC_B_21	EXTC_B_20	EXTC_B_19	EXTC_B_18	EXTC_B_17	EXTC_B_16	EXTC_B_15	EXTC_B_14	EXTC_B_13	EXTC_B_12	EXTC_B_11	EXTC_B_10	EXTC_B_09	EXTC_B_08	EXTC_B_07	EXTC_B_06	EXTC_B_05	EXTC_B_04	EXTC_B_03	EXTC_B_02	EXTC_B_01	EXTC_B_00
	Reset value	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
0x5C	GPIO_EXTD	EXTD_B_31	EXTD_B_30	EXTD_B_29	EXTD_B_28	EXTD_B_27	EXTD_B_26	EXTD_B_25	EXTD_B_24	EXTD_B_23	EXTD_B_22	EXTD_B_21	EXTD_B_20	EXTD_B_19	EXTD_B_18	EXTD_B_17	EXTD_B_16	EXTD_B_15	EXTD_B_14	EXTD_B_13	EXTD_B_12	EXTD_B_11	EXTD_B_10	EXTD_B_09	EXTD_B_08	EXTD_B_07	EXTD_B_06	EXTD_B_05	EXTD_B_04	EXTD_B_03	EXTD_B_02	EXTD_B_01	EXTD_B_00
	Reset value	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
0x60	GPIO_LSSYNC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LSSYNC_SYNC
	Reset value	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

8 System configuration controller (SYSCFG)

8.1 Introduction

SYSCFG defined system configuration and status including UID, SRAM configure, boot mode, UART mode, SPI mode, external IRQ definition, reset, DMA handshaking configure, advanced timer break source, GPIO ENAUX and Flash BIST status.

8.2 SYSCFG main features

- Configure peripheral operation mode
- Configure boot mode
- Configure external interrupts line multiplexing to the Interrupt Controller (ICTL)
- Configure DMA handshaking interface
- Reset peripheral
- Configure GPIO auxiliary ports
- Configure advanced timer break trigger

8.3 SYSCFG functional description

8.3.1 Interrupt assignment

Interrupt source from GPIO and peripherals to the ICTL can be configured using the SYS_IRQ_CFG_0 and SYS_IRQ_CFG_1 register. Please refer to Table 16. External IRQ assignment for more information on the interrupt source selection.

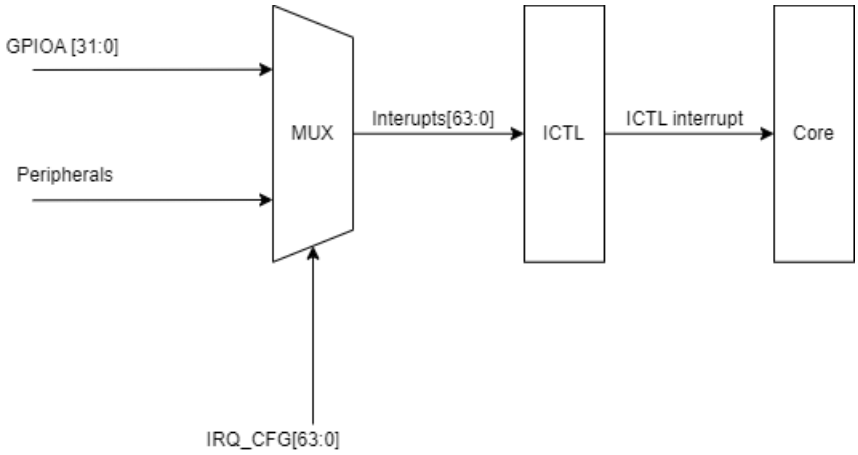


Figure 14 Diagram of the external interrupts line multiplexing to ICTL

8.3.2 DMA handshaking interface assignment

DMA handshaking signals from peripherals to the DMA controller can be assigned using the SYS_DMA_CFG and SYS_DMA_CFG_2 register. Please refer to Table 17. DMA handshaking channel assignment 1 and Table 18. DMA handshaking channel assignment 2 for more information on DMA channel assignments

8.3.3 GPIO auxiliary ports

XT32H05x has a total of 9 different GPIO main configurations (CFG_0 - CFG_8) with additional auxiliary configurations under GPIO CFG_2. The auxiliary GPIO configuration under GPIO CFG_2 can be assigned using the SYS_GPIO_ENAUX_A, SYS_GPIO_ENAUX_B, SYS_GPIO_ENAUX_C, SYS_GPIO_ENAUX_D register. Please refer to the IO pin assignment tables under the Port mux (PADI) section for more information.

When the GPIO is configured to CFG_2, the SYS_GPIO_ENAUX_X registers should be assigned according to the intended usage of the hardware resources. Please refer to section 1.4.17, 1.4.18, 1.4.19, 1.4.21 for more information.

If a GPIO is configured as an input under CFG2_AUX0 or CFG2_AUX1 to act as the external timer clocks or triggers, the ICG should also enable the corresponding external clock for the timers. Please refer to section 4.4.19 for more information.

8.3.4 Peripheral software reset

Peripherals in XT32H05x can be reset individually through the SYS_SOFT_RESET_CFG register. The following diagram shows the block diagram of the reset generation circuit.

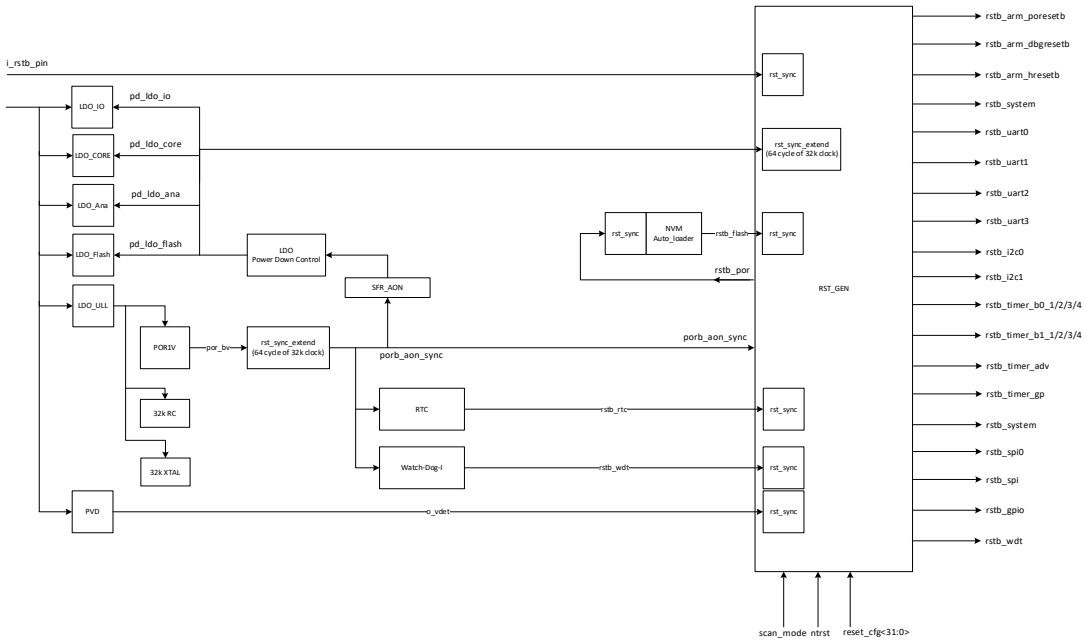


Figure 15 Reset generation block diagram

8.3.5 UART baudrate auto detection

XT32H05x supports up to 4 UART with automatic baud rate detection. The Auto-Baud Detection system detects the baud rate of the master controller and self-adjust accordingly.
To auto detect the UART baud rate, please follow the steps shown below:

1. Reset UARTx by clearing the relevant RST_CFG_UARTx bit to 0 in the SYS_SOFT_RESET_CFG register
2. Set the UARTx auto baud rate detect interrupt enable bit, UARTx_ADCFG_IE, if interrupt is used
3. Set the UARTx auto baud rate detect enable bit, UARTx_ADCFG_EN, to 1
4. Poll the corresponding UARTx auto band rate detect flag, UARTx_ADFLG, to see if it has been set to 1. Alternatively, the software should jump to the interrupt service routine when the UARTx auto baud rate detect interrupt is enabled and auto-baud detection is completed.
5. Read the baud rate detected in the SYS_UARTx_ADBAUD_REG register. The band rate value shows the equivalent number of cycles in the system clock.
6. Release the UARTx reset by setting the relevant RST_CFG_UARTx bit to 1.

8.3.6 Hardware divider

The device incorporates a 32-bit hardware divider to accelerate the division operation. To perform 32-bit hardware division, first write the dividend to SYS_HDIV_A follow by the divisor to SYS_HDIV_B. The quotient of the division results can be read from the SYS_HDIV_Q register and the remainder of the division results can be read from SYS_HDIV_R register.
Note: The dividend, SYS_HDIV_A, must be written before the divisor, SYS_HDIV_B.

8.3.7 RC oscillator Auto Frequency Calibration (AFC)

The device incorporates auto calibration logic to achieve an accurate RC oscillator frequency for the 32KHz RC oscillator and the 16MHz RC oscillator. The calibration process requires two clock sources. One act as the golden reference clock and the other one is the RC clock to be calibrated.

For a given number of clock cycles in the golden reference clock, the corresponding number of RC clock cycle expected can be computed theoretically. The calibration logic compares and adjusts the difference between the number of clock cycle expected theoretically vs. the number of cycles measured for the RC clock iteratively to determine the calibration coefficient value. The process terminates once the difference count value is smaller than the value defined in the AFCCTRL_DIFFCNT[7:0] in the SYS_AFC_CTRL register.
To begin the calibration process, first specify the total number of reference clock cycles to be included

in the calibration cycle by setting the AFCCTRL_RFCNT[13:0] and select the RC clock to be calibrated by setting the AFCCFG_CBCLKS bit in the SYS_AFC_CTRL register. Next, set the theoretical count value expected to AFCCFG_STDCNT[19:0] and the initial seed value to AFCCTRL_CBADJ[7:0] for the hardware calibration search algorithm in the SYS_AFC_CFG register. To aid in the calibration process, the reference clock used for calibration may be further divide down by specifying the divider ratio through AFCRES_RFDIV[15:0] in the SYS_AFC_RES register. Once these options are configured, the device is ready to perform the RC clock calibration. The calibration process starts when the AFCCTRL_AFCEN is set to 1 in the SYS_AFC_CTRL register. Note that additional interrupt and clock source configuration may be required to automate the programming of the calibration value obtained to the respective RC clock configuration registers.

Reference clock count means calibration time of one stage. The maximum number of stages is the same as the number of result bits. Standard count is the arm of calibration. Different count shows the resolution of calibration. n is the number of result bits.

$$Time_{calib} = \frac{Count_{refclk}}{Freq_{refclk}}$$

$$Count_{stand} = Time_{calib} \times Freq_{stand} = Count_{refclk} \times \frac{Freq_{stand}}{Freq_{refclk}}$$

$$Count_{diff} = \frac{Count_{stand}}{2^{n-1}}$$

For two level AFC operation, two level use the same calibration time and the arm, n1 is the number of first level result bits and n2 is the number of second level result bits.

$$Time1_{calib} = Time2_{calib} = \frac{Count_{refclk}}{Freq_{refclk}}$$

$$Count1_{stand} = Count2_{stand} = Time_{calib} \times Freq_{stand} = Count_{refclk} \times \frac{Freq_{stand}}{Freq_{refclk}}$$

$$Count1_{diff} = \frac{Count_{stand}}{2^{n1}}$$

$$Count2_{diff} = \frac{Count_{stand}}{2^{n1+n2}}$$

For example,

$$Freq_{refclk} = 32768\ Hz$$
$$Freq_{stand} = 16000000\ Hz$$
$$n1 = 7$$
$$n2 = 3$$
$$Time1_{calib} = Time2_{calib} = 500\ ms$$
$$Count_{refclk} = 16339$$
$$Count1_{stand} = Count2_{stabd} = 8000000$$
$$Count1_{diff} = 62500 - 500 = 62000$$
$$Count2_{diff} = 7812.5 - 312.5 = 7500$$

The following shows a sample AFC calibration flow:

1. Configure calibration
    a. Programming SYS_AFC_CTRL
        i. Write calibration time to SYS_AFC_CTRL.AFCCTRL_RFCNT[29:16];
        ii. Select clock source by programming SYS_AFC_CTRL.AFCCTRL_CBCLKS[5], b'0 means 32KHz RC oscillator as clock source, b'1 means 16M RC oscillator as clock source;
        iii. Write counter difference to SYS_AFC_CTRL.AFCCTRL_DIFFCNT[15:8];
    b. Programming SYS_AFC_CFG
        i. Write initial value to SYS_AFC_CFG.AFCCFG_CBADJ[7:0];
        ii. Write counter stand value to SYS_AFC_CFG.AFCCFG_STDCNT[27:8];
    c. Set reference clock divider by programming SYS_AFC_RES.AFCRES_RFDIV[31:16];
    d. Used AFC by writing 0 to SYS_AFC_CTRL.AFCCTRL_AFCBYS[1];
    e. Enable interrupts by writing 1 to SYS_AFC_CTRL.AFCCTRL_AEIE[4], SYS_AFC_CTRL.AFCCTRL_ANIE[3] and SYS_AFC_CTRL.AFCCTRL_ALIE[2];
    f. Switch clock run in calibration mode
        i. For 16MHZ RC clock source, write 1 to ICG_RC16M_CTRL.ICG_RC16M_CTRL_CALI[31];
        ii. For 32KHz RC clock source, write 1 to AON_CLK_CTRL. AON_CLKCTRL_32K_CAL [4].
2. Start AFC, enable AFC by writing 1 to SYS_AFC_CTRL. AFCCTRL_AFCEN[0]
3. Read result and write to clock configuration.
    a. Read calibration result from SYS_AFC_RES.AFCRES_ADJRES[]
    b. Write the result to clock configuration
        i. For 16MHZ RC clock source, write the result to ICG_RC16M_CTRL.ICG_RC16M_CTRL_CRES[7:0] ;
        ii. for 32KHz RC clock source, write the result to AON_CLK_CTRL. AON_CLKCTRL_32KRC_RESTUNE[21:16].
    c. Disable AFC by writing 0 to SYS_AFC_CTRL. AFCCTRL_AFCEN[0]
    d. Switch clock to normal operation mode
        i. For 16MHZ RC clock source, write 0 to ICG_RC16M_CTRL.ICG_RC16M_CTRL_CALI[31];
        ii. For 32KHz RC clock source, write 0 to AON_CLK_CTRL. AON_CLKCTRL_32K_CAL [4].

8.3.8 Inter-communication between main core and accelerator

XT32H05x incorporates hardware accelerator to assists the main system control and computation for improved system performance. The Cortex-M0+ core communicates with the hardware accelerator through the mailbox scheme. Handshake between the Cortex-M0+ core and the hardware accelerator are facilitated through interrupt generation and interrupt clearance.

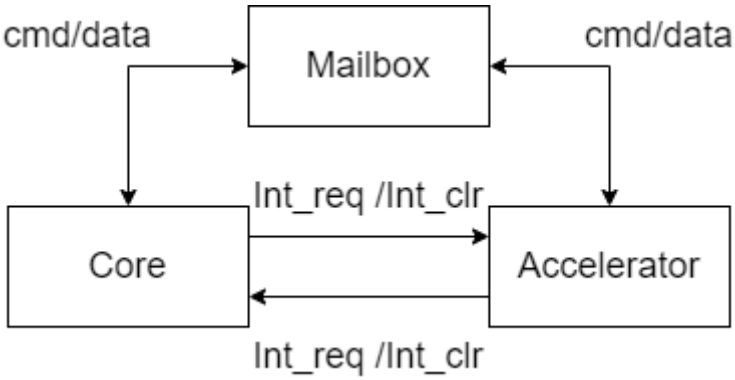


Figure 16 Handshaking between core and accelerator

8.3.8.1 Handshaking scheme

- To trigger an interrupt for the accelerated core, write 1 to SYS_CTSM_INT_HOST2CTSM.BIT[31].
- To check for interrupts from the accelerated core, read SYS_CTSM_INT_CTSM2HOST;
- To clear an interrupt from the accelerated core, write 0 to SYS_CTSM_CTRL.BIT[4].

8.3.9 Timers port configuration

8.3.9.1 Input

The SYS_TIM_A1_IMX register can be used to configure the input port of advanced timer 1 to the other advanced timers and general timers.
SYS_TIM_A2_IMX, SYS_TIM_G1_IMX, SYS_TIM_G2_IMX, SYS_TIM_G3_IMX, and SYS_TIM_G4_IMX registers are similar to SYS_TIM_A1_IMX.

8.3.9.2 Output

The channels of advanced timers and general timers can output either complementary signals or single-ended signal.
The SYS_TIM_CMS_OC register can be used to configure the output port of advanced timers and general timers as either complementary or single-ended.

8.4 SYSCFG registers

8.4.1 SYS_ROM_CFG

Base address: 0x4000_0800
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0010

Bit	Description	Type
<31:8>	Reserved	R/W
<7:4>	**ROM_CFG_RME1[3:0]**: Read margin input. These bits are used for setting the read margin. It programs the sense amp differential settings and adjusts the trade-off between speed and robustness. Users are advised to use the default value.	R/W
<3:2>	Reserved	R/W0
<1>	**ROM_CFG_RME**: Read margin enable 1'b0: Use the default ROM_CFG_RME1 setting. Settings defined in ROM_CFG_RME1 not effective 1'b1: Apply the ROM_CFG_RME1 setting to the ROM	R/W
<0>	**ROM_CFG_LS**: Light Sleep Enable The ROM goes into low power mode when this bit is set to 1. Please note that the ROM becomes inaccessible when it is in light sleep mode. 1'b0: Disable ROM light sleep mode 1'b1: Enable ROM light sleep mode	R/W

8.4.2 SYS_PVD_INT_CFG

Base address: 0x4000_0800
Address offset: 0x04
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:28>	Reserved	R/W
<27>	**PVD_INT_CFG_BP_FLT**: bypass PVD glitch filter 1'b0: Enable the glitch filter 1'b1: Bypass the glitch filter	R/W
<26>	**PVD_INT_CFG_INT_FLAG**: PVD interrupt flag. Write 0 to clear the flag The PVD interrupt flag will be set only when PVD_INT_CFG_INT_EN=1'b1. Once the interrupt controller response to the PVD interrupt request, the first step is to disable the PVD interrupt (PVD_INT_CFG_INT_EN =1'b0) and clear this flag. 1'b0: No PVD interrupt	R/W0

	1'b1: PVD interrupt flagged	
<25>	**PVD_INT_CFG_INT_POL**: PVD interrupt polarity 1'b0: High level PVD interrupt 1'b1: Low level PVD interrupt	R/W
<24>	**PVD_INT_CFG_INT_EN**: PVD interrupt enable 1'b0: PVD interrupt disable 1'b1: PVD interrupt enable	R/W
<23:0>	Reserved	R/W

8.4.3 SYS_SRAM_CTRL_1

Base address: 0x4000_0800
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_5022

Bit	Description	Type
<31:19>	Reserved	R/W
<18:16>	**SRAM_CFG_1_WPULSE[2:0]**: Write Assist pulse to control the pulse width of the negative voltage on the 16kB system SRAM bitline. Users are advised to keep the default value.	R/W
<15>	Reserved	
<14:12>	**SRAM_CFG_1_WA[2:0]**: Write assist bits to control the negative voltage on the SRAM bitline. Users are advised to keep the default value.	R/W0
<11:10>	Reserved	
<9:8>	**SRAM_CFG_1_RA[1:0]**: Read assist bits to control the word line under-drive. Users are advised to keep the default value.	R/W
<7:4>	**SRAM_CFG_1_RME1[3:0]**: These bits are used for setting the read margin. It programs the sense amp differential settings and adjusts the trade-off between speed and robustness. Users are advised to use the default value.	R/W
<3:2>	Reserved	
<1>	**SRAM_CFG_1_RME**: Read-write margin enable bit. This selects between the default Read-Write margin setting and the SRAM_CFG_1_RME1[3:0] setting. 1'b0: Use the default SRAM_CFG_1_RME1 setting. Settings defined in SRAM_CFG_1_RME1 not effective 1'b1: Apply the SRAM_CFG_1_RME1 setting to the SRAM	R/W
<0>	**SRAM_CFG_1_TEST**: Test bit to bypass self-timed circuit. Users are advised to keep the default value.	

8.4.4 SYS_SRAM_CTRL_PD

Base address: 0x4000_0800
Address offset: 0x1C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:27>	Reserved	R/W
<26>	**SRAM_CFG_PD_SHUTDOWN_4**: Shutdown control for the 128x32 LED display buffer 1'b0: Power on 1'b1: Power down	R/W
<25>	**SRAM_CFG_PD_DSLEEP_4**: Deep sleep enable for the 128x32 LED display buffer 1'b0: Normal mode 1'b1: Deep sleep mode	R/W
<24>	**SRAM_CFG_PD_LSLEEP_4**: Light sleep enable for the 128x32 LED display buffer 1'b0: Normal mode 1'b1: Light sleep mode	R/W
<23:19>	Reserved	R/W
<18>	**SRAM_CFG_PD_SHUTDOWN_3**: Shutdown control for the CTSM 4kB mailbox memory 1'b0: Power on 1'b1: Power down	R/W
<17>	**SRAM_CFG_PD_DSLEEP_3**: Deep sleep enable for the CTSM 4kB mailbox memory 1'b0: Normal mode 1'b1: Deep sleep mode	R/W
<16>	**SRAM_CFG_PD_LSLEEP_3**: Light sleep enable for the CTSM 4kB mailbox memory 1'b0: Normal mode 1'b1: Light sleep mode	R/W
<15:11>	Reserved	R/W
<10>	**SRAM_CFG_PD_SHUTDOWN_2**: Shutdown control for the CTSM 16kB SRAM 1'b0: Power on 1'b1: Power down	R/W
<9>	**SRAM_CFG_PD_DSLEEP_2**: Deep sleep enable for the CTSM 16kB SRAM 1'b0: Normal mode 1'b1: Deep sleep mode	R/W

<8>	**SRAM_CFG_PD_LSLEEP_2**: Light sleep enable for the CTSM 16kB SRAM 1'b0: Normal mode 1'b1: Light sleep mode	R/W
<7:3>	Reserved	R/W
<2>	**SRAM_CFG_PD_SHUTDOWN_1**: Shutdown control for the 16kB main system SRAM 1'b0: Power on 1'b1: Power down	R/W
<1>	**SRAM_CFG_PD_DSLEEP_1**: Deep sleep enable for the 16kB main system SRAM 1'b0: Normal mode 1'b1: Deep sleep mode	R/W
<0>	**SRAM_CFG_PD_LSLEEP_1**: Light sleep enable for the 16kB main system SRAM 1'b0: Normal mode 1'b1: Light sleep mode	R/W

8.4.5 SYS_CFG

Base address: 0x4000_0800
Address offset: 0x24
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	**SYS_CFG_BOOTCTRL**: boot mode control selection 1'b0: Boot mode control from chip pin 1'b1: Boot mode control from SFR register	R/W
<30>	**SYS_CFG_FLASHDMA**: Enable FLASH DMA transfer 1'b0: Flash DMA transfer disabled 1'b1: Flash DMA transfer enabled	R/W
<29:28>	**SYS_CFG_BOOT_MODE[1:0]**: Soft boot mode config for FOTA application 2'b00: Boot from Flash 2'b01: Boot from ROM 2'b10: Boot from SRAM 2'b11: Boot from Flash	R/W
<27:26>	**SYS_CFG_SPI2_RXDA[1:0]**: SPI 2 RX ack select 2'b00: dma_ack[15] when dma_cfg[31:30] is 2'b00 2'b01: dma_ack[11] when dma_cfg[23:22] is 2'b01 2'b10: dma_ack[3] when dma_cfg[7:6] is 2'b10 2'b11: dma_ack[15] when dma_cfg[31:30] is 2'b11	R/W
<25:24>	**SYS_CFG_SPI2_TXDA[1:0]**: SPI 2 RX ack select	R/W

Bit	Description	Type
	2’b00: dma_ack[14] when dma_cfg[29:28] is 2’b00 2’b01: dma_ack[10] when dma_cfg[21:20] is 2’b01 2’b10: dma_ack[2] when dma_cfg[1:0] is 2’b10 2’b11: dma_ack[14] when dma_cfg[29:28] is 2’b11	
<23:22>	**SYS_CFG_I2C1_RXDA[1:0]**: I2C 1 RX ack select 2’b00: dma_ack[9] when dma_cfg[19:18] is 2’b00 2’b01: dma_ack[1] when dma_cfg[3:2] is 2’b01 2’b10: dma_ack[15] when dma_cfg[31:30] is 2’b10 2’b11: dma_ack[9] when dma_cfg[19:18] is 2’b11	R/W
<21:20>	**SYS_CFG_I2C1_TXDA[1:0]**: I2C 1 TX ack select 2’b00: dma_ack[8] when dma_cfg[17:16] is 2’b00 2’b01: dma_ack[0] when dma_cfg[1:0] is 2’b01 2’b10: dma_ack[14] when dma_cfg[29:28] is 2’b10 2’b11: dma_ack[8] when dma_cfg[17:16] is 2’b11	R/W
<19:18>	**SYS_CFG_SPI1_RXDA[1:0]**: SPI 1 RX ack select 2’b00: dma_ack[13] when dma_cfg[27:26] is 2’b00 2’b01: dma_ack[9] when dma_cfg[19:18] is 2’b01 2’b10: dma_ack[1] when dma_cfg[3:2] is 2’b10 2’b11: dma_ack[13] when dma_cfg[27:26] is 2’b11	R/W
<17:16>	**SYS_CFG_SPI1_TXDA[1:0]**: SPI 1 RX ack select 2’b00: dma_ack[12] when dma_cfg[25:24] is 2’b00 2’b01: dma_ack[8] when dma_cfg[17:16] is 2’b01 2’b10: dma_ack[0] when dma_cfg[1:0] is 2’b10 2’b11: dma_ack[12] when dma_cfg[25:24] is 2’b11	R/W
<15:12>	Reserved.	R/W
<11>	**SYS_CFG_UART4_MODE**: UART4 mode of operations 1’b0: UART mode 1’b1: IrDA mode	R/W
<10>	**SYS_CFG_UART3_MODE**: UART3 mode of operations 1’b0: UART mode 1’b1: IrDA mode	R/W
<9>	**SYS_CFG_UART2_MODE**: UART2 mode of operations 1’b0: UART mode 1’b1: IrDA mode	R/W
<8>	**SYS_CFG_UART1_MODE**: UART1 mode of operations 1’b0: UART mode 1’b1: IrDA mode	R/W
<7:5>	Reserved	R/W
<3>	**SYS_CFG_HDIV_EN**: Enable hardware divider 1’b0: Hardware divider disabled 1’b1: Hardware divider enabled	R/W

Bit	Description	Type
<2>	Reserved	R/W
<1>	**SYS_CFG_SPI2_MODE**: SPI2 operation mode 1'b0: Slave mode 1'b1: Master mode	R/W
<0>	**SYS_CFG_SPI1_MODE**: SPI1 operation mode 1'b0: Slave mode 1'b1: Master mode	R/W

8.4.6 SYS_IRQ_CFG_1

Base address: 0x4000_0800
Address offset: 0x28
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**IRQ_CFG_IRQ_x**: Interrupt Source selection control for Interrupt controller's interrupt entry x (x=0 to 31)	R/W

8.4.7 SYS_IRQ_CFG_2

Base address: 0x4000_0800
Address offset: 0x2C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**IRQ_CFG_IRQ_x**: Interrupt Source selection control for Interrupt controller's interrupt entry x (x=32 to 63)	R/W

IRQ assign is shown in the following table.

Table 16 External IRQ assignment

Domain	#	src_sel	irq_cfg<n> = 1'b0	irq_cfg<n> = 1'b1
IRQ_INTC	0	irq_cfg_0<0>	ic_gen_call_intr	aduart_1_intr
IRQ_INTC	1	irq_cfg_0<1>	ic_rx_full_intr	aduart_2_intr
IRQ_INTC	2	irq_cfg_0<2>	ic_rd_req_intr	aduart_3_intr
IRQ_INTC	3	irq_cfg_0<3>	ic_start_det_intr	aduart_4_intr
IRQ_INTC	4	irq_cfg_0<4>	ic_restart_det_intr	
IRQ_INTC	5	irq_cfg_0<5>	ic_stop_det_intr	
IRQ_INTC	6	irq_cfg_0<6>	ic_activity_intr	
IRQ_INTC	7	irq_cfg_0<7>	ic_tx_empty_intr	
IRQ_INTC	8	irq_cfg_0<8>	ic_tx_abrt_intr	
IRQ_INTC	9	irq_cfg_0<9>	ic_tx_over_intr	
IRQ_INTC	10	irq_cfg_0<10>	ic_rx_done_intr	
IRQ_INTC	11	irq_cfg_0<11>	ic_rx_under_intr	
IRQ_INTC	12	irq_cfg_0<12>	ic_rx_over_intr	
IRQ_INTC	13	irq_cfg_0<13>	ssi_mst_intr	
IRQ_INTC	14	irq_cfg_0<14>	ssi_rxu_intr	
IRQ_INTC	15	irq_cfg_0<15>	ssi_rxo_intr	
IRQ_INTC	16	irq_cfg_0<16>	ssi_rxf_intr	
IRQ_INTC	17	irq_cfg_0<17>	ssi_txo_intr	
IRQ_INTC	18	irq_cfg_0<18>	ssi_txe_intr	
IRQ_INTC	19	irq_cfg_0<19>	ssi_mst_intr	
IRQ_INTC	20	irq_cfg_0<20>	ssi_rxu_intr	
IRQ_INTC	21	irq_cfg_0<21>	ssi_rxo_intr	
IRQ_INTC	22	irq_cfg_0<22>	ssi_rxf_intr	
IRQ_INTC	23	irq_cfg_0<23>	ssi_txo_intr	
IRQ_INTC	24	irq_cfg_0<24>	ssi_txe_intr	
IRQ_INTC	25	irq_cfg_0<25>	Intr_basic_timer1_1	
IRQ_INTC	26	irq_cfg_0<26>	Intr_basic_timer1_2	
IRQ_INTC	27	irq_cfg_0<27>	Intr_basic_timer1_3	
IRQ_INTC	28	irq_cfg_0<28>	Intr_basic_timer1_4	
IRQ_INTC	29	irq_cfg_0<29>	Intr_basic_timer2_1	
IRQ_INTC	30	irq_cfg_0<30>	Intr_basic_timer2_2	
IRQ_INTC	31	irq_cfg_0<31>	Intr_basic_timer2_3	
IRQ_INTC	32	irq_cfg_1<0>	Intr_basic_timer2_4	gpio_intr_0
IRQ_INTC	33	irq_cfg_1<1>	ic_gen_call_intr	gpio_intr_1
IRQ_INTC	34	irq_cfg_1<2>	ic_rx_full_intr	gpio_intr_2
IRQ_INTC	35	irq_cfg_1<3>	ic_rd_req_intr	gpio_intr_3

Domain	#	src_sel	irq_cfg<n> = 1'b0	irq_cfg<n> = 1'b1
IRQ_INTC	36	irq_cfg_1<4>	ic_start_det_intr	gpio_intr_4
IRQ_INTC	37	irq_cfg_1<5>	ic_restart_det_intr	gpio_intr_5
IRQ_INTC	38	irq_cfg_1<6>	ic_stop_det_intr	gpio_intr_6
IRQ_INTC	39	irq_cfg_1<7>	ic_activity_intr	gpio_intr_7
IRQ_INTC	40	irq_cfg_1<8>	ic_tx_empty_intr	gpio_intr_8
IRQ_INTC	41	irq_cfg_1<9>	ic_tx_abrt_intr	gpio_intr_9
IRQ_INTC	42	irq_cfg_1<10>	ic_tx_over_intr	gpio_intr_10
IRQ_INTC	43	irq_cfg_1<11>	ic_rx_done_intr	gpio_intr_11
IRQ_INTC	44	irq_cfg_1<12>	ic_rx_under_intr	gpio_intr_12
IRQ_INTC	45	irq_cfg_1<13>	ic_rx_over_intr	gpio_intr_13
IRQ_INTC	46	irq_cfg_1<14>		gpio_intr_14
IRQ_INTC	47	irq_cfg_1<15>		gpio_intr_15
IRQ_INTC	48	irq_cfg_1<16>	adc_inj_eoc_intr	gpio_intr_16
IRQ_INTC	49	irq_cfg_1<17>	adc_inj_eos_intr	gpio_intr_17
IRQ_INTC	50	irq_cfg_1<18>	adc_reg_eoc_intr	gpio_intr_18
IRQ_INTC	51	irq_cfg_1<19>	adc_reg_eos_intr	gpio_intr_19
IRQ_INTC	52	irq_cfg_1<20>	adc_inj_awd_intr	gpio_intr_20
IRQ_INTC	53	irq_cfg_1<21>	adc_reg_awd_intr	gpio_intr_21
IRQ_INTC	54	irq_cfg_1<22>		gpio_intr_22
IRQ_INTC	55	irq_cfg_1<23>		gpio_intr_23
IRQ_INTC	56	irq_cfg_1<24>	led_prd_intr	gpio_intr_24
IRQ_INTC	57	irq_cfg_1<25>	led_com_intr	gpio_intr_25
IRQ_INTC	58	irq_cfg_1<26>	led_frm_intr	gpio_intr_26
IRQ_INTC	59	irq_cfg_1<27>	led_udt_intr	gpio_intr_27
IRQ_INTC	60	irq_cfg_1<28>	afc_norm_intr	gpio_intr_28
IRQ_INTC	61	irq_cfg_1<29>	afc_loop_intr	gpio_intr_29
IRQ_INTC	62	irq_cfg_1<30>	afc_err_intr	gpio_intr_30
IRQ_INTC	63	irq_cfg_1<31>	int_ctsm2host	gpio_intr_31

8.4.8 SYS_DMA_CFG

Base address: 0x4000_0800
Address offset: 0x30
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**DMA_CFG_HWCHx[1:0]**: DMA handshaking interface x (x= 0 to 15) Selection 2'bxx: refer to Table 17 and Table 18.	R/W

The SYS_DMA_CFG and SYS_DMA_CFG_2 forms the DMA handshaking interface assignment for the DMA controller. Handshaking interface assignment is shown in the following table.

Table 17 DMA handshaking interface assignment 1.

dma_cfg_2<n+1,n>, dma_cfg<n+1:n>	Interface #	4'b0000	4'b0001	4'b0010	4'b0011
dma_cfg_2<1:0>, dma_cfg<1:0>	0	UART1_dma_tx_req	I2C1_dma_tx_req	SPI1_dma_tx_req	LED_dma_frm_req
dma_cfg_2<3:2>, dma_cfg<3:2>	1	UART1_dma_rx_req	I2C1_dma_rx_req	SPI1_dma_rx_req	LED_dma_udt_req
dma_cfg_2<5:4>, dma_cfg<5:4>	2	UART2_dma_tx_req		TIMA1_dma_req_trg_com	LED_dma_com_req
dma_cfg_2<7:6>, dma_cfg<7:6>	3	UART2_dma_rx_req			TIMG1_dma_req_trg_com
dma_cfg_2<9:8>, dma_cfg<9:8>	4		UART3_dma_tx_req		
dma_cfg_2<11:10>, dma_cfg<11:10>	5		UART3_dma_rx_req		
dma_cfg_2<13:12>, dma_cfg<13:12>	6	ADC_eos_inj	UART4_dma_tx_req	TIMA1_dma_req_cc1	
dma_cfg_2<15:14>, dma_cfg<15:14>	7	ADC_eos_reg	UART4_dma_rx_req	TIMA1_dma_req_cc4	
dma_cfg_2<17:16>, dma_cfg<17:16>	8	I2C1_dma_tx_req	SPI1_dma_tx_req	TIMA1_dma_req_cc2	
dma_cfg_2<19:18>, dma_cfg<19:18>	9	I2C1_dma_rx_req	SPI1_dma_rx_req	TIMA1_dma_req_cc3	
dma_cfg_2<21:20>, dma_cfg<21:20>	10	SPI2_dma_tx_req	I2C2_dma_tx_req		
dma_cfg_2<23:22>, dma_cfg<23:22>	11	SPI2_dma_rx_req	I2C2_dma_rx_req		TiIMG1_dma_req_udt

dma_cfg_2<n+1,n>, dma_cfg<n+1:n>	Interface #	4'b0000	4'b0001	4'b0010	4'b0011
dma_cfg_2<25:24>, dma_cfg<25:24>	12	SPI1_dma_tx_req	UART3_dma_tx_req	TIMA1_dma_req_udt	TIMG1_dma_req_cc2
dma_cfg_2<27:26>, dma_cfg<27:26>	13	SPI1_dma_rx_req	UART3_dma_rx_req		TIMG1_dma_req_cc3
dma_cfg_2<29:28>, dma_cfg<29:28>	14	SPI2_dma_tx_req	I2C1_dma_tx_req	UART1_dma_tx_req	TIMG1_dma_req_cc1
dma_cfg_2<31:30>, dma_cfg<31:30>	15	SPI2_dma_rx_req	I2C1_dma_rx_req	UART1_dma_rx_req	TIMG1_dma_req_cc4

Table 18 DMA handshaking interface assignment 2.

dma_cfg_2<n+1,n>, dma_cfg<n+1:n>	Interface #	4'b0100	4'b1000	4'b1100	4'b1101
dma_cfg_2<1:0>, dma_cfg<1:0>	0			TIMG3_dma_req_trg_com	
dma_cfg_2<3:2>, dma_cfg<3:2>	1	TIMA2_dma_req_trg_com			TIMG4_dma_req_trg_com
dma_cfg_2<5:4>, dma_cfg<5:4>	2	SPI2_dma_tx_req		TIMG3_dma_req_udt	
dma_cfg_2<7:6>, dma_cfg<7:6>	3	SPI2_dma_rx_req	TIMG2_dma_req_trg_com		TIMG4_dma_req_udt
dma_cfg_2<9:8>, dma_cfg<9:8>	4			TIMG3_dma_req_cc1	
dma_cfg_2<11:10>, dma_cfg<11:10>	5			TIMG3_dma_req_cc4	
dma_cfg_2<13:12>, dma_cfg<13:12>	6			TIMG3_dma_req_cc2	
dma_cfg_2<15:14>, dma_cfg<15:14>	7			TIMG3_d6ma_req_cc3	
dma_cfg_2<17:16>, dma_cfg<17:16>	8	TIMA2_dma_req_cc1			
dma_cfg_2<19:18>, dma_cfg<19:18>	9	TIMA2_dma_req_cc4			
dma_cfg_2<21:20>, dma_cfg<21:20>	10	TIMA2_dma_req_cc2	TIMG2_dma_req_udt	UART4_dma_tx_req	TIMG4_dma_req_cc1
dma_cfg_2<23:22>, dma_cfg<23:22>	11	TIMA2_dma_req_cc3		UART4_dma_rx_req	TIMG4_dma_req_cc4
dma_cfg_2<25:24>, dma_cfg<25:24>	12		TIMG2_dma_req_cc1		TIMG4_dma_req_cc2
dma_cfg_2<27:26>, dma_cfg<27:26>	13		TIMG2_dma_req_cc4		TIMG4_dma_req_cc3
dma_cfg_2<29:28>, dma_cfg<29:28>	14	TIMA2_dma_req_udt	TIMG2_dma_req_cc2		
dma_cfg_2<31:30>, dma_cfg<31:30>	15		TIMG2_dma_req_cc3		

8.4.9 SYS_WDT_CFG

Base address: 0x4000_0800
Address offset: 0x34
Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31>	**WDTW_CFG_RST**: Windowed Watch-dog reset 1: Normal 0: Reset	R/W
<30:2>	Reserved	R/W
<1>	**WDTW_CFG_SPD**: Windowed Watch-dog speed up	R/W
<0>	**WDTW_CFG_EXTCLK**: Windowed Watch-dog external clock enable	R/W

8.4.10 SYS_SOFT_RESET_CFG

Base address: 0x4000_0800
Address offset: 0x38
Type: Read/Write
Reset value: 0xFFFF_FFFF

Bit	Description	Type
<31:23>	Reserved	R/W
<22>	**RST_CFG_TIMG4**: Soft reset of General Purpose Timer (TIM_GP) 1'b0: Soft reset 1'b1: No reset	R/W
<21>	**RST_CFG_TIMG3**: Soft reset of General Purpose Timer (TIM_GP) 1'b0: Soft reset 1'b1: No reset	R/W
<20>	**RST_CFG_TIMG2**: Soft reset of General Purpose Timer (TIM_GP) 1'b0: Soft reset 1'b1: No reset	R/W
<19>	**RST_CFG_TIMA2**: Soft reset of advanced Timer (ePWM) 1'b0: Soft reset 1'b1: No reset	R/W
<18>	**RST_CFG_TIMG1**: Soft reset of General Purpose Timer (TIM_GP) 1'b0: Soft reset 1'b1: No reset	R/W
<17>	**RST_CFG_TIMA1**: Soft reset of advanced Timer (ePWM) 1'b0: Soft reset 1'b1: No reset	R/W
<16>	**RST_CFG_GPIO**: Soft reset of GPIO Debounce 1'b0: Soft reset 1'b1: No reset	R/W
<15>	**RST_CFG_SPI2**: Soft reset of SPI2 1'b0: Soft reset 1'b1: No reset	R/W
<14>	**RST_CFG_SPI1**: Soft reset of SPI1 1'b0: Soft reset 1'b1: No reset	R/W
<13>	**RST_CFG_TIMB24**: Soft reset of Basic Timer 2, Channel 4 1'b0: Soft reset 1'b1: No reset	R/W
<12>	**RST_CFG_TIMB23**: Soft reset of Basic Timer 2, Channel 3 1'b0: Soft reset 1'b1: No reset	R/W

Bit	Description	Type
<11>	**RST_CFG_TIMB22**: Soft reset of Basic Timer 2, Channel 2 1'b0: Soft reset 1'b1: No reset	R/W
<10>	**RST_CFG_TIMB21**: Soft reset of Basic Timer 2, Channel 1 1'b0: Soft reset 1'b1: No reset	R/W
<9>	**RST_CFG_TIMB14**: Soft reset of Basic Timer 1, Channel 4 1'b0: Soft reset 1'b1: No reset	R/W
<8>	**RST_CFG_TIMB13**: Soft reset of Basic Timer 1, Channel 3 1'b0: Soft reset 1'b1: No reset	R/W
<7>	**RST_CFG_TIMB12**: Soft reset of Basic Timer 1, Channel 2 1'b0: Soft reset 1'b1: No reset	R/W
<6>	**RST_CFG_TIMB11**: Soft reset of Basic Timer 1, Channel 1 1'b0: Soft reset 1'b1: No reset	R/W
<5>	**RST_CFG_I2C2**: Soft reset of I2C2 1'b0: Soft reset 1'b1: No reset	R/W
<4>	**RST_CFG_I2C1**: Soft reset of I2C1 1'b0: Soft reset 1'b1: No reset	R/W
<3>	**RST_CFG_UART4**: Soft reset of UART 4 1'b0: Soft reset 1'b1: No reset	R/W
<2>	**RST_CFG_UART3**: Soft reset of UART 3 1'b0: Soft reset 1'b1: No reset	R/W
<1>	**RST_CFG_UART2**: Soft reset of UART 2 1'b0: Soft reset 1'b1: No reset	R/W
<0>	**RST_CFG_UART1**: Soft reset of UART 1 1'b0: Soft reset 1'b1: No reset	R/W

8.4.11 SYS_DMA_CFG_2

Base address: 0x4000_0800
Address offset: 0x3C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**DMA_CFG_2_HWCHx[3:2]**: DMA handshaking interface x (x= 15 to 0) Selection 2'bxx: refer to Table 17 and Table 18.	R/W

8.4.12 SYS_TIMER_ADV_CFG

Base address: 0x4000_0800
Address offset: 0x40
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:40>	Reserved	R/W

8.4.13 SYS_GPIO_ENAUX_A

Base address: 0x4000_0800
Address offset: 0x50
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**GPIOAUX_A_x**: GPIO Port A line x (x=0 to 31) AUX in/output enable control 1'b0: Input 1'b1: Output	R/W

GPIO AUX port A is reserved.

8.4.14 SYS_GPIO_ENAUX_B

Base address: 0x4000_0800
Address offset: 0x54
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**GPIOAUX_B_x**: GPIO port B line x (x=0 to 31) AUX in/output enable control 1'b0: Input 1'b1: Output	R/W

GPIO AUX port B assignment is shown in the following table.
Note: when set GPIO_AUX to input mode, the ICG (TIMER_EXCLK_ENABLE) need to be selected.

Table 19 GPIO AUX port B assignment

Bit position	Aux_PortB_In	Aux_PortB_Out
<31:27>	Reserved	Reserved
<26>	Reserved	o_epwm_ch1_p
<25>	exclk_timer_b0_3	timer_b0_3_toggle
<24>	i_timer_adv_etr	Reserved
<23>	i_timer_gp_etr	Reserved
<22>	i_timer_gp_ch3	Reserved
<21:19>	Reserved	Reserved
<18>	i_timer_gp_ch1	Reserved
<17>	Reserved	Reserved
<16>	i_timer_gp_ch2	Reserved
<15>	i_timer_gp_ch4	Reserved
<14>	exclk_timer_b0_1	timer_b0_1_toggle
<13>	exclk_timer_b0_2	timer_b0_2_toggle
<12>	exclk_timer_b0_3	timer_b0_3_toggle
<11>	exclk_timer_b1_0	timer_b1_0_toggle
<10>	exclk_timer_b1_1	timer_b1_1_toggle
<9>	exclk_timer_b1_2	timer_b1_2_toggle
<8>	exclk_timer_b1_3	timer_b1_3_toggle
<7>	Reserved	Reserved
<6>	exclk_timer_b1_2	timer_b1_2_toggle
<5>	exclk_timer_b1_1	timer_b1_1_toggle
<4:2>	Reserved	Reserved

Bit position	Aux_PortB_In	Aux_PortB_Out
<1>	Reserved	timer_b0_0_toggle
<0>	exclk_timer_b0_0	Reserved

8.4.15 SYS_GPIO_ENAUX_C

Base address: 0x4000_0800
Address offset: 0x58
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**GPIOAUX_C_x**: GPIO port C line x (x=0 to 31) AUX in/output enable control 1'b0: Input 1'b1: Output	R/W

GPIO AUX port C assignment is shown in the following table.
Note: when set GPIO_AUX to input mode, the ICG (TIMER_EXCLK_ENABLE) need to be selected.

Table 20 GPIO AUX port C assignment

Bit position	Aux_PortC_In	Aux_PortC_Out
<31:27>	Reserved	Reserved
<26>	Reserved	o_adv_timer_brkout2
<25>	Reserved	o_adv_timer_brkout1
<24:19>	Reserved	Reserved
<18>	i_timer_adv_ch2	o_epwm_ch2_n
<17>	Reserved	o_epwm_ch2_p
<16>	i_timer_adv_ch1	o_epwm_ch1_n
<15:10>	Reserved	Reserved
<9>	i_timer_adv_ch3	o_epwm_ch3_n
<8>	Reserved	o_epwm_ch3_p
<7:3>	Reserved	Reserved
<2>	ext_timer_b1_0	timer_b1_0_toggle
<1>	ext_timer_b1_3	timer_b1_3_toggle
<0>	i_timer_adv_brkin2	Reserved

8.4.16 SYS_GPIO_ENAUX_D

Base address: 0x4000_0800
Address offset: 0x5C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**GPIOAUX_D_x**: GPIO Port D line x(x=0 to 31)AUX in/output enable control 1'b0: Output 1'b1: Input	R/W

GPIO AUX port D assignment s shown in the following table.
Note: when set GPIO_AUX to input mode, the ICG (TIMER_EXCLK_ENABLE) need to be selected.

Table 21 GPIO AUX port D assignment

Bit position	Aux_portD_In	Aux_PortD_Out
<31:16>	Reserved	Reserved
<15>	i_timer_adv_ch4	o_epwm_ch4
<14:1>	Reserved	Reserved
<0>	i_timer_adv_brkin1	Reserved

8.4.17 SYS_UID_REG_1

Base address: 0x4000_0800
Address offset: 0xF4
Type: Read
Reset value: 0xFFFF_FFFF

Bit	Description	Type
<31:0>	**UID_REG_1_UID[31:0]**: UID low word	R

8.4.18 SYS_UID_REG_2

Base address: 0x4000_0800
Address offset: 0xF8
Type: Read
Reset value: 0xFFFF_FFFF

Bit	Description	Type
<31:0>	**UID_REG_2_UID[31:0]**: UID mid word	R

8.4.19 SYS_UID_REG_3

Base address: 0x4000_0800
Address offset: 0xFC
Type: Read
Reset value: 0xFFFF_FFFF

Bit	Description	Type
<31:0>	**UID_REG_3_UID[31:0]**: UID high word	R

96bit UID location

UID_REG_3				UID_REG_2				UID_REG_1			
Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0

8.4.20 SYS_UART_ADCFG_REG

Base address: 0x4000_0800
Address offset: 0x0100
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:27>	Reserved.	R/W
<26>	**UART4_ADCFG_CLR**: UART4 auto baud rate detection flag clear 1'b0: No action 1' b1: Clear the flag Note: This bit must be set to 1 when auto detection is completed and then written to 0.	R/W
<25>	**UART4_ADCFG_IE**: UART4 auto baud rate detection interrupt enable 1'b0: UART4 auto baud rate interrupt disabled 1'b1: UART4 auto baud rate interrupt enabled	R/W
<24>	**UART4_ADCFG_EN**: UART4 auto baud rate detection enable 1'b0: UART4 auto baud rate detection disabled 1'b1: UART4 auto baud rate detection enabled	R/W
<23:19>	Reserved.	R/W
<18>	**UART3_ADCFG_CLR**: UART3 auto baud rate detection flag clear 1'b0: No action 1' b1: Clear the flag Note: This bit must be set to 1 when auto detection is completed and then written to 0.	R/W
<17>	**UART3_ADCFG_IE**: UART3 auto baud rate detection interrupt enable 1'b0: UART3 auto baud rate interrupt disabled 1'b1: UART3 auto baud rate interrupt enabled	R/W
<16>	**UART3_ADCFG_EN**: UART3 auto baud rate detection enable	R/W

Bit	Description	Type
	1’b0: UART3 auto baud rate detection disabled 1’b1: UART3 auto baud rate detection enabled	
<15:11>	Reserved.	R/W
<10>	**UART2_ADCFG_CLR**: UART2 auto baud rate detection flag clear 1’b0: No action 1’ b1: Clear the flag Note: This bit must be set to 1 when auto detection is completed and then written to 0.	R/W
<9>	**UART2_ADCFG_IE**: UART2 auto baud rate detection interrupt enable 1’b0: UART2 auto baud rate interrupt disabled 1’b1: UART2 auto baud rate interrupt enabled	R/W
<8>	**UART2_ADCFG_EN**: UART2 auto baud rate detection enable 1’b0: UART2 auto baud rate detection disabled 1’b1: UART2 auto baud rate detection enabled	R/W
<7:3>	Reserved.	R/W
<2>	**UART1_ADCFG_CLR**: UART1 auto baud rate detection flag clear 1’b0: No action 1’ b1: Clear the flag Note: This bit must be set to 1 when auto detection is completed and then written to 0.	R/W
<1>	**UART1_ADCFG_IE**: UART1 auto baud rate detection interrupt enable 1’b0: UART1 auto baud rate interrupt disabled 1’b1: UART1 auto baud rate interrupt enabled	R/W
<0>	**UART1_ADCFG_EN**: UART1 auto baud rate detection enable 1’b0: UART1 auto baud rate detection disabled 1’b1: UART1 auto baud rate detection enabled	R/W

8.4.21 SYS_UART_ADIF_REG

Base address: 0x4000_0800
Address offset: 0x0104
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved.	R/W
<3>	**UART4_ADFLG**: UART4 auto baud rate detection completed 0: No baud rate update. 1: Auto detection finished, update the baud rate	R/W
<2>	**UART3_ADFLG**: UART3 auto baud rate detection completed 0: No baud rate update. 1: Auto detection finished, update the baud rate	R/W
<1>	**UART2_ADFLG**: UART2 auto baud rate detection completed 1'b0: No baud rate update. 1'b1: Auto detection finished, update the baud rate	R/W
<0>	**UART1_ADFLG**: UART1 auto baud rate detection completed 1'b0: No baud rate update. 1'b1: Auto detection finished, update the baud rate	R/W

8.4.22 SYS_UART1_ADBAUD_REG

Base address: 0x4000_0800
Address offset: 0x0108
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**UART1_ADBR[31:0]**:UART1 baud rate detected	R/W

8.4.23 SYS_UART2_ADBAUD_REG

Base address: 0x4000_0800
Address offset: 0x010C
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**UART2_BR[31:0]**:UART2 baud rate detected	R/W

8.4.24 SYS_UART3_ADBAUD_REG

Base address: 0x4000_0800
Address offset: 0x0110
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**UART3_ADBR[31:0]**:UART3 baud rate detected	R/W

8.4.25 SYS_UART4_ADBAUD_REG

Base address: 0x4000_0800
Address offset: 0x0114
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**UART4_ADBR[31:0]**:UART4 baud rate detected	R/W

8.4.26 SYS_CTSM_CTRL

Base address: 0x4000_0800
Address offset: 0x0120
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:5>	Reserved.	R/W
<4>	**CTSM_CTRL_CLRINT**: Clear the interrupt generation register in CTSM	R/W
<3:1>	Reserved.	R/W
<0>	**CTSM_CTRL_EN**: CTSM module enable	R/W

8.4.27 SYS_CTSM_INT_HOST2CTSM

Base address: 0x4000_0800
Address offset: 0x0124
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	**CTSM_INT_INTGEN**: interrupt indicator, from host to CTSM	R/W
<30:15>	Reserved.	R/W
<16>	**CTSM_INT_SOINIT**: Sensor offset initial	R/W
<15>	**CTSM_INT_MSCAN**: measurement scan	R/W
<14>	**CTSM_INT_AFC**: AF calibration	R/W
<13>	**CTSM_INT_GETST**: get CTSM status	R/W
<12>	**CTSM_INT_JINIT**: judgment initial	R/W
<11:7>	Reserved.	R/W
<6>	**CTSM_INT_IDCSTRT**: internal current measurement start	R/W
<5>	**CTSM_INT_TCSTRT**: temperature correction start	R/W
<4>	**CTSM_INT_DIASTRT**: diagnosis start	R/W
<3>	**CTSM_INT_DCSTRT**: current measurement start	R/W
<2>	**CTSM_INT_CSTRT**: correction start	R/W
<1>	**CTSM_INT_MSTRT**: measurement start	R/W
<0>	**CTSM_INT_MSTOP**: measurement stop	R/W

8.4.28 SYS_CTSM_INT_CTSM2HOST

Base address: 0x4000_0800
Address offset: 0x0128
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:5>	Reserved.	R/W
<7>	**CTSM_SR_ERROR**: error report	R/W
<6>	**CTSM_SR_JCLPT**: judgment completed	R/W
<5>	**CTSM_SR_SCLPT**: scan completed	R/W
<4>	**CTSM_SR_MCLPT**: measurement completed	R/W
<3>	**CTSM_SR_BUSY**: busy	R/W
<2>	**CTSM_SR_JRRD**: judgment result read request	R/W
<1>	**CTSM_SR_MARRD**: measurement average result read request	R/W
<0>	**CTSM_SR_MRRD**: measurement result read request	R/W

8.4.29 SYS_HDIV_A

Base address: 0x4000_0800
Address offset: 0x0130
Type: Read/Write
Reset value: 0x0000_0001

Bit	Description	Type
<31:0>	**HDIV_A[31:0]**: HDIV dividend	R/W

8.4.30 SYS_HDIV_B

Base address: 0x4000_0800
Address offset: 0x0134
Type: Read/Write
Reset value: 0x0000_0001

Bit	Description	Type
<31:0>	**HDIV_B[31:0]**: HDIV divisor	R/W

8.4.31 SYS_HDIV_Q

Base address: 0x4000_0800
Address offset: 0x0138
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**HDOV_Q[31:0]**: HDIV quotient	R/W

8.4.32 SYS_HDIV_R

Base address: 0x4000_0800
Address offset: 0x013C
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**HDIV_R[31:0]**: HDIV remainder	R/W

8.4.33 SYS_AFC_CTRL

Base address: 0x4000_0800
Address offset: 0x0140
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	Reserved.	R/W
<29:16>	**AFCCTRL_RFCNT[13:0]**: Reference clock cycle for calibration The reference clock cycle for calibration specifies the total time of a calibration cycle where the total time of the calibration cycle = Number of reference clock cycle for calibration / Frequency of the reference clock.	R/W
<15:8>	**AFCCTRL_DIFFCNT[7:0]**: Calibration difference count value The calibration difference count value is related to the calibration precision and defines the termination criteria for the auto-calibration logic. The auto-calibration process stops when the difference count value computed during calibration is smaller than the value specified.	R/W
<7:6>	Reserved.	R/W
<5>	**AFCCTRL_CBCLKS**: Calibration Clock select Select the clock to be calibrated by setting this bit. 1'b0: 32K RC 1'b1: 16M RC	R/W
<4>	**AFCCTRL_AEIE**: Error interrupt enable When this bit is enabled, an interrupt would be triggered if the AEIF bit is set at the SYS_AFC_RES register. 1'b0: disable 1'b1: enable	R/W
<3>	**AFCCTRL_ANIE**: Normal interrupt enable When this bit is enabled, an interrupt would be triggered if the ANIF bit is set at the SYS_AFC_RES register. 1'b0: disable 1'b1: enable	R/W
<2>	**AFCCTRL_ALIE**: Loop interrupt enable When this bit is enabled, an interrupt would be triggered if the ALIF bit is set at the SYS_AFC_RES register. 1'b0: disable 1'b1: enable	R/W
<1>	**AFCCTRL_AFCBYS**: AFC bypass enable. 1'b0: disable bypass; 2'b1: enable bypass.	R/W
<0>	**AFCCTRL_AFCEN**: Enable AFC	R/W

	Enable the auto-calibration process by setting this bit. 1'b0: disable AFC; 1'b1: enable AFC.	

8.4.34 SYS_AFC_CFG

Base address: 0x4000_0800
Address offset: 0x0144
Type: Read/Write
Reset value: 0x0000_0030

Bit	Description	Type
<31:28>	Reserved.	R/W
<27:8>	**AFCCFG_STDCNT[19:0]**: Count value expected for the calibration process Set this value to the total number of clock cycle expected from the target RC clock for the calibration cycle defined.	R/W
<7:0>	**AFCCFG_CBADJ[7:0]**: Seed value for the auto-calibration logic Set an initial seed value for the auto-calibration logic to search for the calibration coefficient.	R/W

8.4.35 SYS_AFC_RES

Base address: 0x4000_0800
Address offset: 0x0148
Type: Read/Write
Reset value: 0x0001_0000

Bit	Description	Type
<31:16>	**AFCRES_RFDIV[15:0]**: Reference clock divider ratio Specify the reference clock divider ratio for the reference clock. The reference clock frequency would be divided down by the value set in this field.	R/W
<15:11>	Reserved.	R/W
<10>	**AFCRES_AEIF**: Auto frequency calibration error interrupt flag 1'b0: Write 0 to clear the flag 1'b1: AFC error interrupt flag	R/W0
<9>	**AFCRES_ANIF**: Calibration completed within tolerance interrupt flag 1'b0: Write 0 to clear the flag 1'b1: Interrupt flag to indicate that the calibration completed within the tolerance specified in AFCCTRL_DIFFCNT[7:0]	R/W0
<8>	**AFCRES_ALIF**: Calibration completed outside the tolerance interrupt flag 1'b0: Write 0 to clear the flag 1'b1: Interrupt flag to indicate that the calibration completed but it does not meet the tolerance specified in AFCCTRL_DIFFCNT[7:0]	R/W0
<7:0>	**AFCRES_ADJRES[7:0]**: Calibration Results	R

	This field stored the auto-calibration results determined by the hardware. This value should be programmed to the respective RC clock configuration register to obtain the calibrated RC clock.	

8.4.36 SYS_TIM_A1_IMX

Base address: 0x4000_0800
Address offset: 0x0180
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:17>	Reserved.	R/W
<18:16>	**TIMA1IMX_ETR[2:0]**: TIM_A1 ETR input port source 3'b000: Primary TIM_A1 IO ETR selected for ETR input 3'b001: Primary TIM_A2 IO ETR selected for ETR input 3'b010: Primary TIM_G1 IO ETR selected for ETR input 3'b011: Primary TIM_G2 IO ETR selected for ETR input 3'b100: Primary TIM_G3 IO ETR selected for ETR input 3'b101: Primary TIM_G4 IO ETR selected for ETR input 3'b11x: TIM_A1 ETR input disable	R/W
<15>	Reserved.	R/W
<14:12>	**TIMA1IMX_CH_4[2:0]**: TIM_A1 channel 4 input port source 3'b000: Primary TIM_A1 IO channel 4 selected for channel 4 input 3'b001: Primary TIM_A2 IO channel 4 selected for channel 4 input 3'b010: Primary TIM_G1 IO channel 4 selected for channel 4 input 3'b011: Primary TIM_G2 IO channel 4 selected for channel 4 input 3'b100: Primary TIM_G3 IO channel 4 selected for channel 4 input 3'b101: Primary TIM_G4 IO channel 4 selected for channel 4 input 3'b11x: TIM_A1 channel 4 input disable	R/W
<11>	Reserved.	R/W
<10:8>	**TIMA1IMX_CH_3[2:0]**: TIM_A1 channel 3 input port source 3'b000: Primary TIM_A1 IO channel 3 selected for channel 3 input 3'b001: Primary TIM_A2 IO channel 3 selected for channel 3 input 3'b010: Primary TIM_G1 IO channel 3 selected for channel 3 input 3'b011: Primary TIM_G2 IO channel 3 selected for channel 3 input 3'b100: Primary TIM_G3 IO channel 3 selected for channel 3 input 3'b101: Primary TIM_G4 IO channel 3 selected for channel 3 input 3'b11x: TIM_A1 channel 3 input disable	R/W
<7>	Reserved.	R/W
<6:4>	**TIMA1IMX_CH_2[2:0]**: TIM_A1 channel 2 input port source 3'b000: Primary TIM_A1 IO channel 2 selected for channel 2 input	R/W

Bit	Description	Type
	3'b001: Primary TIM_A2 IO channel 2 selected for channel 2 input 3'b010: Primary TIM_G1 IO channel 2 selected for channel 2 input 3'b011: Primary TIM_G2 IO channel 2 selected for channel 2 input 3'b100: Primary TIM_G3 IO channel 2 selected for channel 2 input 3'b101: Primary TIM_G4 IO channel 2 selected for channel 2 input 3'b11x: TIM_A1 channel 2 input disable	
<3>	Reserved.	R/W
<2:0>	**TIMA1IMX_CH_1[2:0]**: TIM_A1 channel 1 input port source 3'b000: Primary TIM_A1 IO channel 1 selected for channel 1 input 3'b001: Primary TIM_A2 IO channel 1 selected for channel 1 input 3'b010: Primary TIM_G1 IO channel 1 selected for channel 1 input 3'b011: Primary TIM_G2 IO channel 1 selected for channel 1 input 3'b100: Primary TIM_G3 IO channel 1 selected for channel 1 input 3'b101: Primary TIM_G4 IO channel 1 selected for channel 1 input 3'b11x: TIM_A1 channel 1 input disable	R/W

8.4.37 SYS_TIM_A2_IMX

Base address: 0x4000_0800
Address offset: 0x0184
Type: Read/Write
Reset value: 0x0001_1111

Bit	Description	Type
<31:17>	Reserved.	R/W
<18:16>	**TIMA2IMX_ETR[2:0]**: TIM_A2 ETR input port source 3'b000: Primary TIM_A1 IO ETR selected for ETR input 3'b001: Primary TIM_A2 IO ETR selected for ETR input 3'b010: Primary TIM_G1 IO ETR selected for ETR input 3'b011: Primary TIM_G2 IO ETR selected for ETR input 3'b100: Primary TIM_G3 IO ETR selected for ETR input 3'b101: Primary TIM_G4 IO ETR selected for ETR input 3'b11x: TIM_A2 ETR input disable	R/W
<15>	Reserved.	R/W
<14:12>	**TIMA2IMX_CH_4[2:0]**: TIM_A2 channel 4 input port source 3'b000: Primary TIM_A1 IO channel 4 selected for channel 4 input 3'b001: Primary TIM_A2 IO channel 4 selected for channel 4 input 3'b010: Primary TIM_G1 IO channel 4 selected for channel 4 input 3'b011: Primary TIM_G2 IO channel 4 selected for channel 4 input 3'b100: Primary TIM_G3 IO channel 4 selected for channel 4 input 3'b101: Primary TIM_G4 IO channel 4 selected for channel 4 input 3'b11x: TIM_A2 channel 4 input disable	R/W

Bit	Description	Type
<11>	Reserved.	R/W
<10:8>	**TIMA2IMX_CH_3[2:0]**: TIM_A2 channel 3 input port source 3'b000: Primary TIM_A1 IO channel 3 selected for channel 3 input 3'b001: Primary TIM_A2 IO channel 3 selected for channel 3 input 3'b010: Primary TIM_G1 IO channel 3 selected for channel 3 input 3'b011: Primary TIM_G2 IO channel 3 selected for channel 3 input 3'b100: Primary TIM_G3 IO channel 3 selected for channel 3 input 3'b101: Primary TIM_G4 IO channel 3 selected for channel 3 input 3'b11x: TIM_A2 channel 3 input disable	R/W
<7>	Reserved.	R/W
<6:4>	**TIMA2IMX_CH_2[2:0]**: TIM_A2 channel 2 input port source 3'b000: Primary TIM_A1 IO channel 2 selected for channel 2 input 3'b001: Primary TIM_A2 IO channel 2 selected for channel 2 input 3'b010: Primary TIM_G1 IO channel 2 selected for channel 2 input 3'b011: Primary TIM_G2 IO channel 2 selected for channel 2 input 3'b100: Primary TIM_G3 IO channel 2 selected for channel 2 input 3'b101: Primary TIM_G4 IO channel 2 selected for channel 2 input 3'b11x: TIM_A2 channel 2 input disable	R/W
<3>	Reserved.	R/W
<2:0>	**TIMA2IMX_CH_1[2:0]**: TIM_A2 channel 1 input port source 3'b000: Primary TIM_A1 IO channel 1 selected for channel 1 input 3'b001: Primary TIM_A2 IO channel 1 selected for channel 1 input 3'b010: Primary TIM_G1 IO channel 1 selected for channel 1 input 3'b011: Primary TIM_G2 IO channel 1 selected for channel 1 input 3'b100: Primary TIM_G3 IO channel 1 selected for channel 1 input 3'b101: Primary TIM_G4 IO channel 1 selected for channel 1 input 3'b11x: TIM_A2 channel 1 input disable	R/W

8.4.38 SYS_TIM_G1_IMX

Base address: 0x4000_0800
Address offset: 0x0188
Type: Read/Write
Reset value: 0x0002_2222

Bit	Description	Type
<31:17>	Reserved.	R/W
<18:16>	**TIMG1IMX_ETR[2:0]**: TIM_G1 ETR input port source 3'b000: Primary TIM_A1 IO ETR selected for ETR input 3'b001: Primary TIM_A2 IO ETR selected for ETR input 3'b010: Primary TIM_G1 IO ETR selected for ETR input 3'b011: Primary TIM_G2 IO ETR selected for ETR input	R/W

Bit	Description	Type
	3’b100: Primary TIM_G3 IO ETR selected for ETR input 3’b101: Primary TIM_G4 IO ETR selected for ETR input 3’b11x: TIM_G1 ETR input disable	
<15>	Reserved.	R/W
<14:12>	**TIMG1IMX_CH_4[2:0]**: TIM_G1 channel 4 input port source 3’b000: Primary TIM_A1 IO channel 4 selected for channel 4 input 3’b001: Primary TIM_A2 IO channel 4 selected for channel 4 input 3’b010: Primary TIM_G1 IO channel 4 selected for channel 4 input 3’b011: Primary TIM_G2 IO channel 4 selected for channel 4 input 3’b100: Primary TIM_G3 IO channel 4 selected for channel 4 input 3’b101: Primary TIM_G4 IO channel 4 selected for channel 4 input 3’b11x: TIM_G1 channel 4 input disable	R/W
<11>	Reserved.	R/W
<10:8>	**TIMG1IMX_CH_3[2:0]**: TIM_G1 channel 3 input port source 3’b000: Primary TIM_A1 IO channel 3 selected for channel 3 input 3’b001: Primary TIM_A2 IO channel 3 selected for channel 3 input 3’b010: Primary TIM_G1 IO channel 3 selected for channel 3 input 3’b011: Primary TIM_G2 IO channel 3 selected for channel 3 input 3’b100: Primary TIM_G3 IO channel 3 selected for channel 3 input 3’b101: Primary TIM_G4 IO channel 3 selected for channel 3 input 3’b11x: TIM_G1 channel 3 input disable	R/W
<7>	Reserved.	R/W
<6:4>	**TIMG1IMX_CH_2[2:0]**: TIM_G1 channel 2 input port source 3’b000: Primary TIM_A1 IO channel 2 selected for channel 2 input 3’b001: Primary TIM_A2 IO channel 2 selected for channel 2 input 3’b010: Primary TIM_G1 IO channel 2 selected for channel 2 input 3’b011: Primary TIM_G2 IO channel 2 selected for channel 2 input 3’b100: Primary TIM_G3 IO channel 2 selected for channel 2 input 3’b101: Primary TIM_G4 IO channel 2 selected for channel 2 input 3’b11x: TIM_G1 channel 2 input disable	R/W
<3>	Reserved.	R/W
<2:0>	**TIMG1IMX_CH_1[2:0]**: TIM_G1 channel 1 input port source 3’b000: Primary TIM_A1 IO channel 1 selected for channel 1 input 3’b001: Primary TIM_A2 IO channel 1 selected for channel 1 input 3’b010: Primary TIM_G1 IO channel 1 selected for channel 1 input 3’b011: Primary TIM_G2 IO channel 1 selected for channel 1 input 3’b100: Primary TIM_G3 IO channel 1 selected for channel 1 input 3’b101: Primary TIM_G4 IO channel 1 selected for channel 1 input 3’b11x: TIM_G1 channel 1 input disable	R/W

8.4.39 SYS_TIM_G2_IMX

Base address: 0x4000_0800
Address offset: 0x018C
Type: Read/Write
Reset value: 0x0003_3333

Bit	Description	Type
<31:17>	Reserved.	R/W
<18:16>	**TIMG2IMX_ETR[2:0]**: TIM_G2 ETR input port source 3'b000: Primary TIM_A1 IO ETR selected for ETR input 3'b001: Primary TIM_A2 IO ETR selected for ETR input 3'b010: Primary TIM_G1 IO ETR selected for ETR input 3'b011: Primary TIM_G2 IO ETR selected for ETR input 3'b100: Primary TIM_G3 IO ETR selected for ETR input 3'b101: Primary TIM_G4 IO ETR selected for ETR input 3'b11x: TIM_ G2 ETR input disable	R/W
<15>	Reserved.	R/W
<14:12>	**TIMG2IMX_CH_4[2:0]**: TIM_ G2 channel 4 input port source 3'b000: Primary TIM_A1 IO channel 4 selected for channel 4 input 3'b001: Primary TIM_A2 IO channel 4 selected for channel 4 input 3'b010: Primary TIM_G1 IO channel 4 selected for channel 4 input 3'b011: Primary TIM_G2 IO channel 4 selected for channel 4 input 3'b100: Primary TIM_G3 IO channel 4 selected for channel 4 input 3'b101: Primary TIM_G4 IO channel 4 selected for channel 4 input 3'b11x: TIM_ G2 channel 4 input disable	R/W
<11>	Reserved.	R/W
<10:8>	**TIMG2IMX_CH_3[2:0]**: TIM_ G2 channel 3 input port source 3'b000: Primary TIM_A1 IO channel 3 selected for channel 3 input 3'b001: Primary TIM_A2 IO channel 3 selected for channel 3 input 3'b010: Primary TIM_G1 IO channel 3 selected for channel 3 input 3'b011: Primary TIM_G2 IO channel 3 selected for channel 3 input 3'b100: Primary TIM_G3 IO channel 3 selected for channel 3 input 3'b101: Primary TIM_G4 IO channel 3 selected for channel 3 input 3'b11x: TIM_ G2 channel 3 input disable	R/W
<7>	Reserved.	R/W
<6:4>	**TIMG2IMX_CH_2[2:0]**: TIM_ G2 channel 2 input port source 3'b000: Primary TIM_A1 IO channel 2 selected for channel 2 input 3'b001: Primary TIM_A2 IO channel 2 selected for channel 2 input 3'b010: Primary TIM_G1 IO channel 2 selected for channel 2 input 3'b011: Primary TIM_G2 IO channel 2 selected for channel 2 input 3'b100: Primary TIM_G3 IO channel 2 selected for channel 2 input	R/W

Bit	Description	Type
	3’b101: Primary TIM_G4 IO channel 2 selected for channel 2 input 3’b11x: TIM_ G2 channel 2 input disable	
<3>	Reserved.	R/W
<2:0>	**TIMG2IMX_CH_1[2:0]**: TIM_ G2 channel 1 input port source 3’b000: Primary TIM_A1 IO channel 1 selected for channel 1 input 3’b001: Primary TIM_A2 IO channel 1 selected for channel 1 input 3’b010: Primary TIM_G1 IO channel 1 selected for channel 1 input 3’b011: Primary TIM_G2 IO channel 1 selected for channel 1 input 3’b100: Primary TIM_G3 IO channel 1 selected for channel 1 input 3’b101: Primary TIM_G4 IO channel 1 selected for channel 1 input 3’b11x: TIM_ G2 channel 1 input disable	R/W

8.4.40 SYS_TIM_G3_IMX

Base address: 0x4000_0800
Address offset: 0x0190
Type: Read/Write
Reset value: 0x0004_4444

Bit	Description	Type
<31:17>	Reserved.	R/W
<18:16>	**TIMG3IMX_ETR[2:0]**: TIM_G2 ETR input port source 3’b000: Primary TIM_A1 IO ETR selected for ETR input 3’b001: Primary TIM_A2 IO ETR selected for ETR input 3’b010: Primary TIM_G1 IO ETR selected for ETR input 3’b011: Primary TIM_G2 IO ETR selected for ETR input 3’b100: Primary TIM_G3 IO ETR selected for ETR input 3’b101: Primary TIM_G4 IO ETR selected for ETR input 3’b11x: TIM_ G2 ETR input disable	R/W
<15>	Reserved.	R/W
<14:12>	**TIMG3IMX_CH_4[2:0]**: TIM_ G2 channel 4 input port source 3’b000: Primary TIM_A1 IO channel 4 selected for channel 4 input 3’b001: Primary TIM_A2 IO channel 4 selected for channel 4 input 3’b010: Primary TIM_G1 IO channel 4 selected for channel 4 input 3’b011: Primary TIM_G2 IO channel 4 selected for channel 4 input 3’b100: Primary TIM_G3 IO channel 4 selected for channel 4 input 3’b101: Primary TIM_G4 IO channel 4 selected for channel 4 input 3’b11x: TIM_ G2 channel 4 input disable	R/W
<11>	Reserved.	R/W
<10:8>	**TIMG3IMX_CH_3[2:0]**: TIM_ G2 channel 3 input port source 3’b000: Primary TIM_A1 IO channel 3 selected for channel 3 input 3’b001: Primary TIM_A2 IO channel 3 selected for channel 3 input	R/W

Bit	Description	Type
	3’b010: Primary TIM_G1 IO channel 3 selected for channel 3 input 3’b011: Primary TIM_G2 IO channel 3 selected for channel 3 input 3’b100: Primary TIM_G3 IO channel 3 selected for channel 3 input 3’b101: Primary TIM_G4 IO channel 3 selected for channel 3 input 3’b11x: TIM_ G2 channel 3 input disable	
<7>	Reserved.	R/W
<6:4>	**TIMG3IMX_CH_2[2:0]**: TIM_ G2 channel 2 input port source 3’b000: Primary TIM_A1 IO channel 2 selected for channel 2 input 3’b001: Primary TIM_A2 IO channel 2 selected for channel 2 input 3’b010: Primary TIM_G1 IO channel 2 selected for channel 2 input 3’b011: Primary TIM_G2 IO channel 2 selected for channel 2 input 3’b100: Primary TIM_G3 IO channel 2 selected for channel 2 input 3’b101: Primary TIM_G4 IO channel 2 selected for channel 2 input 3’b11x: TIM_ G2 channel 2 input disable	R/W
<3>	Reserved.	R/W
<2:0>	**TIMG3IMX_CH_1[2:0]**: TIM_ G2 channel 1 input port source 3’b000: Primary TIM_A1 IO channel 1 selected for channel 1 input 3’b001: Primary TIM_A2 IO channel 1 selected for channel 1 input 3’b010: Primary TIM_G1 IO channel 1 selected for channel 1 input 3’b011: Primary TIM_G2 IO channel 1 selected for channel 1 input 3’b100: Primary TIM_G3 IO channel 1 selected for channel 1 input 3’b101: Primary TIM_G4 IO channel 1 selected for channel 1 input 3’b11x: TIM_ G2 channel 1 input disable	R/W

8.4.41 SYS_TIM_G4_IMX

Base address: 0x4000_0800
Address offset: 0x0194
Type: Read/Write
Reset value: 0x0005_5555

Bit	Description	Type
<31:17>	Reserved.	R/W
<18:16>	**TIMG4IMX_ETR[2:0]**: TIM_G4 ETR input port source 3’b000: Primary TIM_A1 IO ETR selected for ETR input 3’b001: Primary TIM_A2 IO ETR selected for ETR input 3’b010: Primary TIM_G1 IO ETR selected for ETR input 3’b011: Primary TIM_G2 IO ETR selected for ETR input 3’b100: Primary TIM_G3 IO ETR selected for ETR input 3’b101: Primary TIM_G4 IO ETR selected for ETR input 3’b11x: TIM_ G4 ETR input disable	R/W
<15>	Reserved.	R/W

Bit	Description	Type
<14:12>	**TIMG4IMX_CH_4[2:0]**: TIM_ G4 channel 4 input port source 3'b000: Primary TIM_A1 IO channel 4 selected for channel 4 input 3'b001: Primary TIM_A2 IO channel 4 selected for channel 4 input 3'b010: Primary TIM_G1 IO channel 4 selected for channel 4 input 3'b011: Primary TIM_G2 IO channel 4 selected for channel 4 input 3'b100: Primary TIM_G3 IO channel 4 selected for channel 4 input 3'b101: Primary TIM_G4 IO channel 4 selected for channel 4 input 3'b11x: TIM_ G4 channel 4 input disable	R/W
<11>	Reserved.	R/W
<10:8>	**TIMG4IMX_CH_3[2:0]**: TIM_ G4 channel 3 input port source 3'b000: Primary TIM_A1 IO channel 3 selected for channel 3 input 3'b001: Primary TIM_A2 IO channel 3 selected for channel 3 input 3'b010: Primary TIM_G1 IO channel 3 selected for channel 3 input 3'b011: Primary TIM_G2 IO channel 3 selected for channel 3 input 3'b100: Primary TIM_G3 IO channel 3 selected for channel 3 input 3'b101: Primary TIM_G4 IO channel 3 selected for channel 3 input 3'b11x: TIM_ G4 channel 3 input disable	R/W
<7>	Reserved.	R/W
<6:4>	**TIMG4IMX_CH_2[2:0]**: TIM_ G4 channel 2 input port source 3'b000: Primary TIM_A1 IO channel 2 selected for channel 2 input 3'b001: Primary TIM_A2 IO channel 2 selected for channel 2 input 3'b010: Primary TIM_G1 IO channel 2 selected for channel 2 input 3'b011: Primary TIM_G2 IO channel 2 selected for channel 2 input 3'b100: Primary TIM_G3 IO channel 2 selected for channel 2 input 3'b101: Primary TIM_G4 IO channel 2 selected for channel 2 input 3'b11x: TIM_ G4 channel 2 input disable	R/W
<3>	Reserved.	R/W
<2:0>	**TIMG4IMX_CH_1[2:0]**: TIM_ G4 channel 1 input port source 3'b000: Primary TIM_A1 IO channel 1 selected for channel 1 input 3'b001: Primary TIM_A2 IO channel 1 selected for channel 1 input 3'b010: Primary TIM_G1 IO channel 1 selected for channel 1 input 3'b011: Primary TIM_G2 IO channel 1 selected for channel 1 input 3'b100: Primary TIM_G3 IO channel 1 selected for channel 1 input 3'b101: Primary TIM_G4 IO channel 1 selected for channel 1 input 3'b11x: TIM_ G4 channel 1 input disable	R/W

8.4.42 SYS_TIM_CMS_OC

Base address: 0x4000_0800
Address offset: 0x01A0
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:12>	Reserved	R/W
<11>	**TIMCMSOC_TIMG4_CH2_OS**: TIM_G3 output channel 2 complementary or single-ended output selection 1'b0: TIM_G3 complementary output (Ch2_n) selected 1'b1: TIM_G3 single-ended output (Ch6) selected	R/W
<10>	**TIMCMSOC_TIMG4_CH1_OS**: TIM_G3 output channel 1 complementary or single-ended output selection 1'b0: TIM_G3 complementary output (Ch1_n) selected 1'b1: TIM_G3 single-ended output (Ch5) selected	R/W
<9>	**TIMCMSOC_TIMG3_CH2_OS**: TIM_G2 output channel 2 complementary or single-ended output selection 1'b0: TIM_G2 complementary output (Ch2_n) selected 1'b1: TIM_G2 single-ended output (Ch6) selected	R/W
<8>	**TIMCMSOC_TIMG3_CH1_OS**: TIM_G2 output channel 1 complementary or single-ended output selection 1'b0: TIM_G2 complementary output (Ch1_n) selected 1'b1: TIM_G2 single-ended output (Ch5) selected	R/W
<7>	**TIMCMSOC_TIMG2_CH2_OS**: TIM_G1 output channel 2 complementary or single-ended output selection 1'b0: TIM_G1 complementary output (Ch2_n) selected 1'b1: TIM_G1 single-ended output (Ch6) selected	R/W
<6>	**TIMCMSOC_TIMG2_CH1_OS**: TIM_G1 output channel 1 complementary or single-ended output selection 1'b0: TIM_G1 complementary output (Ch1_n) selected 1'b1: TIM_G1 single-ended output (Ch5) selected	R/W
<5>	**TIMCMSOC_TIMG1_CH2_OS**: TIM_G0 output channel 2 complementary or single-ended output selection 1'b0: TIM_G0 complementary output (Ch2_n) selected 1'b1: TIM_G0 single-ended output (Ch6) selected	R/W
<4>	**TIMCMSOC_TIMG1_CH1_OS**: TIM_G0 output channel 1 complementary or single-ended output selection 1'b0: TIM_G0 complementary output (Ch1_n) selected 1'b1: TIM_G0 single-ended output (Ch5) selected	R/W
<3>	**TIMCMSOC_TIMA2_CH2_OS**: TIM_A1 output channel 2 complementary or	R/W

Bit	Description	Type
	single-ended output selection 1’b0: TIM_A1 complementary output (Ch2_n) selected 1’b1: TIM_A1 single-ended output (Ch6) selected	
<2>	**TIMCMSOC_TIMA2_CH1_OS**: TIM_A1 output channel 1 complementary or single-ended output selection 1’b0: TIM_A1 complementary output (Ch1_n) selected 1’b1: TIM_A1 single-ended output (Ch5) selected	
<1>	**TIMCMSOC_TIMA1_CH2_OS**: TIM_A0 output channel 2 complementary or single-ended output selection 1’b0: TIM_A0 complementary output (Ch2_n) selected 1’b1: TIM_A0 single-ended output (Ch6) selected	R/W
<0>	**TIMCMSOC_TIMA1_CH1_OS**: TIM_A0 output channel 1 complementary or single-ended output selection 1’b0: TIM_A0 complementary output (Ch1_n) selected 1’b1: TIM_A0 single-ended output (Ch5) selected	R/W

8.4.43 SYSCFG register map

The base address of the SYSCFG registers is 0x4000_0800.
The following table gives the SYSCFG register map and reset values.

Table 22 SYSCFG register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ROM_CFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	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	1	0	0	0	0
0x04	PVD_INT_CFG	Res	Res	Res	Res	PVD_INT_CFG_BP_FLAG	PVD_INT_CFG_INT_FLAG	PVD_INT_CFG_INT_POL	PVD_INT_CFG_INT_EN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	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
0x08		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	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
0x0C	SRAM_CTRL_1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0
0x1C	SRAM_CTRL_PD	Res	Res	Res	Res	Res	SRAM_SHUTDOWN_4	SRAM_DSLEEP_4	SRAM_LSLEEP_4	Res	Res	Res	Res	Res	SRAM_SHUTDOWN_3	SRAM_DSLEEP_3	SRAM_LSLEEP_3	Res	Res	Res	Res	Res	SRAM_SHUTDOWN_2	SRAM_DSLEEP_2	SRAM_LSLEEP_2	Res	Res	Res	Res	Res	SRAM_SHUTDOWN_1	SRAM_DSLEEP_1	SRAM_LSLEEP_1
	Reset value	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
0x24	SYS_CFG	SYS_CFG_BOOTCTRL	SYS_CFG_FLASHDMA		SYS_CFG_BOOT_MODE[1:0]		SYS_CFG_SPI2_RXDA[1:0]		SYS_CFG_SPI2_TXDA[1:0]		SYS_CFG_I2C1_RXDA[1:0]		SYS_CFG_I2C1_TXDA[1:0]		SYS_CFG_SPI1_RXDA[1:0]			SYS_CFG_SPI1_TXDA[1:0]		Res	Res	Res	SYS_CFG_UART4_MODE	SYS_CFG_UART3_MODE	SYS_CFG_UART2_MODE	SYS_CFG_UART1_MODE	Res	Res	Res	Res	SYS_CFG_HDIV_EN	Res	SYS_CFG_SPI2_MODE	SYS_CFG_SPI1_MODE
	Reset value	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
0x28	IRQ_CFG_1	IRQ_CFG_IRQ_31	IRQ_CFG_IRQ_30	IRQ_CFG_IRQ_29	IRQ_CFG_IRQ_28	IRQ_CFG_IRQ_27	IRQ_CFG_IRQ_26	IRQ_CFG_IRQ_25	IRQ_CFG_IRQ_24	IRQ_CFG_IRQ_23	IRQ_CFG_IRQ_22	IRQ_CFG_IRQ_21	IRQ_CFG_IRQ_20	IRQ_CFG_IRQ_19	IRQ_CFG_IRQ_18	IRQ_CFG_IRQ_17	IRQ_CFG_IRQ_16	IRQ_CFG_IRQ_15	IRQ_CFG_IRQ_14	IRQ_CFG_IRQ_13	IRQ_CFG_IRQ_12	IRQ_CFG_IRQ_11	IRQ_CFG_IRQ_10	IRQ_CFG_IRQ_09	IRQ_CFG_IRQ_08	IRQ_CFG_IRQ_07	IRQ_CFG_IRQ_06	IRQ_CFG_IRQ_05	IRQ_CFG_IRQ_04	IRQ_CFG_IRQ_03	IRQ_CFG_IRQ_02	IRQ_CFG_IRQ_01	IRQ_CFG_IRQ_00
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	IRQ_CFG_2	IRQ_CFG_IRQ_63	IRQ_CFG_IRQ_62	IRQ_CFG_IRQ_61	IRQ_CFG_IRQ_60	IRQ_CFG_IRQ_59	IRQ_CFG_IRQ_58	IRQ_CFG_IRQ_57	IRQ_CFG_IRQ_56	IRQ_CFG_IRQ_55	IRQ_CFG_IRQ_54	IRQ_CFG_IRQ_53	IRQ_CFG_IRQ_52	IRQ_CFG_IRQ_51	IRQ_CFG_IRQ_50	IRQ_CFG_IRQ_49	IRQ_CFG_IRQ_48	IRQ_CFG_IRQ_47	IRQ_CFG_IRQ_46	IRQ_CFG_IRQ_45	IRQ_CFG_IRQ_44	IRQ_CFG_IRQ_43	IRQ_CFG_IRQ_42	IRQ_CFG_IRQ_41	IRQ_CFG_IRQ_40	IRQ_CFG_IRQ_39	IRQ_CFG_IRQ_38	IRQ_CFG_IRQ_37	IRQ_CFG_IRQ_36	IRQ_CFG_IRQ_35	IRQ_CFG_IRQ_34	IRQ_CFG_IRQ_33	IRQ_CFG_IRQ_32
	Reset value	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
0x30	DMA_CFG		DMA_CFG_HWCH15[1:0]		DMA_CFG_HWCH14[1:0]		DMA_CFG_HWCH13[1:0]		DMA_CFG_HWCH12[1:0]		DMA_CFG_HWCH11[1:0]		DMA_CFG_HWCH10[1:0]		DMA_CFG_HWCH09[1:0]		DMA_CFG_HWCH08[1:0]		DMA_CFG_HWCH07[1:0]		DMA_CFG_HWCH06[1:0]		DMA_CFG_HWCH05[1:0]		DMA_CFG_HWCH04[1:0]		DMA_CFG_HWCH03[1:0]		DMA_CFG_HWCH02[1:0]		DMA_CFG_HWCH01[1:0]		DMA_CFG_HWCH00[1:0]
	Reset value	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
0x34	WDT_CFG	WDTW_CFG_RST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDTW_CFG_SPD	WDTW_CFG_EXTCLK
	Reset value	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
0x38	RESET_CFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	RST_CFG_TIMG4	RST_CFG_TIMG3	RST_CFG_TIMG2	RST_CFG_TIMA2	RST_CFG_TIMG1	RST_CFG_TIMA1	RST_CFG_GPIO	RST_CFG_SPI2	RST_CFG_SPI1	RST_CFG_TIMB24	RST_CFG_TIMB23	RST_CFG_TIMB22	RST_CFG_TIMB21	RST_CFG_TIMB14	RST_CFG_TIMB13	RST_CFG_TIMB12	RST_CFG_TIMB11	RST_CFG_I2C2	RST_CFG_I2C1	RST_CFG_UART4	RST_CFG_UART3	RST_CFG_UART2	RST_CFG_UART1
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x3C	DMA_CFG_2		DMA_CFG_2_HWCH15[1:0]		DMA_CFG_2_HWCH14[1:0]		DMA_CFG_2_HWCH13[1:0]		DMA_CFG_2_HWCH12[1:0]		DMA_CFG_2_HWCH11[1:0]		DMA_CFG_2_HWCH10[1:0]		DMA_CFG_2_HWCH09[1:0]		DMA_CFG_2_HWCH08[1:0]		DMA_CFG_2_HWCH07[1:0]		DMA_CFG_2_HWCH06[1:0]		DMA_CFG_2_HWCH05[1:0]		DMA_CFG_2_HWCH04[1:0]		DMA_CFG_2_HWCH03[1:0]		DMA_CFG_2_HWCH02[1:0]		DMA_CFG_2_HWCH01[1:0]		DMA_CFG_2_HWCH00[1:0]
	Reset value	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
0x40	TIMA_CFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	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
0x50	GPIO_ENAUX_A	GPIOAUX_A_31	GPIOAUX_A_30	GPIOAUX_A_29	GPIOAUX_A_28	GPIOAUX_A_27	GPIOAUX_A_26	GPIOAUX_A_25	GPIOAUX_A_24	GPIOAUX_A_23	GPIOAUX_A_22	GPIOAUX_A_21	GPIOAUX_A_20	GPIOAUX_A_19	GPIOAUX_A_18	GPIOAUX_A_17	GPIOAUX_A_16	GPIOAUX_A_15	GPIOAUX_A_14	GPIOAUX_A_13	GPIOAUX_A_12	GPIOAUX_A_11	GPIOAUX_A_10	GPIOAUX_A_09	GPIOAUX_A_08	GPIOAUX_A_07	GPIOAUX_A_06	GPIOAUX_A_05	GPIOAUX_A_04	GPIOAUX_A_03	GPIOAUX_A_02	GPIOAUX_A_01	GPIOAUX_A_00
	Reset value	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
0x54	GPIO_ENAUX_B	GPIOAUX_B_31	GPIOAUX_B_30	GPIOAUX_B_29	GPIOAUX_B_28	GPIOAUX_B_27	GPIOAUX_B_26	GPIOAUX_B_25	GPIOAUX_B_24	GPIOAUX_B_23	GPIOAUX_B_22	GPIOAUX_B_21	GPIOAUX_B_20	GPIOAUX_B_19	GPIOAUX_B_18	GPIOAUX_B_17	GPIOAUX_B_16	GPIOAUX_B_15	GPIOAUX_B_14	GPIOAUX_B_13	GPIOAUX_B_12	GPIOAUX_B_11	GPIOAUX_B_10	GPIOAUX_B_09	GPIOAUX_B_08	GPIOAUX_B_07	GPIOAUX_B_06	GPIOAUX_B_05	GPIOAUX_B_04	GPIOAUX_B_03	GPIOAUX_B_02	GPIOAUX_B_01	GPIOAUX_B_00
	Reset value	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
0x58	GPIO_ENAUX_C	GPIOAUX_C_31	GPIOAUX_C_30	GPIOAUX_C_29	GPIOAUX_C_28	GPIOAUX_C_27	GPIOAUX_C_26	GPIOAUX_C_25	GPIOAUX_C_24	GPIOAUX_C_23	GPIOAUX_C_22	GPIOAUX_C_21	GPIOAUX_C_20	GPIOAUX_C_19	GPIOAUX_C_18	GPIOAUX_C_17	GPIOAUX_C_16	GPIOAUX_C_15	GPIOAUX_C_14	GPIOAUX_C_13	GPIOAUX_C_12	GPIOAUX_C_11	GPIOAUX_C_10	GPIOAUX_C_09	GPIOAUX_C_08	GPIOAUX_C_07	GPIOAUX_C_06	GPIOAUX_C_05	GPIOAUX_C_04	GPIOAUX_C_03	GPIOAUX_C_02	GPIOAUX_C_01	GPIOAUX_C_00
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	GPIO_ENAUX_D	GPIOAUX_D_31	GPIOAUX_D_30	GPIOAUX_D_29	GPIOAUX_D_28	GPIOAUX_D_27	GPIOAUX_D_26	GPIOAUX_D_25	GPIOAUX_D_24	GPIOAUX_D_23	GPIOAUX_D_22	GPIOAUX_D_21	GPIOAUX_D_20	GPIOAUX_D_19	GPIOAUX_D_18	GPIOAUX_D_17	GPIOAUX_D_16	GPIOAUX_D_15	GPIOAUX_D_14	GPIOAUX_D_13	GPIOAUX_D_12	GPIOAUX_D_11	GPIOAUX_D_10	GPIOAUX_D_09	GPIOAUX_D_08	GPIOAUX_D_07	GPIOAUX_D_06	GPIOAUX_D_05	GPIOAUX_D_04	GPIOAUX_D_03	GPIOAUX_D_02	GPIOAUX_D_01	GPIOAUX_D_00
	Reset value	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
0xF4	UID_REG_1	UID_REG_1_UID[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0xF8	UID_REG_2	UID_REG_2_UID[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0xFC	UID_REG_3	UID_REG_3_UID[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x100	UART_ADCFG	Res	Res	Res	Res	Res	UART4_ADCFG_CLR	UART4_ADCFG_IE	UART4_ADCFG_EN	Res	Res	Res	Res	Res	UART3_ADCFG_CLR	UART3_ADCFG_IE	UART3_ADCFG_EN	Res	Res	Res	Res	Res	UART2_ADCFG_CLR	UART2_ADCFG_IE	UART2_ADCFG_EN	Res	Res	Res	Res	Res	UART1_ADCFG_CLR	UART1_ADCFG_IE	UART1_ADCFG_EN
	Reset value	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
0x104	UART_ADIF	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UART4_ADFLG	UART3_ADFLG	UART2_ADFLG	UART1_ADFLG
	Reset value	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
0x108	UART1_ADBR	UART1_ADBR[31:0]																															
	Reset value	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
0x10C	UART2_ADBR	UART2_ADBR[31:0]																															
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x110	UART3_ADBR	UART3_ADBR[31:0]																															
	Reset value	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
0x114	UART4_ADBR	UART4_ADBR[31:0]																															
	Reset value	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
0x120	CTSM_CTRL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CTSM_CTRL_CLRINT	Res	Res	Res	CTSM_CTRL_EN
	Reset value	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
0x124	CTSM_INT	CTSM_INT_INTGEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CTSM_INT_SOINIT	CTSM_INT_MSCAN	CTSM_INT_AFC	CTSM_INT_GETST	CTSM_INT_JINIT	Res	Res	Res	Res	Res	CTSM_INT_IDCSTRT	CTSM_INT_TCSTRT	CTSM_INT_DIASTRT	CTSM_INT_DCSTRT	CTSM_INT_CSTRT	CTSM_INT_MSTRT	CTSM_INT_MSTOP
	Reset value	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
0x128	CTSM_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CTSM_SR_ERROR	CTSM_SR_JCPLT	CTSM_SR_SCPLT	CTSM_SR_MCPLT	CTSM_SR_BUSY	CTSM_SR_JRRD	CTSM_SR_MARRD	CTSM_SR_MRRD
	Reset value	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
0x130	HDIV_A	HDIV_A[31:0]																															
	Reset value	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	1
0x134	HDIV_B	HDIV_B[31:0]																															
	Reset value	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	1
0x138	HDIV_Q	HDIV_Q[31:0]																															
	Reset value	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
0x13C	HDIV_R	HDIV_R[31:0]																															

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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
0x140	AFC_CTRL	Res	Res														AFCCTRL_RFCNT[13:0]								AFCCTRL_DIFFCNT[7:0]			AFCCTRL_CBCLKS	AFCCTRL_AEIE	AFCCTRL_ANIE	AFCCTRL_ALIE	AFCCTRL_AFCBYS	AFCCTRL_AFCEN
	Reset value	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
0x144	AFC_CFG	Res	Res	Res	Res																				AFCCFG_STDCNT[19:0]								AFCCFG_CBADJ[7:0]
	Reset value	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	1	0	0	0	0
0x148	AFC_RES																AFCRES_RFDIV[15:0]	Res	Res	Res	Res	Res	AFCRES_AEIF	AFCRES_ANIF	AFCRES_ALIF								AFCRES_ADJRES[7:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x180	TIM_A1_IMX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			TIMA1IMX_ETR[2:0]	Res			TIMA1IMX_CH_4[2:0]	Res			TIMA1IMX_CH_3[2:0]	Res			TIMA1IMX_CH_2[2:0]	Res			TIMA1IMX_CH_1[2:0]
	Reset value	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
0x184	TIM_A2_IMX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			TIMA2IMX_ETR[2:0]	Res			TIMA2IMX_CH_4[2:0]	Res			TIMA2IMX_CH_3[2:0]	Res			TIMA2IMX_CH_2[2:0]	Res			TIMA2IMX_CH_1[2:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0x188	TIM_G1_IMX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			TIMG1IMX_ETR[2:0]	Res			TIMG1IMX_CH_4[2:0]	Res			TIMG1IMX_CH_3[2:0]	Res			TIMG1IMX_CH_2[2:0]	Res			TIMG1IMX_CH_1[2:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0x18C	TIM_G2_IMX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			TIMG2IMX_ETR[2:0]	Res			TIMG2IMX_CH_4[2:0]	Res			TIMG2IMX_CH_3[2:0]	Res			TIMG2IMX_CH_2[2:0]	Res			TIMG2IMX_CH_1[2:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0x190	TIM_G3_IMX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res			TIMG3IMX_ETR[2:0]	Res			TIMG3IMX_CH_4[2:0]	Res			TIMG3IMX_CH_3[2:0]	Res			TIMG31IMX_CH_2[2:0]	Res			TIMG3IMX_CH_1[2:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x194	TIM_G4_IMX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIMG4IMX_ETR[2:0]			Res	TIMG4IMX_CH_4[2:0]			Res	TIMG4IMX_CH_3[2:0]			Res	TIMG4IMX_CH_2[2:0]			Res	TIMG4IMX_CH_1[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0x1A0	TIM_CMS_OC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIMG4_CH2_OS	TIMG4_CH1_OS	TIMG3_CH2_OS	TIMG3_CH1_OS	TIMG2_CH2_OS	TIMG2_CH1_OS	TIMG1_CH2_OS	TIMG1_CH1_OS	TIMA2_CH2_OS	TIMA2_CH1_OS	TIMA1_CH2_OS	TIMA1_CH1_OS
	Reset value	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

9 Port mux (PADI)

9.1 Introduction

PADI assign function to 62 IOs.
The relation of IOs and pins are shown in the following tables.

Table 23 XT32H05x IO assignment (64 Pins)

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
61	1	CH4P	PB28	PC4		GTO4_CH1P						CH20
62	2	CH4N	PB29	PC5		GTO4_CH1N						CH19
1	3		PC20	PD0	ATIN1_ETR		UART1_TX	ATIN1_BRK1		PB30	LED0	CH11
2	4		PD9	PC3	ATIN2_CH3				ATO2_CH3P	PB31		CH36
3	5		PD7	PC2		BTOUT21	UART1_CTS	BTIN21	ATO2_CH3N			
4	6	CTSU30	PD6	PC1	GTIN4_CH1	BTOUT24	UART1_RTS	BTIN24	GTO4_CH1P			
5	7	CTSU29	PD5	PC0	ATIN1_BRK2		UART1_RX		ATO1_CH3P		LED1	
6	8		RSTb									
7	9	XO_HS	PC21	PD4			EXTCLK_HS					
8	10	XI_HS	PC22	PD3								
9	11	CTSU28	PA0	PD15	ATIN1_CH3	ATO1_CH3N					LED2	
10	12	CTSU27	PC23	PB26	NMI	ATO1_CH1P	SPI2_CSB				LED3	
11	13	XO_LS	PD10	PD2			EXTCLK_LS			PC14		
12	14	XI_LS	PD11	PD1						PC15		
PS	15											
PS	16											
13	17	CTSU26	PD12	PC16	ATIN1_CH1	ATO1_CH1N	SPI2_SCLK	I2C1_SCK	UART2_TX		LED4	CH35
14	18	CTSU25	PD13	PC17		ATO1_CH2P	SPI2_MOSO	I2C1_SDA	UART2_RX		LED5	CH34
15	19	CTSU24	PD14	PC18	ATIN1_CH2	ATO1_CH2N	SPI2_MISI		UART2_CTS		LED6	CH33
16	20	CTSU23	PB26	PC19		ATO2_BRK1			UART2_RTS			CH32
17	21	CTSU22	PA4	PB25	BTIN14	BTOUT14	SWDIO				LED7	CH44
18	22		PA11	PC31	GTIN2_CH2	GTO2_CH2P	UART2_TX		ATO2_CH1P			CH37
19	23		PA10	PC30	GTIN2_ETR	GTO2_CH3P	UART2_RX		ATO2_CH1N	GTO2_CH2N		CH38
20	24	CTSU21	PA9	PC29	GTIN3_CH2	GTO3_CH2P	I2C2_SCK		ATO2_CH2P	PD31		CH12
21	25	CTSU20	PA8	PC28	GTIN4_CH2	GTO4_CH2P	I2C2_SDA		ATO2_CH2N	PD30		CH10

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
22	26	CTSU19	PD29	PC27	GTIN2_CH1	GTO2_CH1P				SPI1_CSB		CH29
23	27	CTSU18	PD28	PC26		GTO2_CH1N			ATO1_BRK2	SPI1_MOSO		CH30
24	28	CTSU17	PD27	PC25	GTIN2_CH4	GTO2_CH4			ATO1_BRK1	SPI1_MISI		CH31
25	29	CTSU16	PD26	PC24			SWCLK		PB7	SPI1_SCLK	LED8	CH39
26	30	CH3P	PD25	PB6	BTIN23	BTOUT23	SWDIO					CH18
27	31	CH3N	PD24	PB5	BTIN22	BTOUT22	SWCLK	ATO2_BRK2	PC6			CH17
28	32	CTSU15	PA3	PB24	ATIN1_ETR	ATO1_CH3P	I2C1_SCK		PC7	SPI1_MISI		CMP1P
29	33	CTSU14	PA1	PC8	GTIN1_ETR	ATO1_CH3N	I2C1_SDA	SWDIO		SPI1_MOSO		CMP1N
30	34	**DAC1OUT**	PA2	PC9	ATIN1_BRK1	GTO1_CH1N	OCMP1	SWCLK				CH7
31	35	CH2P	PA10	PC10								CH16
32	36	CH2N	PA11	PC11								CH15
33	37	CH1P	PB27	PC12								CH14
34	38	CH1N	PD21	PC13			SPI1_SCLK					CH13
35	39	CTSU13	PD20	PB15	ATIN1_CH2	ATO1_CH2P	UART3_TX					CMP2P
36	40	CTSU12	PA5	PB14	BTIN12	ATO1_CH2N	UART3_RX			SPI2_CSB		CMP2N
37	41		PA31	PB13	BTIN13	BTOUT13	OCMP2	GTIN1_ETR	I2C2_SCK	SWCLK		CH6
38	42	CTSU11	PD19	PB12	BTIN14	ATO1_CH1P	UART2_RX	I2C1_SCK	I2C2_SDA	SPI2_MISI		CMP3P
39	43	CTSU10	PD18	PB11	BTIN21	ATO1_CH1N	UART2_TX	I2C1_SDA		SPI2_MOSO		CMP3N
40	44	**DAC2OUT**	PA30	PB10	BTIN22	BTOUT22	SPI1_MOSO	OCMP3	UART3_TX	SWDIO		CH5
41	45	CH4N, CMP4N, CTSU9	PA29	PB9	BTIN11	BTOUT11	SPI1_MISI	I2C2_SDA	UART3_RX	SPI2_SCLK		CH28
42	46	CH4P, CMP4P	PA28	PB8	BTIN12	BTOUT12	SPI1_SCLK	I2C2_SCK				CH4
43	47	CTSU8	PA27	PD22	GTIN1_CH1	GTO1_CH1P						CH40
44	48	CTSU_CAP	PA26	PD23	ATIN1_CH4	ATO1_CH4	SPI1_CSB	OCMP4				CH41
45	49	CH1P (CH21)	PA25	PB23					GTO1_CH4			CTSU7
46	50	CH2P (CH22)	PA24	PB22					GTO2_CH2P			CTSU6
47	51	CH3P (CH23)	PA23	PB21					GTO3_CH1P			CTSU5
48	52	CH4P (CH24)	PA22	PB20					GTO3_CH1N			CTSU4
49	53	CH2N, CTSU1	PA21	PB19		ATO1_CH3P	I2C1_SCK					CH26
50	54	CH2P	PA20	PB18		ATO1_CH3N	I2C1_SDA					CH2
51	55	CH1N, CTSU0	PA19	PB17		ATO1_CH2P						CH25
52	56	CH1P	PA18	PB16		ATO1_CH2N						CH1
53	57	CTSU3	PA17	PD8	GTIN1_CH2	GTO1_CH2P		SPI1_CSB		GTO2_CH2P		CH9
54	58		PA16	PB4	ATIN2_CH4	ATO2_CH4	I2C1_SCK	SPI1_SCK				OCMP2
55	59		PA15	PB3	ATIN2_BRK1	GTO2_CH2N	I2C1_SDA	SPI1_MISI	UART4_RX			O_PGA2

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
56	60		PA14	PB2	GTIN2_ETR	GTO2_CH3N		SPI1_MOSO	UART4_TX			CH42
57	61	CH3N, CTSU31	PA13	PB1	ATIN2_BRK2	ATO1_CH1P	UART4_RX					CH27
58	62	CH3P	PA12	PB0		ATO1_CH1N	UART4_TX					CH3
59	63		PA7	PD17	ATIN2_ETR							CH43
60	64	CTSU2	PA6	PD16	GTIN1_CH3	GTO1_CH3P				GTO2_CH2N		CH8

Table 24 XT32H0xxExxx IO assignment (52 Pins)

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
1	2		PC20	PD0	ATIN1_ETR		UART1_TX	ATIN1_BRK1		PB30	LED0	CH11
4	3	CTSU30	PD6	PC1	GTIN4_CH1	BTOUT24	UART1_RTS	BTIN24	GTO4_CH1P			
5	4	CTSU29	PD5	PC0	ATIN1_BRK2		UART1_RX		ATO1_CH3P		LED1	
6	5		RSTb									
7	6	XO_HS	PC21	PD4			EXTCLK_HS					
8	7	XI_HS	PC22	PD3								
9	8	CTSU28	PA0	PD15	ATIN1_CH3	ATO1_CH3N					LED2	
10	9	CTSU27	PC23	PB26	NMI	ATO1_CH1P	SPI2_CSB				LED3	
11	10	XO_LS	PD10	PD2			EXTCLK_LS			PC14		
12	11	XI_LS	PD11	PD1						PC15		
PS	12											
PS	13											
13	14	CTSU26	PD12	PC16	ATIN1_CH1	ATO1_CH1N	SPI2_SCLK	I2C1_SCK	UART2_TX		LED4	CH35
14	15	CTSU25	PD13	PC17		ATO1_CH2P	SPI2_MOSO	I2C1_SDA	UART2_RX		LED5	CH34
15	16	CTSU24	PD14	PC18	ATIN1_CH2	ATO1_CH2N	SPI2_MISI		UART2_CTS		LED6	CH33
16	17	CTSU23	PB26	PC19		ATO2_BRK1			UART2_RTS			CH32
17	18	CTSU22	PA4	PB25	BTIN14	BTOUT14	SWDIO				LED7	CH44
18	19		PA11	PC31	GTIN2_CH2	GTO2_CH2P	UART2_TX		ATO2_CH1P			CH37
19	20		PA10	PC30	GTIN2_ETR	GTO2_CH3P	UART2_RX		ATO2_CH1N	GTO2_CH2N		CH38
20	21	CTSU21	PA9	PC29	GTIN3_CH2	GTO3_CH2P	I2C2_SCK		ATO2_CH2P	PD31		CH12
21	22	CTSU20	PA8	PC28	GTIN4_CH2	GTO4_CH2P	I2C2_SDA		ATO2_CH2N	PD30		CH10
22	23	CTSU19	PD29	PC27	GTIN2_CH1	GTO2_CH1P				SPI1_CSB		CH29
23	24	CTSU18	PD28	PC26		GTO2_CH1N			ATO1_BRK2	SPI1_MOSO		CH30
24	25	CTSU17	PD27	PC25	GTIN2_CH4	GTO2_CH4			ATO1_BRK1	SPI1_MISI		CH31
25	26	CTSU16	PD26	PC24			SWCLK		PB7	SPI1_SCLK	LED8	CH39
28	27	CTSU15	PA3	PB24	ATIN1_ETR	ATO1_CH3P	I2C1_SCK	—	PC7	SPI1_MISI		CMP1P
29	28	CTSU14	PA1	PC8	GTIN1_ETR	ATO1_CH3N	I2C1_SDA	SWDIO		SPI1_MOSO		CMP1N
30	29	**DAC1OUT**	PA2	PC9	ATIN1_BRK1	GTO1_CH1N	OCMP1	SWCLK				CH7
35	30	CTSU13	PD20	PB15	ATIN1_CH2	ATO1_CH2P	UART3_TX					CMP2P
36	31	CTSU12	PA5	PB14	BTIN12	ATO1_CH2N	UART3_RX			SPI2_CSB		CMP2N
37	32		PA31	PB13	BTIN13	BTOUT13	OCMP2	GTIN1_ETR	I2C2_SCK	SWCLK		CH6
38	33	CTSU11	PD19	PB12	BTIN14	ATO1_CH1P	UART2_RX	I2C1_SCK	I2C2_SDA	SPI2_MISI		CMP3P
39	34	CTSU10	PD18	PB11	BTIN21	ATO1_CH1N	UART2_TX	I2C1_SDA		SPI2_MOSO		CMP3N

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
40	35	DAC2OUT	PA30	PB10	BTIN22	BTOUT22	SPI1_MOSO	OCMP3	UART3_TX	SWDIO		CH5
41	36	CH4N, CMP4N, CTSU9	PA29	PB9	BTIN11	BTOUT11	SPI1_MISI	I2C2_SDA	UART3_RX	SPI2_SCLK		CH28
42	37	CH4P, CMP4P	PA28	PB8	BTIN12	BTOUT12	SPI1_SCLK	I2C2_SCK				CH4
43	38	CTSU8	PA27	PD22	GTIN1_CH1	GTO1_CH1P						CH40
44	39	CTSU_CAP	PA26	PD23	ATIN1_CH4	ATO1_CH4	SPI1_CSB	OCMP4				CH41
45	40	CH1P (CH21)	PA25	PB23					GTO1_CH4			CTSU7
46	41	CH2P (CH22)	PA24	PB22					GTO2_CH2P			CTSU6
47	42	CH3P (CH23)	PA23	PB21					GTO3_CH1P			CTSU5
48	43	CH4P (CH24)	PA22	PB20					GTO3_CH1N			CTSU4
49	44	CH2N, CTSU1	PA21	PB19		ATO1_CH3P	I2C1_SCK					CH26
50	45	CH2P	PA20	PB18		ATO1_CH3N	I2C1_SDA					CH2
51	46	CH1N, CTSU0	PA19	PB17		ATO1_CH2P						CH25
52	47	CH1P	PA18	PB16		ATO1_CH2N						CH1
53	48	CTSU3	PA17	PD8	GTIN1_CH2	GTO1_CH2P		SPI1_CSB		GTO2_CH2P		CH9
55	49		PA15	PB3	ATIN2_BRK1	GTO2_CH2N	I2C1_SDA	SPI1_MISI	UART4_RX			O_PGA2
56	50		PA14	PB2	GTIN2_ETR	GTO2_CH3N		SPI1_MOSO	UART4_TX			CH42
57	51	CH3N, CTSU31	PA13	PB1	ATIN2_BRK2	ATO1_CH1P	UART4_RX					CH27
58	52	CH3P	PA12	PB0		ATO1_CH1N	UART4_TX					CH3
60	1	CTSU2	PA6	PD16	GTIN1_CH3	GTO1_CH3P				GTO2_CH2N		CH8

Table 25 XT32H0xxDxxx IO assignment (48 Pins)

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
1	37		PC20	PD0	ATIN1_ETR		UART1_TX	ATIN1_BRK1		PB30	LED0	CH11
4	38	CTSU30	PD6	PC1	GTIN4_CH1	BTOUT24	UART1_RTS	BTIN24	GTO4_CH1P			
5	39	CTSU29	PD5	PC0	ATIN1_BRK2		UART1_RX		ATO1_CH3P		LED1	
6	40		RSTb									
7	41	XO_HS	PC21	PD4			EXTCLK_HS					
8	42	XI_HS	PC22	PD3								
9	43	CTSU28	PA0	PD15	ATIN1_CH3	ATO1_CH3N					LED2	
10	44	CTSU27	PC23	PB26	NMI	ATO1_CH1P	SPI2_CSB				LED3	
11	45	XO_LS	PD10	PD2			EXTCLK_LS			PC14		
12	46	XI_LS	PD11	PD1						PC15		
PS	47											
PS	48											
13	1	CTSU26	PD12	PC16	ATIN1_CH1	ATO1_CH1N	SPI2_SCLK	I2C1_SCK	UART2_TX		LED4	CH35
14	2	CTSU25	PD13	PC17		ATO1_CH2P	SPI2_MOSO	I2C1_SDA	UART2_RX		LED5	CH34
15	3	CTSU24	PD14	PC18	ATIN1_CH2	ATO1_CH2N	SPI2_MISI		UART2_CTS		LED6	CH33
16	4	CTSU23	PB26	PC19		ATO2_BRK1			UART2_RTS			CH32
17	5	CTSU22	PA4	PB25	BTIN14	BTOUT14	SWDIO				LED7	CH44
20	6	CTSU21	PA9	PC29	GTIN3_CH2	GTO3_CH2P	I2C2_SCK		ATO2_CH2P	PD31		CH12
21	7	CTSU20	PA8	PC28	GTIN4_CH2	GTO4_CH2P	I2C2_SDA		ATO2_CH2N	PD30		CH10
22	8	CTSU19	PD29	PC27	GTIN2_CH1	GTO2_CH1P				SPI1_CSB		CH29
23	9	CTSU18	PD28	PC26		GTO2_CH1N			ATO1_BRK2	SPI1_MOSO		CH30
24	10	CTSU17	PD27	PC25	GTIN2_CH4	GTO2_CH4			ATO1_BRK1	SPI1_MISI		CH31
25	11	CTSU16	PD26	PC24			SWCLK		PB7	SPI1_SCLK	LED8	CH39
28	12	CTSU15	PA3	PB24	ATIN1_ETR	ATO1_CH3P	I2C1_SCK	—	PC7	SPI1_MISI		CMP1P
29	13	CTSU14	PA1	PC8	GTIN1_ETR	ATO1_CH3N	I2C1_SDA	SWDIO		SPI1_MOSO		CMP1N
30	14	**DAC1OUT**	PA2	PC9	ATIN1_BRK1	GTO1_CH1N	OCMP1	SWCLK				CH7
35	15	CTSU13	PD20	PB15	ATIN1_CH2	ATO1_CH2P	UART3_TX					CMP2P
36	16	CTSU12	PA5	PB14	BTIN12	ATO1_CH2N	UART3_RX			SPI2_CSB		CMP2N
37	17		PA31	PB13	BTIN13	BTOUT13	OCMP2	GTIN1_ETR	I2C2_SCK	SWCLK		CH6
38	18	CTSU11	PD19	PB12	BTIN14	ATO1_CH1P	UART2_RX	I2C1_SCK	I2C2_SDA	SPI2_MISI		CMP3P
39	19	CTSU10	PD18	PB11	BTIN21	ATO1_CH1N	UART2_TX	I2C1_SDA		SPI2_MOSO		CMP3N
40	20	**DAC2OUT**	PA30	PB10	BTIN22	BTOUT22	SPI1_MOSO	OCMP3	UART3_TX	SWDIO		CH5
41	21	CH4N, CMP4N, CTSU9	PA29	PB9	BTIN11	BTOUT11	SPI1_MISI	I2C2_SDA	UART3_RX	SPI2_SCLK		CH28

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
42	22	CH4P, CMP4P	PA28	PB8	BTIN12	BTOUT12	SPI1_SCLK	I2C2_SCK				CH4
43	23	CTSU8	PA27	PD22	GTIN1_CH1	GTO1_CH1P						CH40
44	24	CTSU_CAP	PA26	PD23	ATIN1_CH4	ATO1_CH4	SPI1_CSB	OCMP4				CH41
45	25	CH1P (CH21)	PA25	PB23					GTO1_CH4			CTSU7
46	26	CH2P (CH22)	PA24	PB22					GTO2_CH2P			CTSU6
47	26	CH3P (CH23)	PA23	PB21					GTO3_CH1P			CTSU5
48	28	CH4P (CH24)	PA22	PB20					GTO3_CH1N			CTSU4
49	29	CH2N, CTSU1	PA21	PB19		ATO1_CH3P	I2C1_SCK					CH26
50	30	CH2P	PA20	PB18		ATO1_CH3N	I2C1_SDA					CH2
51	31	CH1N, CTSU0	PA19	PB17		ATO1_CH2P						CH25
52	32	CH1P	PA18	PB16		ATO1_CH2N						CH1
53	33	CTSU3	PA17	PD8	GTIN1_CH2	GTO1_CH2P		SPI1_CSB		GTO2_CH2P		CH9
57	34	CH3N, CTSU31	PA13	PB1	ATIN2_BRK2	ATO1_CH1P	UART4_RX					CH27
58	35	CH3P	PA12	PB0		ATO1_CH1N	UART4_TX					CH3
60	36	CTSU2	PA6	PD16	GTIN1_CH3	GTO1_CH3P				GTO2_CH2N		CH8

Table XT32H0xxBxxx IO assignment (32 Pins)

PAD #	Pin #	CFG_0	CFG1	CFG2_GPIO	CFG2_AUXI	CFG2_AUXO	CFG3	CFG4	CFG5	CFG6	CFG_7	CFG_8
1	32		PC20	PD0	ATIN1_ETR		UART1_TX	ATIN1_BRK1		PB30	LED0	CH11
5	1	CTSU29	PD5	PC0	ATIN1_BRK2		UART1_RX		ATO1_CH3P		LED1	
6	2		RSTb									
9	3	CTSU28	PA0	PD15	ATIN1_CH3	ATO1_CH3N					LED2	
10	4	CTSU27	PC23	PB26	NMI	ATO1_CH1P	SPI2_CSB				LED3	
11	5	XO_LS	PD10	PD2			EXTCLK_LS			PC14		
12	6	XI_LS	PD11	PD1						PC15		
PS	7											
PS	8											
13	9	CTSU26	PD12	PC16	ATIN1_CH1	ATO1_CH1N	SPI2_SCLK	I2C1_SCK	UART2_TX		LED4	CH35
14	10	CTSU25	PD13	PC17		ATO1_CH2P	SPI2_MOSO	I2C1_SDA	UART2_RX		LED5	CH34
15	11	CTSU24	PD14	PC18	ATIN1_CH2	ATO1_CH2N	SPI2_MISI		UART2_CTS		LED6	CH33
17	12	CTSU22	PA4	PB25	BTIN14	BTOUT14	SWDIO				LED7	CH44
25	13	CTSU16	PD26	PC24			SWCLK		PB7	SPI1_SCLK	LED8	CH39
28	14	CTSU15	PA3	PB24	ATIN1_ETR	ATO1_CH3P	I2C1_SCK	—	PC7	SPI1_MISI		CMP1P
29	15	CTSU14	PA1	PC8	GTIN1_ETR	ATO1_CH3N	I2C1_SDA	SWDIO		SPI1_MOSO		CMP1N
30	16	**DAC1OUT**	PA2	PC9	ATIN1_BRK1	GTO1_CH1N	OCMP1	SWCLK				CH7
35	17	CTSU13	PD20	PB15	ATIN1_CH2	ATO1_CH2P	UART3_TX					CMP2P
36	18	CTSU12	PA5	PB14	BTIN12	ATO1_CH2N	UART3_RX			SPI2_CSB		CMP2N
37	19		PA31	PB13	BTIN13	BTOUT13	OCMP2	GTIN1_ETR	I2C2_SCK	SWCLK		CH6
38	20	CTSU11	PD19	PB12	BTIN14	ATO1_CH1P	UART2_RX	I2C1_SCK	I2C2_SDA	SPI2_MISI		CMP3P
39	21	CTSU10	PD18	PB11	BTIN21	ATO1_CH1N	UART2_TX	I2C1_SDA		SPI2_MOSO		CMP3N
40	22	**DAC2OUT**	PA30	PB10	BTIN22	BTOUT22	SPI1_MOSO	OCMP3	UART3_TX	SWDIO		CH5
41	23	CH4N, CMP4N, CTSU9	PA29	PB9	BTIN11	BTOUT11	SPI1_MISI	I2C2_SDA	UART3_RX	SPI2_SCLK		CH28
42	24	CH4P, CMP4P	PA28	PB8	BTIN12	BTOUT12	SPI1_SCLK	I2C2_SCK				CH4
44	25	CTSU_CAP	PA26	PD23	ATIN1_CH4	ATO1_CH4	SPI1_CSB	OCMP4				CH41
49	26	CH2N, CTSU1	PA21	PB19		ATO1_CH3P	I2C1_SCK					CH26
50	27	CH2P	PA20	PB18		ATO1_CH3N	I2C1_SDA					CH2
51	28	CH1N, CTSU0	PA19	PB17		ATO1_CH2P						CH25
52	29	CH1P	PA18	PB16		ATO1_CH2N						CH1
57	30	CH3N, CTSU31	PA13	PB1	ATIN2_BRK2	ATO1_CH1P	UART4_RX					CH27
58	31	CH3P	PA12	PB0		ATO1_CH1N	UART4_TX					CH3

9.2 PADI main features

- Assignment functions to IO
- configure pad pullup or pulldown and driver strength

9.3 PADI functional description

9.3.1 Configure IO

The XT32H05x microcontroller has a maximum of 62 GPIOs. Each GPIO is assigned an identity ranging from 1 to 62, while the corresponding index for each IO ranges from 0 to 61. This means that IO1 corresponds to index 0, and IO62 corresponds to index 61.

Table 26 Configure IO functions

IOx_CFG[3:0]	EXTCLK enable	GPIOx_AUX	Description
0000	x	x	CFG0
0001	x	x	CFG1
0010	0	0	CFG2_GPIO
0010	1	0	CFG2_AUXI
0010	x	1	CFG2_AUX2
0011	x	x	CFG3
0100	x	x	CFG4
0101	x	x	CFG5
0110	x	x	CFG6
0111	x	x	CFG7
1000	x	x	CFG8

Configuration IO performs the following steps:

1. Write PORTMUX_CFG_x to set function;
2. Based on function, write IO_PULLUP_CFG_x and IO_PULLDW_CFG_x to set pullup or pulldown for pad;
3. Based on function, write IO_DS_CFG_x to set drive strength for pad.

9.4 PADI registers

9.4.1 PADI_PORTMUX_CFG_1

Base address: 0x5000_0800
Address offset: 0x00
Type: Read/Write
Reset value: 0x7713_7773

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=1 to 8) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.2 PADI_PORTMUX_CFG_2

Base address: 0x5000_0800
Address offset: 0x04
Type: Read/Write
Reset value: 0x7777_7777

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=9 to 16) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.3 PADI_PORTMUX_CFG_3

Base address: 0x5000_0800
Address offset: 0x08
Type: Read/Write
Reset value: 0x7777_7777

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=17 to 24) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.4 PADI_PORTMUX_CFG_4

Base address: 0x5000_0800
Address offset: 0x0C
Type: Read/Write
Reset value: 0x7777_7777

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=25 to 32) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG4, 4'b0100: CFG3, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.5 PADI_PORTMUX_CFG_5

Base address: 0x5000_0800
Address offset: 0x10
Type: Read/Write
Reset value: 0x6776_7777

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG [3:0]**: IO x (x=33 to 40) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.6 PADI_PORTMUX_CFG_6

Base address: 0x5000_0800
Address offset: 0x14
Type: Read/Write
Reset value: 0x7777_7777

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=41 to 48) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.7 PADI_PORTMUX_CFG_7

Base address: 0x5000_0800
Address offset: 0x18
Type: Read/Write
Reset value: 0x7777_7777

Bit	Description	Type
<31:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=49 to 56) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.8 PADI_PORTMUX_CFG_8

Base address: 0x5000_0800
Address offset: 0x1C
Type: Read/Write
Reset value: 0x0077_7777

Bit	Description	Type
<31:16>	Reserved	R/W
<15:0>	**IOCFG_IO_x_CFG[3:0]**: IO x (x=57 to 62) Function Configuration 4'b0000: CFG0, 4'b0001: CFG1, 4'b0010: CFG2, 4'b0011: CFG3, 4'b0100: CFG4, 4'b0101: CFG5, 4'b0110: CFG6, 4'b0111: CFG7, 4'b1000: CFG8, Others, reserved	R/W

9.4.9 PADI_IO_PULLUP_CFG_1

Base address: 0x5000_0800
Address offset: 0x20
Type: Read/Write
Reset value: 0x0000_0030

Bit	Description	Type
<31:0>	**IO_PULLUP_IO_x**: IO x (x=1 to 32) Pull-up Enable, 1'b0: disable, 1'b1: enable	R/W

9.4.10 PADI_IO_PULLUP_CFG_2

Base address: 0x5000_0800
Address offset: 0x24
Type: Read/Write
Reset value: 0x0000_0080

Bit	Description	Type
<31:28>	Reserved	R/W
<27:0>	**IO_PULLUP_IO_x**: IO x (x=33 to 62) Pull-up Enable, 1'b0: disable, 1'b1: enable	R/W

9.4.11 PADI_IO_PULLDOWN_CFG_1

Base address: 0x5000_0800
Address offset: 0x28
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**IO_PULLDW_IO_x**: IO x (x=1 to 32) Pull-down Enable, 1'b0: disable, 1'b1: enable	R/W

9.4.12 PADI_IO_PULLDOWN_CFG_2

Base address: 0x5000_0800
Address offset: 0x2C
Type: Read/Write
Reset value: 0x0000_0110

Bit	Description	Type
<31:30>	Reserved	R/W
<29:0>	**IO_PULLDW_IO_x**: IO x (x=33 to 62) Pull-down Enable, 1'b0: disable, 1'b1: enable	R/W

9.4.13 PADI_IO_DS_CFG_1

Base address: 0x5000_0800
Address offset: 0x30
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**IO_DS_IO_x**: IO x (1 to 32) Drive strength control, 1'b0: low drive strength, 1'b1: high drive strength	R/W

9.4.14 PADI_IO_DS_CFG_2

Base address: 0x5000_0800
Address offset: 0x34
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	Reserved	R/W
<29:0>	**IO_DS_IO_x**: IO x (33 to 62) Drive strength control, 1'b0: low drive strength, 1'b1: high drive strength	R/W

9.4.15 PADI register map

The base address of the PADI registers is 0x5000_0800.
The following table gives the PADI register map and reset values.

Table 27 PADI register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	PORTMUX_CFG_1	IOCFG_IO_8[3:0]				IOCFG_IO_7[3:0]				IOCFG_IO_6[3:0]				IOCFG_IO_5[3:0]				IOCFG_IO_4[3:0]				IOCFG_IO_3[3:0]				IOCFG_IO_2[3:0]				IOCFG_IO_1[3:0]			
	Reset value	0	1	1	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	1	1
0x04	PORTMUX_CFG_2	IOCFG_IO_16[3:0]				IOCFG_IO_15[3:0]				IOCFG_IO_14[3:0]				IOCFG_IO_13[3:0]				IOCFG_IO_12[3:0]				IOCFG_IO_11[3:0]				IOCFG_IO_10[3:0]				IOCFG_IO_9[3:0]			
	Reset value	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x08	PORTMUX_CFG_3	IOCFG_IO_24[3:0]				IOCFG_IO_23[3:0]				IOCFG_IO_22[3:0]				IOCFG_IO_21[3:0]				IOCFG_IO_20[3:0]				IOCFG_IO_19[3:0]				IOCFG_IO_18[3:0]				IOCFG_IO_17[3:0]			
	Reset value	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	11
0x0C	PORTMUX_CFG_4	IOCFG_IO_32[3:0]				IOCFG_IO_31[3:0]				IOCFG_IO_30[3:0]				IOCFG_IO_29[3:0]				IOCFG_IO_28[3:0]				IOCFG_IO_27[3:0]				IOCFG_IO_26[3:0]				IOCFG_IO_25[3:0]			
	Reset value	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x10	PORTMUX_CFG_5	IOCFG_IO_40[3:0]				IOCFG_IO_39[3:0]				IOCFG_IO_38[3:0]				IOCFG_IO_37[3:0]				IOCFG_IO_36[3:0]				IOCFG_IO_35[3:0]				IOCFG_IO_34[3:0]				IOCFG_IO_33[3:0]			
	Reset value	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x14	PORTMUX_CFG_6	IOCFG_IO_48[3:0]				IOCFG_IO_47[3:0]				IOCFG_IO_46[3:0]				IOCFG_IO_45[3:0]				IOCFG_IO_44[3:0]				IOCFG_IO_43[3:0]				IOCFG_IO_42[3:0]				IOCFG_IO_41[3:0]			
	Reset value	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x18	PORTMUX_CFG_7	IOCFG_IO_56[3:0]				IOCFG_IO_55[3:0]				IOCFG_IO_54[3:0]				IOCFG_IO_53[3:0]				IOCFG_IO_52[3:0]				IOCFG_IO_51[3:0]				IOCFG_IO_50[3:0]				IOCFG_IO_49[3:0]			
	Reset value	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
0x1C	PORTMUX_CFG_8	Res	Res	Res	Res	Res	Res	Res	Res	IOCFG_IO_62[3:0]				IOCFG_IO_61[3:0]				IOCFG_IO_60[3:0]				IOCFG_IO_59[3:0]				IOCFG_IO_58[3:0]				IOCFG_IO_57[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x20	IP_PULLUP_CFG_1	IO_PULLUP_IO_32	IO_PULLUP_IO_31	IO_PULLUP_IO_30	IO_PULLUP_IO_29	IO_PULLUP_IO_28	IO_PULLUP_IO_27	IO_PULLUP_IO_26	IO_PULLUP_IO_25	IO_PULLUP_IO_24	IO_PULLUP_IO_23	IO_PULLUP_IO_22	IO_PULLUP_IO_21	IO_PULLUP_IO_20	IO_PULLUP_IO_19	IO_PULLUP_IO_18	IO_PULLUP_IO_17	IO_PULLUP_IO_16	IO_PULLUP_IO_15	IO_PULLUP_IO_14	IO_PULLUP_IO_13	IO_PULLUP_IO_12	IO_PULLUP_IO_11	IO_PULLUP_IO_10	IO_PULLUP_IO_09	IO_PULLUP_IO_08	IO_PULLUP_IO_07	IO_PULLUP_IO_06	IO_PULLUP_IO_05	IO_PULLUP_IO_04	IO_PULLUP_IO_03	IO_PULLUP_IO_02	IO_PULLUP_IO_01
	Reset value	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	1	0	0	0	0
0x24	IP_PULLUP_CFG_2	Res	Res	IO_PULLUP_IO_62	IO_PULLUP_IO_61	IO_PULLUP_IO_60	IO_PULLUP_IO_59	IO_PULLUP_IO_58	IO_PULLUP_IO_57	IO_PULLUP_IO_56	IO_PULLUP_IO_55	IO_PULLUP_IO_54	IO_PULLUP_IO_53	IO_PULLUP_IO_52	IO_PULLUP_IO_51	IO_PULLUP_IO_50	IO_PULLUP_IO_49	IO_PULLUP_IO_48	IO_PULLUP_IO_47	IO_PULLUP_IO_46	IO_PULLUP_IO_45	IO_PULLUP_IO_44	IO_PULLUP_IO_43	IO_PULLUP_IO_42	IO_PULLUP_IO_41	IO_PULLUP_IO_40	IO_PULLUP_IO_39	IO_PULLUP_IO_38	IO_PULLUP_IO_37	IO_PULLUP_IO_36	IO_PULLUP_IO_35	IO_PULLUP_IO_34	IO_PULLUP_IO_33
	Reset value	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	0	0	0	0	0
0x28	IP_PULLDOWN_CFG_1	IO_PULLDW_IO_32	IO_PULLDW_IO_31	IO_PULLDW_IO_30	IO_PULLDW_IO_29	IO_PULLDW_IO_28	IO_PULLDW_IO_27	IO_PULLDW_IO_26	IO_PULLDW_IO_25	IO_PULLDW_IO_24	IO_PULLDW_IO_23	IO_PULLDW_IO_22	IO_PULLDW_IO_21	IO_PULLDW_IO_20	IO_PULLDW_IO_19	IO_PULLDW_IO_18	IO_PULLDW_IO_17	IO_PULLDW_IO_16	IO_PULLDW_IO_15	IO_PULLDW_IO_14	IO_PULLDW_IO_13	IO_PULLDW_IO_12	IO_PULLDW_IO_11	IO_PULLDW_IO_10	IO_PULLDW_IO_09	IO_PULLDW_IO_08	IO_PULLDW_IO_07	IO_PULLDW_IO_06	IO_PULLDW_IO_05	IO_PULLDW_IO_04	IO_PULLDW_IO_03	IO_PULLDW_IO_02	IO_PULLDW_IO_01
	Reset value	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
0x2C	IP_PDLLDOWN_CFG_2	Res	Res	IO_PULLDW_IO_62	IO_PULLDW_IO_61	IO_PULLDW_IO_60	IO_PULLDW_IO_59	IO_PULLDW_IO_58	IO_PULLDW_IO_57	IO_PULLDW_IO_56	IO_PULLDW_IO_55	IO_PULLDW_IO_54	IO_PULLDW_IO_53	IO_PULLDW_IO_52	IO_PULLDW_IO_51	IO_PULLDW_IO_50	IO_PULLDW_IO_49	IO_PULLDW_IO_48	IO_PULLDW_IO_47	IO_PULLDW_IO_46	IO_PULLDW_IO_45	IO_PULLDW_IO_44	IO_PULLDW_IO_43	IO_PULLDW_IO_42	IO_PULLDW_IO_41	IO_PULLDW_IO_40	IO_PULLDW_IO_39	IO_PULLDW_IO_38	IO_PULLDW_IO_37	IO_PULLDW_IO_36	IO_PULLDW_IO_35	IO_PULLDW_IO_34	IO_PULLDW_IO_33
	Reset value	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	0	1	0	0	0	0
0x30	IP_DS_CFG_1	IO_DS_IO_32	IO_DS_IO_31	IO_DS_IO_30	IO_DS_IO_29	IO_DS_IO_28	IO_DS_IO_27	IO_DS_IO_26	IO_DS_IO_25	IO_DS_IO_24	IO_DS_IO_23	IO_DS_IO_22	IO_DS_IO_21	IO_DS_IO_20	IO_DS_IO_19	IO_DS_IO_18	IO_DS_IO_17	IO_DS_IO_16	IO_DS_IO_15	IO_DS_IO_14	IO_DS_IO_13	IO_DS_IO_12	IO_DS_IO_11	IO_DS_IO_10	IO_DS_IO_09	IO_DS_IO_08	IO_DS_IO_07	IO_DS_IO_06	IO_DS_IO_05	IO_DS_IO_04	IO_DS_IO_03	IO_DS_IO_02	IO_DS_IO_01
	Reset value	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
0x34	IP_DS_CFG_2	Res	Res	IO_DS_IO_62	IO_DS_IO_61	IO_DS_IO_60	IO_DS_IO_59	IO_DS_IO_58	IO_DS_IO_57	IO_DS_IO_56	IO_DS_IO_55	IO_DS_IO_54	IO_DS_IO_53	IO_DS_IO_52	IO_DS_IO_51	IO_DS_IO_50	IO_DS_IO_49	IO_DS_IO_48	IO_DS_IO_47	IO_DS_IO_46	IO_DS_IO_45	IO_DS_IO_44	IO_DS_IO_43	IO_DS_IO_42	IO_DS_IO_41	IO_DS_IO_40	IO_DS_IO_39	IO_DS_IO_38	IO_DS_IO_37	IO_DS_IO_36	IO_DS_IO_35	IO_DS_IO_34	IO_DS_IO_33
	Reset value	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

10 Direct memory access controller (DMA)

10.1 Introduction

The direct memory access (DMA) is a controller used to perform data transfer between memory-mapped peripherals and/or memories from a source to a destination. The DMA controller has a total of 8 independent channels supporting 16 hardware handshaking interfaces. Each channel can be configured to manage memory access requests to/from peripherals.

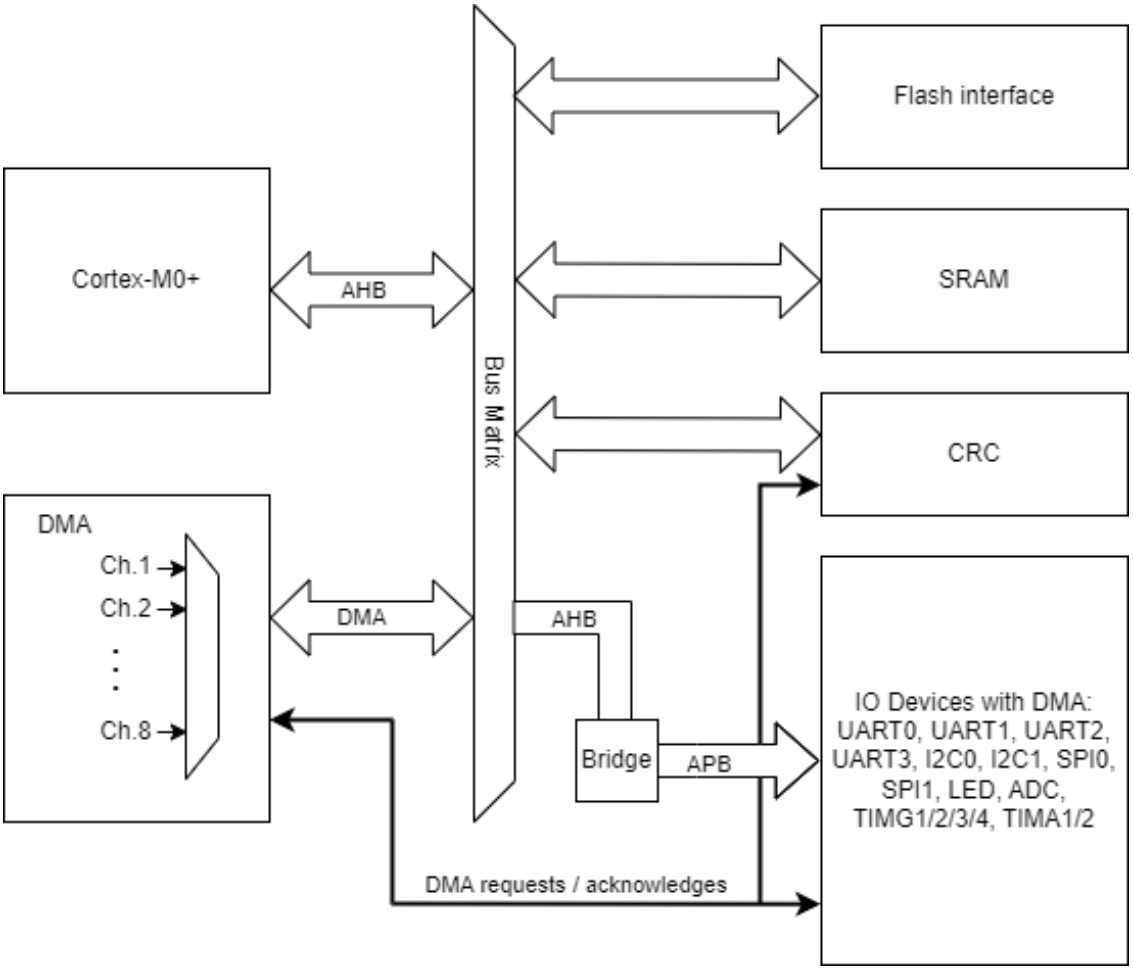


Figure 1. DMA Block diagram

10.2 DMA main features

- 8 independent channels with programmable source and destination for flexible mapping
- Programmable transfer type for each channel from memory to memory, peripheral to memory, memory to peripheral, and peripheral to peripheral
- Programmable channel priority
- Supports hardware or software handshaking flow control
- Interrupt generation upon transfer completion or error condition

10.3 DMA functional description

10.3.1 DMA Transfer

The device supports up to 8 DMA channels and 16 hardware handshaking interfaces. The software configures the DMA controller at the channel level to perform a block transfer. A DMA transfer may be requested from a peripheral using hardware handshaking mode or triggered by the software using software handshaking mode.

The following steps shows a typical DMA transfer:

1. The peripheral sends a DMA request to the DMA controller.
2. The DMA controller service the request, depending on the priority of the channel associated with the peripheral request.
3. Once the DMA controller grants the peripheral, an DMA acknowledge is sent to the peripheral by the DMA controller.
4. The peripheral releases its request as soon as it gets the acknowledge from the DMA controller.
5. Once the request is de-asserted by the peripheral, the DMA controller releases the acknowledge.

The same peripheral or other peripherals may initiate another DMA requests. A non-memory peripheral can request a DMA transfer either through hardware handshaking interface or software handshaking interface on a per-channel basis. Transfers are set up by programming the respective fields of the DMA configuration and DMA control registers: DMA_CFGx_L, DMA_CFGx_H, DMA_CTLx_L, and DMA_CTLx_H.

10.3.2 DMA arbitration

The DMA arbitrates between different DMA channels based on the assigned priority.

When an active channel x is granted by the DMA arbiter, a DMA transfer is initiated. The DMA arbiter considers the set of active channels and selects the one with the highest priority. If the DMA is transferring data for a lower priority channel and a higher priority channel requests service, the DMA will first complete the current transfer before servicing the higher priority channel. The channel priority is specified by setting the CHCFG_PRIOR[2:0] in the DMA_CFGx_L register.

10.3.3 DMA channels

Each channel may handle a DMA transfer between a source and a destination. The amount of data items to transfer is programmable. The DMA controller supports automatic address

increment/decrement/no change operations for either the source or destination address. The DMA source/destination address is specified by the DMA_SARx / DMA_DARx register. The address increment/decrement/no change configuration can be defined by setting the CHCTL_SINC[1:0] for the source address or CHCTL_DINC[1:0] for the destination address in the DMA_CTLx_L register.
The DMA transfer size is measured in terms of the source data transfer width defined by the CHCTL_SRTRW[2:0] in the DMA_CTLx_L register. The transfer size of each DMA channel is specified by setting the block data size, CHCTL_BLOCKTS[7:0], in the DMA_CTLx_H register. The DMA controller will transfer the entire block of data either through single transaction, burst transaction or a combination of single and burst transaction. If the block data size is not a multiple of the burst transaction data size during block transfer, the DMA will first transfer the data using burst transaction followed by single transaction. The data transfer width and the burst transaction length of the source/destination data is specified by the CHCTL_SRCTRW[2:0] / CHCTL_DSTTRW[2:0] and CHCTL_SRCMSIZE[2:0] / CHCTL_DSTMSIZE[2:0] in the DMA_CTLx_L register.

10.3.4 Handshaking interface

The DMA can operate in software or hardware handshaking mode.

10.3.4.1 Software handshaking mode

In the software handshaking mode, the peripheral first sends an interrupt to the interrupt controller. The interrupt service routine then utilizes the software handshaking registers: DMA_REQSRC, DMA_REQDST, DMA_SGLREQRC, DMA_SGLREQDST, DMA_LSTREQSRC, and DMA_LSTREQDST to initiate and control a DMA transaction. In addition, the CHCFG_SRCHS / CHCFG_DSTHS bit in the DMA_CFGx_L register must be set to enable software handshaking.
A burst transaction using software handshaking is initiated by first setting the bit: REQSRC_CH_x / REQDST_CH_x corresponding to channel x in the DMA_REQSRC / DMA_REQDST register to 1. Next, set the SGLREQSRC_CH_x / SGLREQDST_CH_x bit in the DMA_SGLREQSRC / DMA_SGLREQDST register for the same channel x.
To initiate a single transaction using software handshaking, simply write the SGLREQSRC_CH_x / SGLREQDST_CH_x bit in the DMA_SGLREQSRC / DMA_SGLREQDST register to 1.
In both cases, the hardware automatically clears the bit previously set in the DMA_REQSRC / DMA_REQDST as well as the DMA_SGLREQSRC / DMA_SGLREQDST registers upon completion of the transaction.

10.3.4.2 Hardware handshaking mode

Under hardware handshaking mode, a peripheral may utilize different DMA channel for data transfer. The user must first select the appropriate hardware handshaking interface for the peripheral by configuring the SYS_DMA_CFG and SYS_DMA_CFG_2 registers (refer to Chapter 1. System Configuration Controller). Please refer to the hardware handshaking interface assignments in Table 17 and Table 18 in Chapter 1 for details.
Once a particular hardware handshaking interface is selected, the next step is to assign the DMA channel for the hardware handshaking interface. This assignment is performed by configuring the

CHCFG_SRCPER[3:0] / CHCFG_DSTPER[3:0] bits in the DMA_CFGx_H register. Finally, define the handshaking interface polarity and enable hardware handshaking by writing the CHCFG_SRCHSPOL / CHCFG_DSTHSPOL bit and CHCFG_SRCHS / CHCFG_DSTHS bit in the DMA_CFGx_L register.

10.3.5 Register access

Users are advised to observe the following rules/restrictions regarding DMA register access.

1. When writing to DMA_SARx, DMA_DARx and DMA_CTLx registers, the channel should be disable.
2. DMA_CLEARBLOCK, DMA_CLEARDSTTRAN, DMA_CLEARERR, DMA_CLEARSRCTRAN and DMA_CLEARTFR are write only registers and should not be read.
3. DMA_STATUSBLOCK, DMA_STATUSDSTTRAN, DMA_STATUSERR, DMA_STATUSSRCTRAN and DMA_STATUSTFR are read only registers and cannot be written.
4. DMA_STATUSINT is read only register and cannot be written.

10.3.6 Data Transfer programming flow

To transfer data from a source to a destination, please follow the steps shown below:

1. Clear pending interrupts by writing 1 to DMA_CLEARTFR, DMA_CLEARBLOCK, DMA_CLEARSRCTRAN, DMA_CLEARDSTTRAN and DMA_CLEARERR
2. Program the DMA_CTLx_L and DMA_CTLx_H control register for Channel x
   i. Write to CHCTL_BLOCKTS to set block size
   ii. Write to CHCTL_TTFC to set transfer mode
   iii. Write to CHCTL_SRCTRW and CHCTL_DSTTRW to set source and destination transfer width
   iv. Write to CHCTL_SINC / CHCTL_DINC to increment, decrement or fixed address for the source/destination address
   v. Write to CHCTL_SRCMSIZE/CHCTL_DSTMSIZE to set burst transaction length
   vi. Write to CHCTL_INTEN to set interrupt enable
3. Program the DMA_CFGx_L and DMA_CFGx_H configuration register for Channel x
   i. Write CHCFG_SRCHS / CHCFG_DSTHS to set source / destination handshaking interface. If hardware interface is selected, write CHCFG_SRCPER[3:0] / CHCFG_DSTPER[3:0] to assign hardware handshaking interface (refer to Chapter 1. System configuration controller, section 1.4.8, 1.4.11 for details) and write CHCFG_SRCHSPOL / CHCFG_DSTHSPOL to set handshaking interface polarity.
   ii. Define the FIFO and the flow control method by setting the CHCFG_FMODE

and CHCFG_FCMODE

iii. Set the auto-reload cycle mode for the source/destination by writing the CHCFG_RELDSRC and CHCFG_RELDDST

iv. Clear the Channel suspend and define the channel priority by setting the CHCFG_SUSP and CHCFG_PRIOR bit

4. Set source address to DMA_SARx and set destination address to DMA_DARx

5. Clear pending interrupts by writing 1 to DMA_CLEARTFR, DMA_CLEARBLOCK, DMA_CLEARSRCTRAN, DMA_CLEARDSTTRAN and DMA_CLEARERR;

6. Enable DMAC by writing 1 to DMA_DMACFGREG.DMACFG_EN[0]

7. Enable DMA channel by writing 1 to DMA_CHENREG.CHEN_WE_x and to DMA_CHENREG.CHEN_EN_x

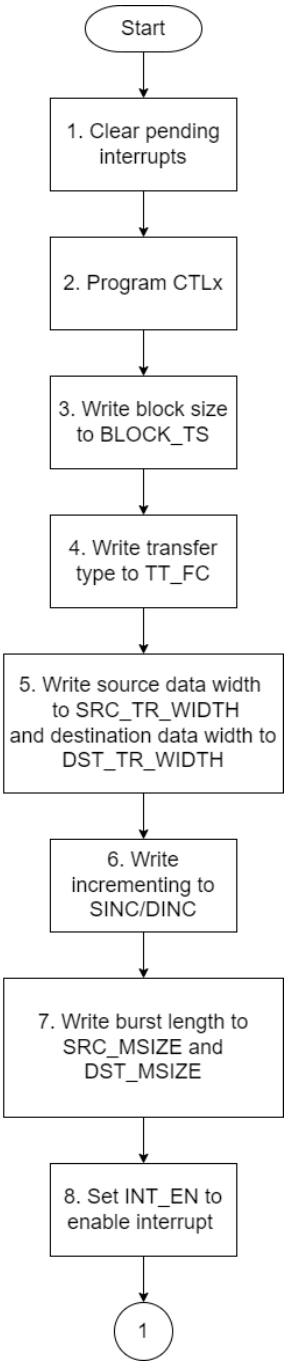


Figure 17 DMA transfer – 1

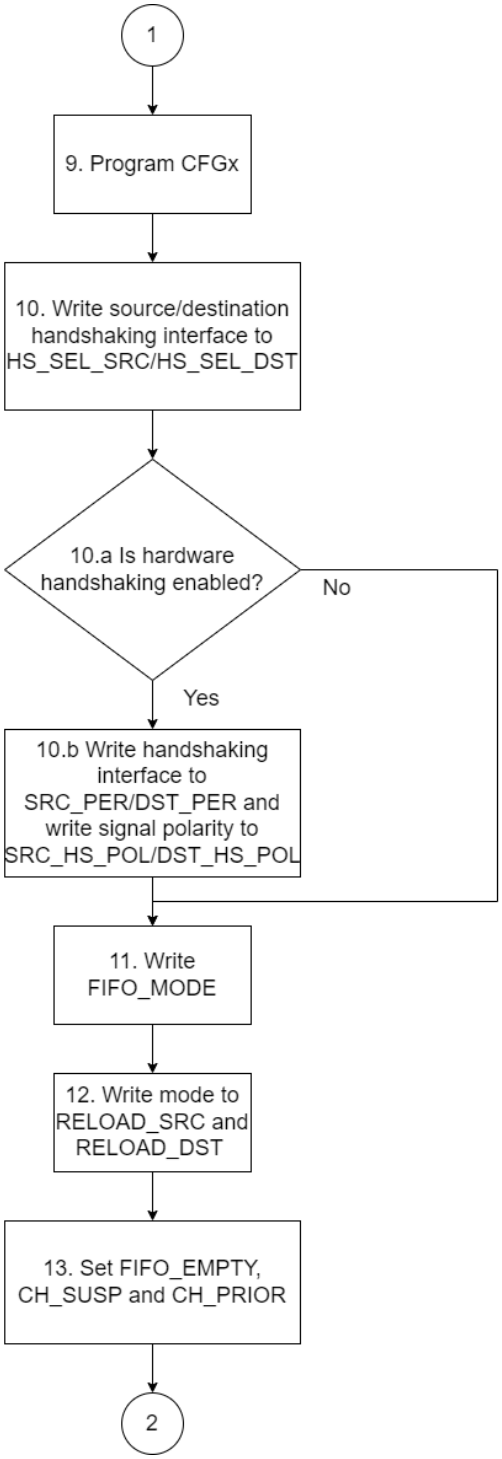


Figure 18 DMA transfer – 2

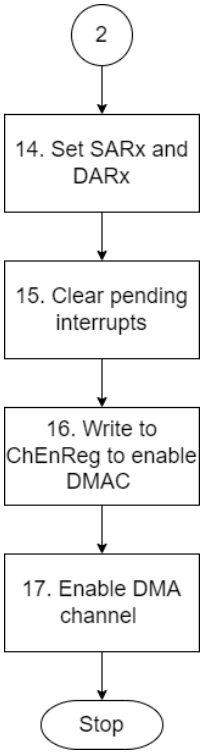


Figure 19 DMA transfer -3

10.3.7 Abort transaction

To abort DMA transfer, follow the steps shown below:

1. Disable interrupts by write 0 to DMA_CTLx_L.CHCTL_INTEN[0]
2. Suspend DMA by writing 1 to DMA_CFGx_L.CHCFG_SUSP[8]
3. Poll the DMA FIFO empty flag by reading DMA_CFGx_L.CHCFG_FIFOE[9]
4. Disable DMA channel by programming DMA_CHENREG.EN_x[7:0]
5. Clear any pending interrupts by writing 1 to DMA_CLEARTFR/ DMA_CLEARBLOCK/ DMA_CLEARSRCTRAN/ DMA_CLEARDSTTRAN/ DMA_CLEARERR

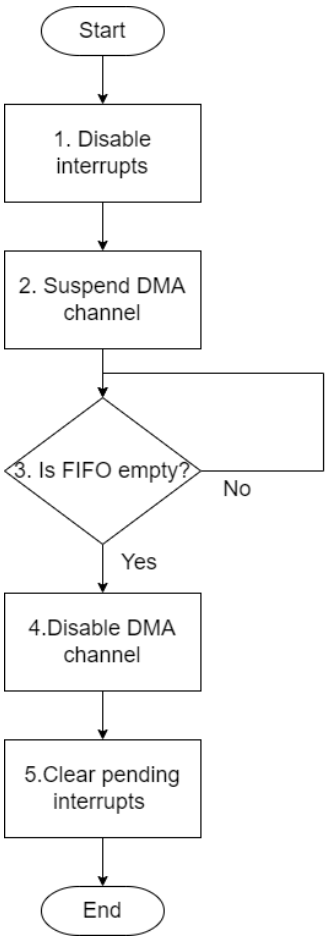


Figure 20 DMA abort

10.4 DMA registers

10.4.1 DMA_DMACFGREG

Base address: 0x8000_0000
Address offset: 0x398
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**DMACFG_EN**: DMAC enable 1' b0: DMAC disable 1' b1: DMAC enable	R/W

10.4.2 DMA_CHENREG

Base address: 0x8000_0000
Address offset: 0x3A0
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**CHEN_WE_x**: Channel x (x=0 to 7) enable write enable 1'b0: disable write 1'b1: enable write	R/W
<7:0>	**CHEN_EN_x**:Channel x (x=0 to 7) enable. When a write channel enable, channel enable write enable should be enabled. 1'b0: DMA channel disable 1'b1: DMA channel enable	R/W

10.4.3 DMA_SARx

Base address: 0x8000_0000
Address offset:
SAR0 0x000
SAR1 0x058
SAR2 0x0B0
SAR3 0x108
SAR4 0x160
SAR5 0x1B8
SAR6 0x210
SAR7 0x268
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**SAR_SAR[31:0]**: Current source address	R/W

10.4.4 DMA_DARx

Base address: 0x8000_0000
Address offset:
DAR0 0x008
DAR1 0x060
DAR2 0x0B8
DAR3 0x110
DAR4 0x168
DAR5 0x1C0
DAR6 0x218
DAR7 0x270
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**DAR_DAR[31:0]**: Current destination address	R/W

10.4.5 DMA_CTLx_L

Base address: 0x8000_0000
Address offset:
CTL0_L 0x018
CTL1_L 0x070
CTL2_L 0x0C8
CTL3_L 0x120
CTL4_L 0x178
CTL5_L 0x1D0
CTL6_L 0x228
CTL7_L 0x280
Type: Read/Write
Reset value: 0x0030_4801

Bit	Description	Type
<31:23>	Reserved	R/W
<22:20>	**CHCTL_TTFC[2:0]**: Transfer type 3'b000: memory to memory 3'b001: memory to peripheral 3'b010: peripheral to memory 3'b011: peripheral to peripheral	R/W
<19:17>	Reserved	R/W
<16:14>	**CHCTL_SRCMSIZE[2:0]**: Source burst transaction length 3'b000: 1 @SRC_TR_WIDTH 3'b001: 4 @SRC_TR_WIDTH	R/W
<13:11>	**CHCTL_DSTMSIZE[2:0]**: Destination burst transaction length 3'b000: 1 @DST_TR_WIDTH 3'b001: 4 @DST_TR_WIDTH	R/W
<10:9>	**CHCTL_SINC[1:0]**: Source address increment 2'b00: increment 2'b01: decrement 2'b1x: no change	R/W
<8:7>	**CHCTL_DINC[1:0]**: Destination address increment 2'b00: increment 2'b01: decrement 2'b1x: no change	R/W
<6:4>	**CHCTL_SRCTRW[2:0]**: Source data transfer width 3'b000: 8 bits 3'b001: 16 bits 3'b010: 32 bits	R/W
<3:1>	**CHCTL_DSTTRW[2:0]**: Destination data transfer width	R/W

Bit	Description	Type
	3’b000: 8 bits 3’b001: 16 bits 3’b010: 32 bits	
<0>	**CHCTL_INTEN**: Interrupt enable 2’b0: disable 2’b1: enable	R/W

10.4.6 DMA_CTLx_H

Base address: 0x8000_0000

Address offset:

CTL0_H 0x01C
CTL1_H 0x074
CTL2_H 0x0CC
CTL3_H 0x124
CTL4_H 0x17C
CTL5_H 0x1D4
CTL6_H 0x22C
CTL7_H 0x284

Type: Read/Write

Reset value: 0x0000_0002

Bit	Description	Type
<31:13>	Reserved	R
<12>	**CHCTL_DONE**：Block transfer is complete.	R/W
<11:8>	Reserved	R/W
<7:0>	**CHCTL_BLOCKTS[7:0]**: Block data size in multiples of source transfer width	R/W

10.4.7 DMA_CFGx_L

Base address: 0x8000_0000

Address offset:

CFG0_L 0x040 (0x0000_0E00)
CFG1_L 0x098 (0x0000_0E20)
CFG2_L 0x0F0 (0x0000_0E40)
CFG3_L 0x148 (0x0000_0E60)
CFG4_L 0x1A0 (0x0000_0E80)
CFG5_L 0x1F8 (0x0000_0EA0)
CFG6_L 0x250 (0x0000_0EC0)
CFG7_L 0x2A8 (0x0000_0EE0)

Type: Read/Write

Reset value: 0x0000_0E00

Bit	Description	Type
<31>	**CHCFG_RELDDST**: Automatic destination reload. The DMA_DARx register will be reloaded automatically from its initial value at the end of every block for multi-block transfer followed by a new block transfer. 1'b0: disable 1'b1: enable	R/W
<30>	**CHCFG_RELDSRC**: Automatic source reload. The DMA_SARx register will be reloaded automatically from its initial value at the end of every block for multi-block transfer followed by a a new block transfer. 1'b0: disable 1'b1: enable	R/W
<29:20>	Reserved	R/W
<19>	**CHCFG_SRCHSPOL**: Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low	R/W
<18>	**CHCFG_DSTHSPOL**: Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low	R/W
<17:12>	Reserved	R/W
<11>	**CHCFG_SRCHS**: Source software or hardware handshaking select. Selects the handshaking interface for source requests on this channel. If the source peripheral is memory, then this bit is ignored. 1'b0: hardware handshaking 1'b1: software handshaking	R/W
<10>	**CHCFG_DSTHS**: Destination software or hardware handshaking select. Selects the handshaking interface for destination requests on this channel. If the destination peripheral is memory, then this bit is ignored.	R/W

Bit	Description	Type
	1'b0: hardware handshaking 1'b1: software handshaking	
<9>	**CHCFG_FIFOE**: Indicates whether the channel FIFO is empty. Can be used in conjunction with CFGx.CH_SUSP to disable channel. This bit is read-only. 1'b0: channel FIFO is not empty. 1'b1: channel FIFO is empty.	R
<8>	**CHCFG_SUSP**: Channel suspend. Suspends all DMA transfer until this bit is cleared. 1'b0: not suspended. 1'b1: suspend	R/W
<7:5>	**CHCFG_PRIOR[2:0]**: Channel priority y (y=0 to 7). A priority of 7 is the highest priority and 0 is the lowest priority. 3'b000: Channel priority is 0 3'b001: Channel priority is 1 … 3'b111: Channel priority is 7	R/W
<4:0>	Reserved	R

10.4.8 DMA_CFGx_H

Base address: 0x8000_0000

Address offset:

CFG0_H 0x044
CFG1_H 0x09C
CFG2_H 0x0F4
CFG3_H 0x14C
CFG4_H 0x1A4
CFG5_H 0x1FC
CFG6_H 0x254
CFG7_H 0x2AC

Type: Read/Write

Reset value: 0x0000_0004

Bit	Description	Type
<31:15>	Reserved	R/W
<14:11>	**CHCFG_DSTPER[3:0]**: Destination handshaking interface y (y=0 to 15). Assigns a hardware handshaking interface to the destination of channel x if the CHCFG_DSTHS bit is set to 0 in the DMA_CFGx_L register.	R/W
<10:7>	**CHCFG_SRCPER[3:0]**: Source handshaking interface y (y=0 to 15). Assigns a hardware handshaking interface to the source of channel x if the CHCFG_SRCHS bit is set to 0 in the DMA_CFGx_L register.	R/W
<6:2>	Reserved	R/W

<1>	**CHCFG_FMODE**: Defines how much space or data needs to be filled in the FIFO before a burst transaction is serviced. 1'b0: Space/data available for single AHB transfer 1'b1: Data available is greater than or equal to half of FIFO depth for destination transfers and space available is greater than or equal to half of FIFO depth for source transfers, except at the end of a burst transaction.	R/W
<0>	**CHCFG_FCMODE**: Specify when the source transaction requests are serviced if the destination is the flow controller. 1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled 1'b1: Source transaction requests are not serviced until the destination transaction request occurs. Data pre-fetching is disabled.	R/W

10.4.9 DMA_RAWTFR

Base address: 0x8000_0000
Address offset: 0x2C0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R
<7:0>	**RAWTFR_CH_x**: Channel x (x=0 to 7) transfer complete interrupt raw status	R

10.4.10 DMA_RAWBLOCK

Base address: 0x8000_0000
Address offset: 0x2C8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**RAWBLOCK_CH_x**: Channel x (x=0 to 7) block transfer complete interrupt raw status	R/W

10.4.11 DMA_RAWSRCTRAN

Base address: 0x8000_0000
Address offset: 0x2D0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R
<7:0>	**RAWSRCTRAN_CH_x**: Channel x (x=0 to 7) source transaction complete interrupt raw status	R

10.4.12 DMA_RAWDSTTRAN

Base address: 0x8000_0000
Address offset: 0x2D8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**RAWDSTTRAN_CH_x**: Channel x (x=0 to 7) destination transaction complete interrupt raw status	R

10.4.13 DMA_RAWERR

Base address: 0x8000_0000
Address offset: 0x2E0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**RAWERR_CH_x**: Channel x (x=0 to 7) error interrupt raw status	R

10.4.14 DMA_STATUSTFR

Base address: 0x8000_0000
Address offset: 0x2E8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**STATTFR_CH_**: Channel x (x=0 to 7) transfer complete interrupt status	R

10.4.15 DMA_STATUSBLOCK

Base address: 0x8000_0000
Address offset: 0x2F0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**STATBLOCK_CH_x**: Channel x (x=0 to 7) block transfer complete interrupt status	R

10.4.16 DMA_STATUSSRCTRAN

Base address: 0x8000_0000
Address offset: 0x2F8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**STATSRCTRAN_CH_x**: Channel x (x=0 to 7) source transaction complete interrupt status	R

10.4.17 DMA_STATUSDSTTRAN

Base address: 0x8000_0000
Address offset: 0x300
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**STATDSTTRAN_CH_x**: Channel x (x=0 to 7) destination transaction complete interrupt status	R

10.4.18 DMA_STATUSERR

Base address: 0x8000_0000
Address offset: 0x308
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**STATERR_CH_x**: Channel x (x=0 to 7) error interrupt status	R

10.4.19 DMA_MASKTFR

Base address: 0x8000_0000
Address offset: 0x310
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**MSKTFR_WE_x**: Chanel x (x=0 to 7) interrupt mask write enable 1'b0: write disabled 1'b1: write enabled	R/W
<7:0>	**MSKTFR_EN_x**: Channel x (x=0 to 7) transfer complete interrupt mask, must be write with WE_MSKTFR_CH[7:0] 1'b0: masked 1'b1: unmasked	R/W

10.4.20 DMA_MASKBLOCK

Base address: 0x8000_0000
Address offset: 0x318
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**MSKBLOCK_WE_x**: Channel x (x=0 to 7) interrupt mask write enable 1'b0: write disabled 1'b1: write enabled	R/W
<7:0>	**MSKBLOCK_EN_x**: Channel x (x=0 to 7) block transfer complete interrupt mask, must be write with WE_MSKBLK_CH[7:0] 1'b0: masked 1'b1: unmasked	R/W

10.4.21 DMA_MASKSRCTRAN

Base address: 0x8000_0000
Address offset: 0x320
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**MSKSRCTRAN_WE_x**: Channel x (x=0 to 7) interrupt mask write enable 1'b0: write disabled 1'b1: write enabled	R/W
<7:0>	**MSKSRCTRAN_EN_x**: Channel x (x=0 to 7) source transaction complete interrupt mask, must be write with WE_MSKSRCT_CH[7:0] 1'b0: masked 1'b1: unmasked	R/W

10.4.22 DMA_MASKDSTTRAN

Base address: 0x8000_0000
Address offset: 0x328
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**MSKDSTTRAN_WE[7:0]**: Channel x (x=0 to 7) interrupt mask write enable 1'b0: Write disabled 1'b1: Write enabled	R/W
<7:0>	**MSKDSTTRAN_EN**: Channel x (x=0 to 7) destination transaction complete interrupt mask, must be write with WE_MSKDSTT_CH[7:0] 1'b0: masked 1'b1: unmasked	R/W

10.4.23 DMA_MASKERR

Base address: 0x8000_0000
Address offset: 0x330
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W

<15:8>	**MSKERR_WE_x**: Channel x (x=0 to 7) interrupt mask write enable 1'b0: Write disabled 1'b1: Write enabled	R/W
<7:0>	**MSKERR_EN_x**: Channel x (x=0 to 7) error interrupt mask, must be write with WE_MSKERR_CH[7:0] 1'b0: Masked 1'b1: Unmasked	R/W

10.4.24 DMA_CLEARTFR

Base address: 0x8000_0000
Address offset: 0x338
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CLRTFR_CH_x**: Channel x (x=0 to 7) transfer complete interrupt clear 1'b0: No effect 1'b1: Clear interrupt	W

10.4.25 DMA_CLEARBLOCK

Base address: 0x8000_0000
Address offset: 0x340
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CLRBLOCK_CH_x**: Channel x (x=0 to 7) block transfer complete interrupt clear 1'b0: No effect 1'b1: Clear interrupt	W

10.4.26 DMA_CLEARSRCTRAN

Base address: 0x8000_0000
Address offset: 0x348
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CLRSRCTRAN_CH_x**: Channel x (x=0 to 7) source transaction complete interrupt clear 1'b0: No effect 1'b1: Clear interrupt	W

10.4.27 DMA_CLEARDSTTRAN

Base address: 0x8000_0000
Address offset: 0x350
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CLRDSTTRAN_CH_x**: Channel x (x=0 to 7) destination transaction complete interrupt clear 1'b0: No effect 1'b1: Clear interrupt	W

10.4.28 DMA_CLEARERR

Base address: 0x8000_0000
Address offset: 0x358
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CLRERR_CH_x**: Channel x (x=0 to 7) error interrupt clear 1'b0: No effect 1'b1: Clear interrupt	W

10.4.29 DMA_STATUSINT

Base address: 0x8000_0000
Address offset: 0x360
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:5>	Reserved	R/W
<4>	**STATINT_ERR**: OR of the contents of STATUSERR	R
<3>	**STATINT_DSTT**: OR of the contents of STATUSDSTTRAN	R
<2>	**STATINT_SRCT**: OR of the contents of STATUSSRCTRAN	R
<1>	**STATINT_BLOCK**: OR of the contents of STATUSBLOCK	R
<0>	**STATINT_TFR**: OR of the contents of STATUSTFR	R

10.4.30 DMA_REQSRC

Base address: 0x8000_0000
Address offset: 0x368
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**REQSRC_WE_x**: Channel x (x=0 to 7) source software transaction request write enable. Each bit in REQSRC_WE_x corresponds to the settings for Channel x. 1'b0: Source request write disabled 1'b1: Source request write enabled	R/W
<7:0>	**REQSRC_CH_x**: Channel x (x=0 to 7) source software transaction request, must be written with REQSRC_WE_x[7:0]. Each bit in REQSRC_CH_x corresponds to the settings for Channel x. 1'b0: Source request inactive 1'b1: Source request active	R/W

10.4.31 DMA_REQDST

Base address: 0x8000_0000
Address offset: 0x370
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**REQDST_WE_x**: Channel x (x=0 to 7) destination software transaction request write enable. Each bit in REQDST_WE_x corresponds to the settings for Channel x. 1'b0: Destination request write disabled 1'b1: Destination request write enabled	R/W
<7:0>	**REQDST_CH_x**: Channel x (x=0 to 7) destination software transaction request, must be written with REQDST_WE_x[7:0]. Each bit in REQDST_CH_x corresponds to the settings for Channel x. 1'b0: Destination request inactive 1'b1: Destination request active	R/W

10.4.32 DMA_SGLREQSRC

Base address: 0x8000_0000
Address offset: 0x378
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**SGLREQSRC_WE_x**: Channel x (x=0 to 7) source single transaction request write enable. 1'b0: Single write disabled 1'b1: Single write enabled	R/W
<7:0>	**SGLREQSRC_CH_x**: Channel x (x=0 to 7) source single or burst request, must be written with SGLREQSRC_WE_x[7:0]. Each bit in SGLREQSRC_CH_x corresponds to the settings for Channel x. 1'b0: Source request inactive 1'b1: Source request active	R/W

10.4.33 DMA_SGLREQDST

Base address: 0x8000_0000
Address offset: 0x380
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**SGLREQDST_WE_x**: Channel x (x=0 to 7) destination single request write enable 1'b0: Destination write disabled 1'b1: Destination write enabled	R/W
<7:0>	**SGLREQDST_CH_x**: Channel x (x=0 to 7) destination single or burst request, must be written with SGLREQDST_WE_x[7:0]. Each bit in SGLREQDST_CH_x corresponds to the settings for Channel x. 1'b0: Destination single or burst request inactive 1'b1: Destination single or burst request active	R/W

10.4.34 DMA_LSTREQSRC

Base address: 0x8000_0000
Address offset: 0x388
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**LSTREQSRC_WE_x**: Channel x (x=0 to 7) source last transaction request write enable 1'b0: Source last transaction request write disabled 1'b1: Source last transaction request write enabled	R/W
<7:0>	**LSTREQSRC_CH_x**: Channel x (x=0 to 7) source last transaction request, must be written with LSTREQSRC_WE_x[7:0]. Each bit in LSTREQSRC_CH_x corresponds to the settings for Channel x. 1'b0: Not last transaction in current block 1'b1: Last transaction in current block	R/W

10.4.35 DMA_LSTREQDST

Base address: 0x8000_0000
Address offset: 0x390
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**LSTREQDST_WE_x**: Channel x (x=0 to 7) destination last transaction request write enable 1'b0: Destination last transaction write disabled 1'b1: Destination last transaction write enabled	R/W
<7:0>	**LSTREQDST_CH_x**: Channel x (x=0 to 7) destination last request, must be written with LSTREQDST_WE_x[7:0]. Each bit in LSTREQDST_CH_x corresponds to the settings for Channel x. 1'b0: Not last transaction in current block 1'b1: Last transaction in current block	R/W

10.4.36 DMA register map

The base address of the DMA registers is 0x8000_0000.
The following table gives the DMA register map and reset values.

Table 28 DMA registers and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	SAR0	SAR_SAR[31:0]																															
	Reset value	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
0x008	DAR0	DAR_DAR[31:0]																															
	Reset value	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
0x018	CTL0_L	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_TTFC[2:0]			Res	Res	Res	CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]			CHCTL_SINC[1:0]		CHCTL_DINC[1:0]		CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]			ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0x01C	CTL0_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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	1	0
0x040	CFG0_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
0x044	CFG0_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]				CHCFG_SRCPER[3:0]				Res	Res	Res	Res	Res	CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x058	SAR1	SAR_SAR[31:0]																															
	Reset value	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
0x060	DAR1	DAR_DAR[31:0]																															
	Reset value	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
0x070	CTL1_L	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_TTFC[2:0]			Res	Res	Res	CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]			CHCTL_SINC[1:0]		CHCTL_DINC[1:0]		CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]			ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x074	CTL1_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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
0x098	CFG1_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0
0x09C	CFG1_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]							CHCFG_SRCPER[3:0]						CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x0B0	SAR2	SAR_SAR[31:0]																															
	Reset value	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
0x0B8	DAR2	DAR_DAR[31:0]																															
	Reset value	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
0x0C8	CTL2_L	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_TTFC[2:0]			Res	Res	Res	CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]			CHCTL_SINC[1:0]		CHCTL_DINC[1:0]		CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]			ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0CC	CTL2_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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
0x0F0	CFG2_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0
0x0F4	CFG2_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]							CHCFG_SRCPER[3:0]						CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x108	SAR3	SAR_SAR[31:0]																															
	Reset value	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
0x110	DAR3	DAR_DAR[31:0]																															

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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
0x120	CTL3_L	Res	Res	Res	Res	Res	Res	Res	Res	Res			CHCTL_TTFC[2:0]	Res	Res	Res			CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]			CHCTL_SINC[1:0]		CHCTL_DINC[1:0]			CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]			ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0x124	CTL3_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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	1	0
0x148	CFG3_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0	0
0x14C	CFG3_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]				CHCFG_SRCPER[3:0]				Res	Res	Res	Res	Res	CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x160	SAR4	SAR_SAR[31:0]																															
	Reset value	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
0x168	DAR4	DAR_DAR[31:0]																															
	Reset value	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
0x178	CTL4_L	Res	Res	Res	Res	Res	Res	Res	Res	Res			CHCTL_TTFC[2:0]	Res	Res	Res			CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]			CHCTL_SINC[1:0]		CHCTL_DINC[1:0]			CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]			ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0x17C	CTL4_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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	1	0
0x1A0	CFG4_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1A4	CFG4_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]				CHCFG_SRCPER[3:0]				Res	Res	Res	Res	Res	CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x1B8	SAR5	SAR_SAR[31:0]																															
	Reset value	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
0x1C0	DAR5	DAR_DAR[31:0]																															
	Reset value	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
0x1D0	CTL5_L	Res	Res	Res	Res	Res	Res	Res	Res	Res			CHCTL_TTFC[2:0]	Res	Res	Res			CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]		CHCTL_SINC[1:0]		CHCTL_DINC[1:0]			CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]		ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0x1D4	CTL5_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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	1	0
0x1F8	CFG5_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP			CHCFG_PRIOR[2:0]		Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0
0x1FC	CFG5_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]				CHCFG_SRCPER[3:0]				Res	Res	Res	Res	Res	CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x210	SAR6	SAR_SAR[31:0]																															
	Reset value	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
0x218	DAR6	DAR_DAR[31:0]																															
	Reset value	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
228	CTL6_L	Res	Res	Res	Res	Res	Res	Res	Res	Res			CHCTL_TTFC[2:0]	Res	Res	Res			CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]		CHCTL_SINC[1:0]		CHCTL_DINC[1:0]			CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]		ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0x22C	CTL6_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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	1	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x250	CFG6_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0
0x254	CFG6_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]				CHCFG_SRCPER[3:0]				Res	Res	Res	Res	Res	CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x268	SAR7	SAR_SAR[31:0]																															
	Reset value	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
0x270	DAR7	DAR_DAR[31:0]																															
	Reset value	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
0x280	CTL7_L	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_TTFC[2:0]			Res	Res	Res	CHCTL_SRCMSIZE[2:0]			CHCTL_DSTMSIZE[2:0]			CHCTL_SINC[1:0]		CHCTL_DINC[1:0]		CHCTL_SRCTRW[2:0]			CHCTL_DSTTRW[2:0]			ICHCTL_NTEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1
0x284	CTL7_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCTL_DONE	Res	Res	Res	Res	CHCTL_BLOCKTS[7:0]							
	Reset value	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	1	0
0x2A0	CFG7_L	CHCFG_RELDDST	CHCFG_RELDSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_SRCHSPOL	CHCFG_DSTHSPOL	Res	Res	Res	Res	Res	Res	CHCFG_SRCHS	CHCFG_DSTHS	CHCFG_FIFOE	CHCFG_SUSP	CHCFG_PRIOR[2:0]			Res	Res	Res	Res	Res
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0
0x2A4	CFG7_H	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHCFG_DSTPER[3:0]				CHCFG_SRCPER[3:0]				Res	Res	Res	Res	Res	CHFG_FMODE	CHFG_FCMODE
	Reset value	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	1	0	0
0x2C0	RAWTFR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RAWTFR_CH_7	RAWTFR_CH_6	RAWTFR_CH_5	RAWTFR_CH_4	RAWTFR_CH_3	RAWTFR_CH_2	RAWTFR_CH_1	RAWTFR_CH_0
	Reset value	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
0x2C8	RAWBLOCK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RAWBLOCK_CH_7	RAWBLOCK_CH_6	RAWBLOCK_CH_5	RAWBLOCK_CH_4	RAWBLOCK_CH_3	RAWBLOCK_CH_2	RAWBLOCK_CH_1	RAWBLOCK_CH_0
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2D0	RAWSRCT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RAWSRCTRAN_CH_7	RAWSRCTRAN_CH_6	RAWSRCTRAN_CH_5	RAWSRCTRAN_CH_4	RAWSRCTRAN_CH_3	RAWSRCTRAN_CH_2	RAWSRCTRAN_CH_1	RAWSRCTRAN_CH_0
	Reset value	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
0x2D8	RAWDSTT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RAWDSTTRAN_CH_7	RAWDSTTRAN_CH_6	RAWDSTTRAN_CH_5	RAWDSTTRAN_CH_4	RAWDSTTRAN_CH_3	RAWDSTTRAN_CH_2	RAWDSTTRAN_CH_1	RAWDSTTRAN_CH_0
	Reset value	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
0x2E0	RAWERR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RAWERR_CH_7	RAWERR_CH_6	RAWERR_CH_5	RAWERR_CH_4	RAWERR_CH_3	RAWERR_CH_2	RAWERR_CH_1	RAWERR_CH_0
	Reset value	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
0x2E8	STATUSTFR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STATTFR_CH_7	STATTFR_CH_6	STATTFR_CH_5	STATTFR_CH_4	STATTFR_CH_3	STATTFR_CH_2	STATTFR_CH_1	STATTFR_CH_0
	Reset value	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
0x2F0	STATUSBLOCK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STATBLOCK_CH_7	STATBLOCK_CH_6	STATBLOCK_CH_5	STATBLOCK_CH_4	STATBLOCK_CH_3	STATBLOCK_CH_2	STATBLOCK_CH_1	STATBLOCK_CH_0
	Reset value	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
0x2F8	STATUSSRCT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STATSRCTRAN_CH_7	STATSRCTRAN_CH_6	STATSRCTRAN_CH_5	STATSRCTRAN_CH_4	STATSRCTRAN_CH_3	STATSRCTRAN_CH_2	STATSRCTRAN_CH_1	STATSRCTRAN_CH_0
	Reset value	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
0x300	STATUSDSTT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STATDSTTRAN_CH_7	STATDSTTRAN_CH_6	STATDSTTRAN_CH_5	STATDSTTRAN_CH_4	STATDSTTRAN_CH_3	STATDSTTRAN_CH_2	STATDSTTRAN_CH_1	STATDSTTRAN_CH_0
	Reset value	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
0x308	STATUSERR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STATERR_CH_7	STATERR_CH_6	STATERR_CH_5	STATERR_CH_4	STATERR_CH_3	STATERR_CH_2	STATERR_CH_1	STATERR_CH_0
	Reset value	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
0x310	MASKTFR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MSKTFR_WE_7	MSKTFR_WE_6	MSKTFR_WE_5	MSKTFR_WE_4	MSKTFR_WE_3	MSKTFR_WE_2	MSKTFR_WE_1	MSKTFR_WE_0	MSKTFR_EN_7	MSKTFR_EN_6	MSKTFR_EN_5	MSKTFR_EN_4	MSKTFR_EN_3	MSKTFR_EN_2	MSKTFR_EN_1	MSKTFR_EN_0
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x318	MASKBLOCK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MSKBLOCK_WE_7	MSKBLOCK_WE_6	MSKBLOCK_WE_5	MSKBLOCK_WE_4	MSKBLOCK_WE_3	MSKBLOCK_WE_2	MSKBLOCK_WE_1	MSKBLOCK_WE_0	MSKBLOCK_EN_7	MSKBLOCK_EN_6	MSKBLOCK_EN_5	MSKBLOCK_EN_4	MSKBLOCK_EN_3	MSKBLOCK_EN_2	MSKBLOCK_EN_1	MSKBLOCK_EN_0
	Reset value	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
0x320	MASKSRCT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MSKSRCTRAN_WE_7	MSKSRCTRAN_WE_6	MSKSRCTRAN_WE_5	MSKSRCTRAN_WE_4	MSKSRCTRAN_WE_3	MSKSRCTRAN_WE_2	MSKSRCTRAN_WE_1	MSKSRCTRAN_WE_0	MSKSRCTRAN_EN_7	MSKSRCTRAN_EN_6	MSKSRCTRAN_EN_5	MSKSRCTRAN_EN_4	MSKSRCTRAN_EN_3	MSKSRCTRAN_EN_2	MSKSRCTRAN_EN_1	MSKSRCTRAN_EN_0
	Reset value	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
0x328	MASKDSTT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MSKDSTTRAN_WE_7	MSKDSTTRAN_WE_6	MSKDSTTRAN_WE_5	MSKDSTTRAN_WE_4	MSKDSTTRAN_WE_3	MSKDSTTRAN_WE_2	MSKDSTTRAN_WE_1	MSKDSTTRAN_WE_0	MSKDSTTRAN_EN_7	MSKDSTTRAN_EN_6	MSKDSTTRAN_EN_5	MSKDSTTRAN_EN_4	MSKDSTTRAN_EN_3	MSKDSTTRAN_EN_2	MSKDSTTRAN_EN_1	MSKDSTTRAN_EN_0
	Reset value	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
0x330	MASKERR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MSKERR_WE_7	MSKERR_WE_6	MSKERR_WE_5	MSKERR_WE_4	MSKERR_WE_3	MSKERR_WE_2	MSKERR_WE_1	MSKERR_WE_0	MSKERR_EN_7	MSKERR_EN_6	MSKERR_EN_5	MSKERR_EN_4	MSKERR_EN_3	MSKERR_EN_2	MSKERR_EN_1	MSKERR_EN_0
	Reset value	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
0x338	CLEARTFR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRTFR_CH_7	CLRTFR_CH_6	CLRTFR_CH_5	CLRTFR_CH_4	CLRTFR_CH_3	CLRTFR_CH_2	CLRTFR_CH_1	CLRTFR_CH_0
	Reset value	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
0x340	CLEARBLOCK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRBLOCK_CH_7	CLRBLOCK_CH_6	CLRBLOCK_CH_5	CLRBLOCK_CH_4	CLRBLOCK_CH_3	CLRBLOCK_CH_2	CLRBLOCK_CH_1	CLRBLOCK_CH_0
	Reset value	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
0x348	CLEARSRCT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRSRCTRAN_CH_7	CLRSRCTRAN_CH_6	CLRSRCTRAN_CH_5	CLRSRCTRAN_CH_4	CLRSRCTRAN_CH_3	CLRSRCTRAN_CH_2	CLRSRCTRAN_CH_1	CLRSRCTRAN_CH_0
	Reset value	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
0x350	CLEARDSTT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRDSTTRAN_CH_7	CLRDSTTRAN_CH_6	CLRDSTTRAN_CH_5	CLRDSTTRAN_CH_4	CLRDSTTRAN_CH_3	CLRDSTTRAN_CH_2	CLRDSTTRAN_CH_1	CLRDSTTRAN_CH_0
	Reset value	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
0x358	CLEARERR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLRERR_CH_7	CLRERR_CH_6	CLRERR_CH_5	CLRERR_CH_4	CLRERR_CH_3	CLRERR_CH_2	CLRERR_CH_1	CLRERR_CH_0
	Reset value	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
0x360	STATUSINT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ERR	DSTT	SRCT	BLOCK	TFR

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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
0x368	REQSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REQSRC_WE_7	REQSRC_WE_6	REQSRC_WE_5	REQSRC_WE_4	REQSRC_WE_3	REQSRC_WE_2	REQSRC_WE_1	REQSRC_WE_0	REQSRC_CH_7	REQSRC_CH_6	REQSRC_CH_5	REQSRC_CH_4	REQSRC_CH_3	REQSRC_CH_2	REQSRC_CH_1	REQSRC_CH_0
	Reset value	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
0x370	REQDST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REQDST_WE_7	REQDST_WE_6	REQDST_WE_5	REQDST_WE_4	REQDST_WE_3	REQDST_WE_2	REQDST_WE_1	REQDST_WE_0	REQDST_CH_7	REQDST_CH_6	REQDST_CH_5	REQDST_CH_4	REQDST_CH_3	REQDST_CH_2	REQDST_CH_1	REQDST_CH_0
	Reset value	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
0x378	SGLREQSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SGLREQSRC_WE_7	SGLREQSRC_WE_6	SGLREQSRC_WE_5	SGLREQSRC_WE_4	SGLREQSRC_WE_3	SGLREQSRC_WE_2	SGLREQSRC_WE_1	SGLREQSRC_WE_0	SGLREQSRC_CH_7	SGLREQSRC_CH_6	SGLREQSRC_CH_5	SGLREQSRC_CH_4	SGLREQSRC_CH_3	SGLREQSRC_CH_2	SGLREQSRC_CH_1	SGLREQSRC_CH_0
	Reset value	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
0x380	SGLREQDST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SGLREQDST_WE_7	SGLREQDST_WE_6	SGLREQDST_WE_5	SGLREQDST_WE_4	SGLREQDST_WE_3	SGLREQDST_WE_2	SGLREQDST_WE_1	SGLREQDST_WE_0	SGLREQDST_CH_7	SGLREQDST_CH_6	SGLREQDST_CH_5	SGLREQDST_CH_4	SGLREQDST_CH_3	SGLREQDST_CH_2	SGLREQDST_CH_1	SGLREQDST_CH_0
	Reset value	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
0x388	LSTREQSRC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LSTREQSRC_WE_7	LSTREQSRC_WE_6	LSTREQSRC_WE_5	LSTREQSRC_WE_4	LSTREQSRC_WE_3	LSTREQSRC_WE_2	LSTREQSRC_WE_1	LSTREQSRC_WE_0	LSTREQSRC_CH_7	LSTREQSRC_CH_6	LSTREQSRC_CH_5	LSTREQSRC_CH_4	LSTREQSRC_CH_3	LSTREQSRC_CH_2	LSTREQSRC_CH_1	LSTREQSRC_CH_0
	Reset value	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
0x390	LSTREQDST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LSTREQDST_WE_7	LSTREQDST_WE_6	LSTREQDST_WE_5	LSTREQDST_WE_4	LSTREQDST_WE_3	LSTREQDST_WE_2	LSTREQDST_WE_1	LSTREQDST_WE_0	LSTREQDST_CH_7	LSTREQDST_CH_6	LSTREQDST_CH_5	LSTREQDST_CH_4	LSTREQDST_CH_3	LSTREQDST_CH_2	LSTREQDST_CH_1	LSTREQDST_CH_0
	Reset value	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
0x398	DMACFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMACFG_EN
	Reset value	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
0x3A0	CHEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CHEN_WE_7	CHEN_WE_6	CHEN_WE_5	CHEN_WE_4	CHEN_WE_3	CHEN_WE_2	CHEN_WE_1	CHEN_WE_0	CHEN_EN_7	CHEN_EN_6	CHEN_EN_5	CHEN_EN_4	CHEN_EN_3	CHEN_EN_2	CHEN_EN_1	CHEN_EN_0
	Reset value	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

11 Nested vectored interrupt controller (NVIC)

11.1 NVIC Main features

NVIC supported 32 maskable interrupts and 4 programmable priority levels.

11.2 Interrupt and exception vectors

Table 29 Vector table

Position	Priority	Type of priority	Handler	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	fixed	Reset	Reset	0x0000_0004
-	-2	fixed	NMI_Handler	FIQ interrupts	0x0000_0008
-	-1	fixed	HardFault_Handler	All class of fault	0x0000_000C
-	-	-	-	Reserved	0x0000_0010 0x0000_0014 0x0000_0018 0x0000_001C 0x0000_0020 0x0000_0024 0x0000_0028
-	3	settable	SVC_Handler	System service	0x0000_002C
-	--	-	-	Reserved	0x0000_0030 0x0000_0034
-	5	settable	PendSV_Handler	Pendable request for system service	0x0000_0038
-	6	settable	SysTick_Handler	System tick timer	0x0000_003C
0	7	settable	DMA_IRQHandler	DMA interrupt	0x0000_0040
1	8	settable	WDTW_IRQHandler	Window watchdog interrupt	0x0000_0044
2	9	settable	COMP1_IRQHandler	Comparator 1 interrupt	0x0000_0048
3	10	settable	RTC_IRQHandler	RTC interrupt	0x0000_004C
4	11	settable	ICTL_IRQHandler	Interrupt controller IRQ interrupt	0x0000_0050
5	12	settable	UART1_IRQHandler	UART1 interrupt	0x0000_0054
6	13	settable	UART2_IRQHandler	UART2 interrupt	0x0000_0058
7	14	settable	UART3_IRQHandler	UART3 interrupt	0x0000_005C
8	15	settable	UART4_IRQHandler	UART4 interrupt	0x0000_0060
9	16	settable	TIM1_BRK_IRQHandler	Advanced timer 1 break interrupt	0x0000_0064
10	17	settable	TIM1_UDT_IRQHandler	Advanced timer 1 update interrupt	0x0000_0068
11	18	settable	TIM1_TRG_COM_IRQHandler	Advanced timer 1 trigger/commutation interrupt	0x0000_006C
12	19	settable	TIM1_CC_IRQHandler	Advanced timer 1 capture/compare interrupt	0x0000_0070
13	20	settable	TIM2_BRK_IRQHandler	Advanced timer 2 break interrupt	0x0000_0074
14	21	settable	TIM2_UDT_IRQHandler	Advanced timer 2 update interrupt	0x0000_0078
15	22	settable	TIM2_TRG_COM_IRQHandler	Advanced timer 2 trigger/commutation interrupt	0x0000_007C
16	23	settable	TIM2_CC_IRQHandler	Advanced timer 2 capture/compare interrupt	0x0000_0080
17	24	settable	TIMG1_UDT_IRQHandler	General timer 1 update interrupt	0x0000_0084
18	25	settable	TIMG1_TRG_COM_IRQHandler	General timer 1 trigger/commutation interrupt	0x0000_0088
19	26	settable	TIMG1_CC_IRQHandler	General timer 1 capture/compare interrupt	0x0000_008C
20	27	settable	TIMG2_UDT_IRQHandler	General timer 2 update interrupt	0x0000_0090
21	28	settable	TIMG2_TRG_COM_IRQHandler	General timer 2 trigger/commutation interrupt	0x0000_0094
22	29	settable	TIMG2_CC_IRQHandler	General timer 2 capture/compare interrupt	0x0000_0098
23	30	settable	TIMG3_UDT_IRQHandler	General timer 3 update interrupt	0x0000_009C
24	31	settable	TIMG3_TRG_COM_IRQHandler	General timer 3 trigger/commutation interrupt	0x0000_00A0
25	32	settable	TIMG3_CC_IRQHandler	General timer 3 capture/compare interrupt	0x0000_00A4
26	33	settable	TIMG4_UDT_IRQHandler	General timer 4 update interrupt	0x0000_00A8
27	34	settable	TIMG4_TRG_COM_IRQHandler	General timer 4 trigger/commutation interrupt	0x0000_00AC

Position	Priority	Type of priority	Handler	Description	Address
28	35	settable	TIMG4_CC_IRQHandler	General timer 4 capture/compare interrupt	0x0000_00B0
29	36	settable	COMP2_IRQHandler	Comparator 2 interrupt	0x0000_00B4
30	37	settable	COMP3_IRQHandler	Comparator 3 interrupt	0x0000_00B8
31	38	settable	COMP4_IRQHandler	Comparator 4 interrupt	0x0000_00BC

12 Interrupt controller (ICTL)

12.1 Introduction

12.2 ICTL main features

- 64 IRQ normal interrupt sources
- 4 FIQ fast interrupt sources
- Software interrupts
- Masking

12.3 ICTL functional description

IRQ and force are active low.

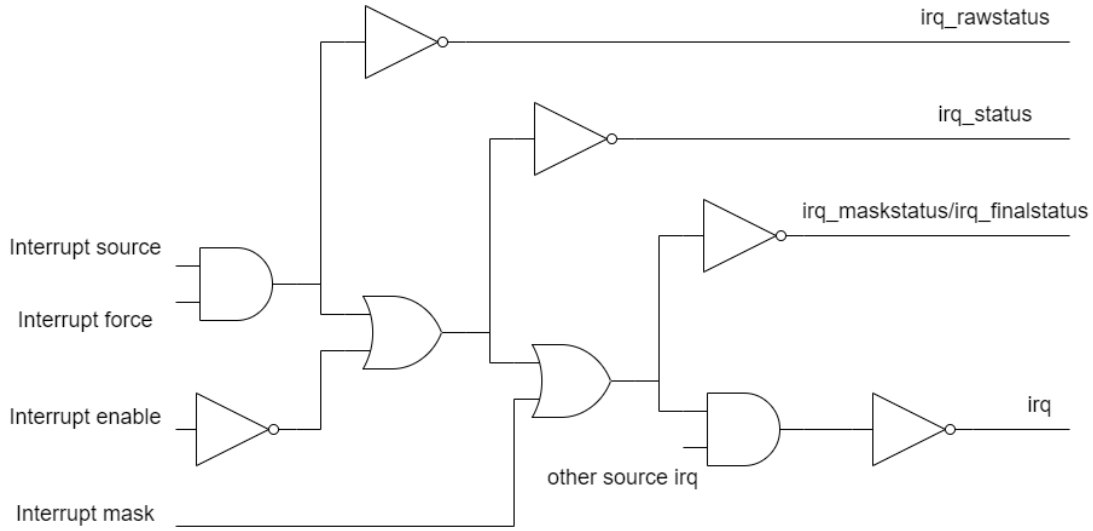


Figure 21 IRQ generation

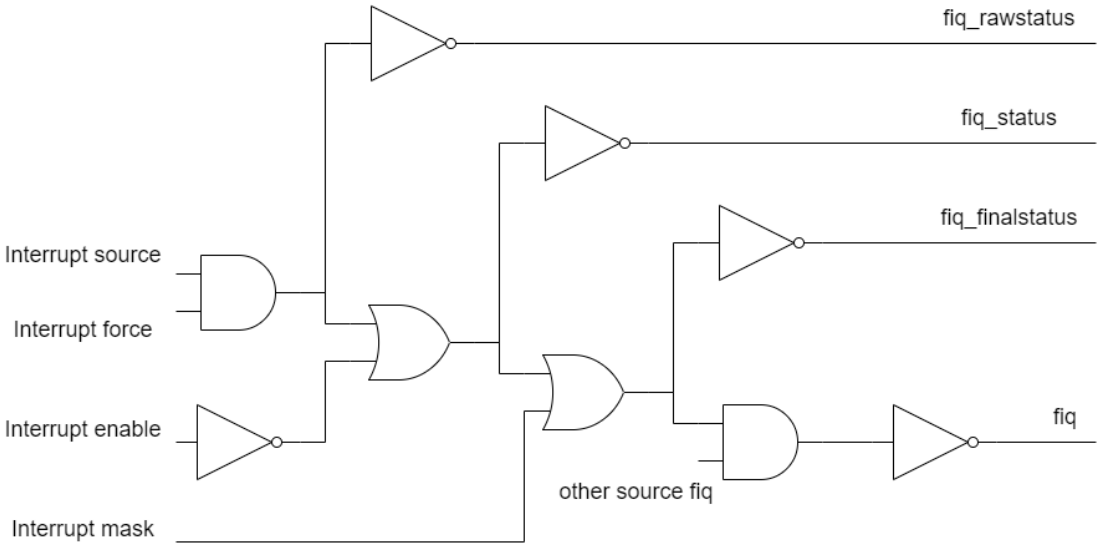


Figure 22 FIQ generation

Domain	#	Interrupt source 1	Interrupt source 2
FIQ_INTC	0	Intr_wdt_w	
FIQ_INTC	1	O_vdet	
FIQ_INTC	2	External_nmi_wakeup Portb<26>	
FIQ_INTC	3		

12.4 ICTL registers

12.4.1 ICTL_IRQ_INTEN_L

Base address: 0x8003_0000
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_EN_CH_x**: IRQ x (x=0 to 31) Interrupt enable 1'b0: disable interrupt 1'b1: enable interrupt	R/W

12.4.2 ICTL_IRQ_INTEN_H

Base address: 0x8003_0000
Address offset: 0x04
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_EN_CH_x** IRQ x (x=32 to 63) Interrupt enable 1'b0: disable interrupt 1'b1: enable interrupt	R/W

12.4.3 ICTL_IRQ_INTMASK_L

Base address: 0x8003_0000
Address offset: 0x08
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_MSK_CH_x**: IRQ x (x=0 to 31) Interrupt mask enable 1'b0: unmask interrupt 1'b1: mask interrupt	R/W

12.4.4 ICTL_IRQ_INTMASK_H

Base address: 0x8003_0000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_MASK_CH_x**: IRQ x (x=32 to 63) Interrupt mask enable 1'b0: unmask interrupt 1'b1: mask interrupt	R/W

12.4.5 ICTL_IRQ_INTFORCE_L

Base address: 0x8003_0000
Address offset: 0x10
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_FRC_CH_x**: IRQ x (x=0 to 31) Interrupt force bit 1'b0: active low 1'b1: active high	R/W

12.4.6 ICTL_IRQ_INTFORCE_H

Base address: 0x8003_0000
Address offset: 0x14
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_FRC_CH_x**: IRQ x (x=32 to 63) Interrupt force bit 1'b0: active low 1'b1: active high	R/W

12.4.7 ICTL_IRQ_RAWSTATUS_L

Base address: 0x8003_0000
Address offset: 0x18
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_RS_CH_x**: IRQ x (x=0 to 31) Interrupt raw status	R/W

12.4.8 ICTL_IRQ_RAWSTATUS_H

Base address: 0x8003_0000
Address offset: 0x1C
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_RS_CH_x**: IRQ x (x=32 to 63) Interrupt raw status	R/W

12.4.9 ICTL_IRQ_STATUS_L

Base address: 0x8003_0000
Address offset: 0x20
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_S_CH_x**: IRQ x (x=0 to 31) Interrupt status	R/W

12.4.10 ICTL_IRQ_STATUS_H

Base address: 0x8003_0000
Address offset: 0x24
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_S_CH_x**: IRQ x (x=32 to 63) Interrupt status	R/W

12.4.11 ICTL_IRQ_MASKSTATUS_L

Base address: 0x8003_0000
Address offset: 0x28
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_MS_CH_x**: IRQ x (x=0 to 31) Interrupt status after the masking stage	R/W

12.4.12 ICTL_IRQ_MASKSTATUS_H

Base address: 0x8003_0000
Address offset: 0x2C
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_MS_CH_x**: IRQ x (x=32 to 63) Interrupt status after the masking stage	R/W

12.4.13 ICTL_IRQ_FINALSTATUS_L

Base address: 0x8003_0000
Address offset: 0x30
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_FS_CH_x**: IRQ x (x=0 to 31) Interrupt final status	R/W

12.4.14 ICTL_IRQ_FINALSTATUS_H

Base address: 0x8003_0000
Address offset: 0x34
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**ICTL_IRQ_FS_CH_x**: IRQ x (x=32 to 63) Interrupt final status	R/W

12.4.15 ICTL_FIQ_INTEN

Base address: 0x8003_0000
Address offset: 0xC0

Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_FIQ_EN_CH_x**: FIQ x (x=0 to 3) Interrupt enable 1'b0: disable interrupt 1'b1: enable interrupt	R/W

12.4.16 ICTL_FIQ_INTMASK

Base address: 0x8003_0000

Address offset: 0xC4

Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_FIQ_MSK_CH_x**: FIQ x (x=0 to 3) Interrupt mask enable 1'b0: unmask interrupt 1'b1: mask interrupt	R/W

12.4.17 ICTL_FIQ_INTFORCE

Base address: 0x8003_0000

Address offset: 0xC8

Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_FIQ_FRC_CH_x**: FIQ x (x=0 to 3) Interrupt force bit 1'b0: active low 1'b1: active high	R/W

12.4.18 ICTL_FIQ_RAWSTATUS

Base address: 0x8003_0000
Address offset: 0xCC
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_FIQ_RS_CH_x**: FIQ x (x=0 to 3) fast interrupt raw status	R/W

12.4.19 ICTL_FIQ_STATUS

Base address: 0x8003_0000
Address offset: 0xD0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_FIQ_S_CH_x**: FIQ x (x=0 to 3) fast interrupt status	R/W

12.4.20 ICTL_FIQ_FINALSTATUS

Base address: 0x8003_0000
Address offset: 0xD4
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_FIQ_FS_CH_x**: FIQ x (x=0 to 3) fast interrupt final status	R/W

12.4.21 ICTL_IRQ_PLEVEL

Base address: 0x8003_0000
Address offset: 0xD8
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3:0>	**ICTL_IRQ_PLEVEL[3:0]**: IRQ Interrupt controller system priority level	R/W

12.4.22 ICTL register map

The base address of the ICTL registers is 0x8003_0000.
The following table gives the ICTL register map and reset values.

Table 30 ICTL register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	IRQ_INTEN_L	ICTL_IRQ_EN_CH_31	ICTL_IRQ_EN_CH_30	ICTL_IRQ_EN_CH_29	ICTL_IRQ_EN_CH_28	ICTL_IRQ_EN_CH_27	ICTL_IRQ_EN_CH_26	ICTL_IRQ_EN_CH_25	ICTL_IRQ_EN_CH_24	ICTL_IRQ_EN_CH_23	ICTL_IRQ_EN_CH_22	ICTL_IRQ_EN_CH_21	ICTL_IRQ_EN_CH_20	ICTL_IRQ_EN_CH_20	ICTL_IRQ_EN_CH_18	ICTL_IRQ_EN_CH_17	ICTL_IRQ_EN_CH_16	ICTL_IRQ_EN_CH_15	ICTL_IRQ_EN_CH_14	ICTL_IRQ_EN_CH_13	ICTL_IRQ_EN_CH_12	ICTL_IRQ_EN_CH_11	ICTL_IRQ_EN_CH_10	ICTL_IRQ_EN_CH_9	ICTL_IRQ_EN_CH_8	ICTL_IRQ_EN_CH_7	ICTL_IRQ_EN_CH_6	ICTL_IRQ_EN_CH_5	ICTL_IRQ_EN_CH_4	ICTL_IRQ_EN_CH_3	ICTL_IRQ_EN_CH_2	ICTL_IRQ_EN_CH_1	ICTL_IRQ_EN_CH_0
	Reset value	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
0x04	IRQ_INTEN_H	ICTL_IRQ_EN_CH_63	ICTL_IRQ_EN_CH_62	ICTL_IRQ_EN_CH_61	ICTL_IRQ_EN_CH_60	ICTL_IRQ_EN_CH_59	ICTL_IRQ_EN_CH_58	ICTL_IRQ_EN_CH_57	ICTL_IRQ_EN_CH_56	ICTL_IRQ_EN_CH_55	ICTL_IRQ_EN_CH_54	ICTL_IRQ_EN_CH_53	ICTL_IRQ_EN_CH_52	ICTL_IRQ_EN_CH_51	ICTL_IRQ_EN_CH_50	ICTL_IRQ_EN_CH_49	ICTL_IRQ_EN_CH_48	ICTL_IRQ_EN_CH_47	ICTL_IRQ_EN_CH_46	ICTL_IRQ_EN_CH_45	ICTL_IRQ_EN_CH_44	ICTL_IRQ_EN_CH_43	ICTL_IRQ_EN_CH_42	ICTL_IRQ_EN_CH_41	ICTL_IRQ_EN_CH_40	ICTL_IRQ_EN_CH_39	ICTL_IRQ_EN_CH_38	ICTL_IRQ_EN_CH_37	ICTL_IRQ_EN_CH_36	ICTL_IRQ_EN_CH_35	ICTL_IRQ_EN_CH_34	ICTL_IRQ_EN_CH_33	ICTL_IRQ_EN_CH_32
	Reset value	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
0x08	IRQ_INTMASK_L	ICTL_IRQ_MSK_CH_31	ICTL_IRQ_MSK_CH_30	ICTL_IRQ_MSK_CH_29	ICTL_IRQ_MSK_CH_28	ICTL_IRQ_MSK_CH_27	ICTL_IRQ_MSK_CH_26	ICTL_IRQ_MSK_CH_25	ICTL_IRQ_MSK_CH_24	ICTL_IRQ_MSK_CH_23	ICTL_IRQ_MSK_CH_22	ICTL_IRQ_MSK_CH_21	ICTL_IRQ_MSK_CH_20	ICTL_IRQ_MSK_CH_20	ICTL_IRQ_MSK_CH_18	ICTL_IRQ_MSK_CH_17	ICTL_IRQ_MSK_CH_16	ICTL_IRQ_MSK_CH_15	ICTL_IRQ_MSK_CH_14	ICTL_IRQ_MSK_CH_13	ICTL_IRQ_MSK_CH_12	ICTL_IRQ_MSK_CH_11	ICTL_IRQ_MSK_CH_10	ICTL_IRQ_MSK_CH_9	ICTL_IRQ_MSK_CH_8	ICTL_IRQ_MSK_CH_7	ICTL_IRQ_MSK_CH_6	ICTL_IRQ_MSK_CH_5	ICTL_IRQ_MSK_CH_4	ICTL_IRQ_MSK_CH_3	ICTL_IRQ_MSK_CH_2	ICTL_IRQ_MSK_CH_1	ICTL_IRQ_MSK_CH_0
	Reset value	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
0x0C	IRQ_INTMASK_H	ICTL_IRQ_MSK_CH_63	ICTL_IRQ_MSK_CH_62	ICTL_IRQ_MSK_CH_61	ICTL_IRQ_MSK_CH_60	ICTL_IRQ_MSK_CH_59	ICTL_IRQ_MSK_CH_58	ICTL_IRQ_MSK_CH_57	ICTL_IRQ_MSK_CH_56	ICTL_IRQ_MSK_CH_55	ICTL_IRQ_MSK_CH_54	ICTL_IRQ_MSK_CH_53	ICTL_IRQ_MSK_CH_52	ICTL_IRQ_MSK_CH_51	ICTL_IRQ_MSK_CH_50	ICTL_IRQ_MSK_CH_49	ICTL_IRQ_MSK_CH_48	ICTL_IRQ_MSK_CH_47	ICTL_IRQ_MSK_CH_46	ICTL_IRQ_MSK_CH_45	ICTL_IRQ_MSK_CH_44	ICTL_IRQ_MSK_CH_43	ICTL_IRQ_MSK_CH_42	ICTL_IRQ_MSK_CH_41	ICTL_IRQ_MSK_CH_40	ICTL_IRQ_MSK_CH_39	ICTL_IRQ_MSK_CH_38	ICTL_IRQ_MSK_CH_37	ICTL_IRQ_MSK_CH_36	ICTL_IRQ_MSK_CH_35	ICTL_IRQ_MSK_CH_34	ICTL_IRQ_MSK_CH_33	ICTL_IRQ_MSK_CH_32
	Reset value	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
0x10	IRQ_INTFORCE_L	ICTL_IRQ_FRC_CH_31	ICTL_IRQ_FRC_CH_30	ICTL_IRQ_FRC_CH_29	ICTL_IRQ_FRC_CH_28	ICTL_IRQ_FRC_CH_27	ICTL_IRQ_FRC_CH_26	ICTL_IRQ_FRC_CH_25	ICTL_IRQ_FRC_CH_24	ICTL_IRQ_FRC_CH_23	ICTL_IRQ_FRC_CH_22	ICTL_IRQ_FRC_CH_21	ICTL_IRQ_FRC_CH_20	ICTL_IRQ_FRC_CH_20	ICTL_IRQ_FRC_CH_18	ICTL_IRQ_FRC_CH_17	ICTL_IRQ_FRC_CH_16	ICTL_IRQ_FRC_CH_15	ICTL_IRQ_FRC_CH_14	ICTL_IRQ_FRC_CH_13	ICTL_IRQ_FRC_CH_12	ICTL_IRQ_FRC_CH_11	ICTL_IRQ_FRC_CH_10	ICTL_IRQ_FRC_CH_9	ICTL_IRQ_FRC_CH_8	ICTL_IRQ_FRC_CH_7	ICTL_IRQ_FRC_CH_6	ICTL_IRQ_FRC_CH_5	ICTL_IRQ_FRC_CH_4	ICTL_IRQ_FRC_CH_3	ICTL_IRQ_FRC_CH_2	ICTL_IRQ_FRC_CH_1	ICTL_IRQ_FRC_CH_0
	Reset value	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
0x14	IRQ_INTFORCE_H	ICTL_IRQ_FRC_CH_63	ICTL_IRQ_FRC_CH_62	ICTL_IRQ_FRC_CH_61	ICTL_IRQ_FRC_CH_60	ICTL_IRQ_FRC_CH_59	ICTL_IRQ_FRC_CH_58	ICTL_IRQ_FRC_CH_57	ICTL_IRQ_FRC_CH_56	ICTL_IRQ_FRC_CH_55	ICTL_IRQ_FRC_CH_54	ICTL_IRQ_FRC_CH_53	ICTL_IRQ_FRC_CH_52	ICTL_IRQ_FRC_CH_51	ICTL_IRQ_FRC_CH_50	ICTL_IRQ_FRC_CH_49	ICTL_IRQ_FRC_CH_48	ICTL_IRQ_FRC_CH_47	ICTL_IRQ_FRC_CH_46	ICTL_IRQ_FRC_CH_45	ICTL_IRQ_FRC_CH_44	ICTL_IRQ_FRC_CH_43	ICTL_IRQ_FRC_CH_42	ICTL_IRQ_FRC_CH_41	ICTL_IRQ_FRC_CH_40	ICTL_IRQ_FRC_CH_39	ICTL_IRQ_FRC_CH_38	ICTL_IRQ_FRC_CH_37	ICTL_IRQ_FRC_CH_36	ICTL_IRQ_FRC_CH_35	ICTL_IRQ_FRC_CH_34	ICTL_IRQ_FRC_CH_33	ICTL_IRQ_FRC_CH_32
	Reset value	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
0x18	IRQ_RSTAT_L	ICTL_IRQ_RS_CH_31	ICTL_IRQ_RS_CH_30	ICTL_IRQ_RS_CH_29	ICTL_IRQ_RS_CH_28	ICTL_IRQ_RS_CH_27	ICTL_IRQ_RS_CH_26	ICTL_IRQ_RS_CH_25	ICTL_IRQ_RS_CH_24	ICTL_IRQ_RS_CH_23	ICTL_IRQ_RS_CH_22	ICTL_IRQ_RS_CH_21	ICTL_IRQ_RS_CH_20	ICTL_IRQ_RS_CH_20	ICTL_IRQ_RS_CH_18	ICTL_IRQ_RS_CH_17	ICTL_IRQ_RS_CH_16	ICTL_IRQ_RS_CH_15	ICTL_IRQ_RS_CH_14	ICTL_IRQ_RS_CH_13	ICTL_IRQ_RS_CH_12	ICTL_IRQ_RS_CH_11	ICTL_IRQ_RS_CH_10	ICTL_IRQ_RS_CH_9	ICTL_IRQ_RS_CH_8	ICTL_IRQ_RS_CH_7	ICTL_IRQ_RS_CH_6	ICTL_IRQ_RS_CH_5	ICTL_IRQ_RS_CH_4	ICTL_IRQ_RS_CH_3	ICTL_IRQ_RS_CH_2	ICTL_IRQ_RS_CH_1	ICTL_IRQ_RS_CH_0
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C	IRQ_RSTAT_H	ICTL_IRQ_RS_CH_63	ICTL_IRQ_RS_CH_62	ICTL_IRQ_RS_CH_61	ICTL_IRQ_RS_CH_60	ICTL_IRQ_RS_CH_59	ICTL_IRQ_RS_CH_58	ICTL_IRQ_RS_CH_57	ICTL_IRQ_RS_CH_56	ICTL_IRQ_RS_CH_55	ICTL_IRQ_RS_CH_54	ICTL_IRQ_RS_CH_53	ICTL_IRQ_RS_CH_52	ICTL_IRQ_RS_CH_51	ICTL_IRQ_RS_CH_50	ICTL_IRQ_RS_CH_49	ICTL_IRQ_RS_CH_48	ICTL_IRQ_RS_CH_47	ICTL_IRQ_RS_CH_46	ICTL_IRQ_RS_CH_45	ICTL_IRQ_RS_CH_44	ICTL_IRQ_RS_CH_43	ICTL_IRQ_RS_CH_42	ICTL_IRQ_RS_CH_41	ICTL_IRQ_RS_CH_40	ICTL_IRQ_RS_CH_39	ICTL_IRQ_RS_CH_38	ICTL_IRQ_RS_CH_37	ICTL_IRQ_RS_CH_36	ICTL_IRQ_RS_CH_35	ICTL_IRQ_RS_CH_34	ICTL_IRQ_RS_CH_33	ICTL_IRQ_RS_CH_32
	Reset value	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
0x20	IRQ_STAT_L	ICTL_IRQ_S_CH_31	ICTL_IRQ_S_CH_30	ICTL_IRQ_S_CH_29	ICTL_IRQ_S_CH_28	ICTL_IRQ_S_CH_27	ICTL_IRQ_S_CH_26	ICTL_IRQ_S_CH_25	ICTL_IRQ_S_CH_24	ICTL_IRQ_S_CH_23	ICTL_IRQ_S_CH_22	ICTL_IRQ_S_CH_21	ICTL_IRQ_S_CH_20	ICTL_IRQ_S_CH_20	ICTL_IRQ_S_CH_18	ICTL_IRQ_S_CH_17	ICTL_IRQ_S_CH_16	ICTL_IRQ_S_CH_15	ICTL_IRQ_S_CH_14	ICTL_IRQ_S_CH_13	ICTL_IRQ_S_CH_12	ICTL_IRQ_S_CH_11	ICTL_IRQ_S_CH_10	ICTL_IRQ_S_CH_9	ICTL_IRQ_S_CH_8	ICTL_IRQ_S_CH_7	ICTL_IRQ_S_CH_6	ICTL_IRQ_S_CH_5	ICTL_IRQ_S_CH_4	ICTL_IRQ_S_CH_3	ICTL_IRQ_S_CH_2	ICTL_IRQ_S_CH_1	ICTL_IRQ_S_CH_0
	Reset value	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
0x24	IRQ_STAT_H	ICTL_IRQ_S_CH_63	ICTL_IRQ_S_CH_62	ICTL_IRQ_S_CH_61	ICTL_IRQ_S_CH_60	ICTL_IRQ_S_CH_59	ICTL_IRQ_S_CH_58	ICTL_IRQ_S_CH_57	ICTL_IRQ_S_CH_56	ICTL_IRQ_S_CH_55	ICTL_IRQ_S_CH_54	ICTL_IRQ_S_CH_53	ICTL_IRQ_S_CH_52	ICTL_IRQ_S_CH_51	ICTL_IRQ_S_CH_50	ICTL_IRQ_S_CH_49	ICTL_IRQ_S_CH_48	ICTL_IRQ_S_CH_47	ICTL_IRQ_S_CH_46	ICTL_IRQ_S_CH_45	ICTL_IRQ_S_CH_44	ICTL_IRQ_S_CH_43	ICTL_IRQ_S_CH_42	ICTL_IRQ_S_CH_41	ICTL_IRQ_S_CH_40	ICTL_IRQ_S_CH_39	ICTL_IRQ_S_CH_38	ICTL_IRQ_S_CH_37	ICTL_IRQ_S_CH_36	ICTL_IRQ_S_CH_35	ICTL_IRQ_S_CH_34	ICTL_IRQ_S_CH_33	ICTL_IRQ_S_CH_32
	Reset value	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
0x28	IRQ_MASKSTAT_L	ICTL_IRQ_MS_CH_31	ICTL_IRQ_MS_CH_30	ICTL_IRQ_MS_CH_29	ICTL_IRQ_MS_CH_28	ICTL_IRQ_MS_CH_27	ICTL_IRQ_MS_CH_26	ICTL_IRQ_MS_CH_25	ICTL_IRQ_MS_CH_24	ICTL_IRQ_MS_CH_23	ICTL_IRQ_MS_CH_22	ICTL_IRQ_MS_CH_21	ICTL_IRQ_MS_CH_20	ICTL_IRQ_MS_CH_20	ICTL_IRQ_MS_CH_18	ICTL_IRQ_MS_CH_17	ICTL_IRQ_MS_CH_16	ICTL_IRQ_MS_CH_15	ICTL_IRQ_MS_CH_14	ICTL_IRQ_MS_CH_13	ICTL_IRQ_MS_CH_12	ICTL_IRQ_MS_CH_11	ICTL_IRQ_MS_CH_10	ICTL_IRQ_MS_CH_9	ICTL_IRQ_MS_CH_8	ICTL_IRQ_MS_CH_7	ICTL_IRQ_MS_CH_6	ICTL_IRQ_MS_CH_5	ICTL_IRQ_MS_CH_4	ICTL_IRQ_MS_CH_3	ICTL_IRQ_MS_CH_2	ICTL_IRQ_MS_CH_1	ICTL_IRQ_MS_CH_0
	Reset value	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
0x2C	IRQ_MASKSTAT_H	ICTL_IRQ_MS_CH_63	ICTL_IRQ_MS_CH_62	ICTL_IRQ_MS_CH_61	ICTL_IRQ_MS_CH_60	ICTL_IRQ_MS_CH_59	ICTL_IRQ_MS_CH_58	ICTL_IRQ_MS_CH_57	ICTL_IRQ_MS_CH_56	ICTL_IRQ_MS_CH_55	ICTL_IRQ_MS_CH_54	ICTL_IRQ_MS_CH_53	ICTL_IRQ_MS_CH_52	ICTL_IRQ_MS_CH_51	ICTL_IRQ_MS_CH_50	ICTL_IRQ_MS_CH_49	ICTL_IRQ_MS_CH_48	ICTL_IRQ_MS_CH_47	ICTL_IRQ_MS_CH_46	ICTL_IRQ_MS_CH_45	ICTL_IRQ_MS_CH_44	ICTL_IRQ_MS_CH_43	ICTL_IRQ_MS_CH_42	ICTL_IRQ_MS_CH_41	ICTL_IRQ_MS_CH_40	ICTL_IRQ_MS_CH_39	ICTL_IRQ_MS_CH_38	ICTL_IRQ_MS_CH_37	ICTL_IRQ_MS_CH_36	ICTL_IRQ_MS_CH_35	ICTL_IRQ_MS_CH_34	ICTL_IRQ_MS_CH_33	ICTL_IRQ_MS_CH_32
	Reset value	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
0x30	IRQ_FINALSTAT_L	ICTL_IRQ_FS_CH_31	ICTL_IRQ_FS_CH_30	ICTL_IRQ_FS_CH_29	ICTL_IRQ_FS_CH_28	ICTL_IRQ_FS_CH_27	ICTL_IRQ_FS_CH_26	ICTL_IRQ_FS_CH_25	ICTL_IRQ_FS_CH_24	ICTL_IRQ_FS_CH_23	ICTL_IRQ_FS_CH_22	ICTL_IRQ_FS_CH_21	ICTL_IRQ_FS_CH_20	ICTL_IRQ_FS_CH_20	ICTL_IRQ_FS_CH_18	ICTL_IRQ_FS_CH_17	ICTL_IRQ_FS_CH_16	ICTL_IRQ_FS_CH_15	ICTL_IRQ_FS_CH_14	ICTL_IRQ_FS_CH_13	ICTL_IRQ_FS_CH_12	ICTL_IRQ_FS_CH_11	ICTL_IRQ_FS_CH_10	ICTL_IRQ_FS_CH_9	ICTL_IRQ_FS_CH_8	ICTL_IRQ_FS_CH_7	ICTL_IRQ_FS_CH_6	ICTL_IRQ_FS_CH_5	ICTL_IRQ_FS_CH_4	ICTL_IRQ_FS_CH_3	ICTL_IRQ_FS_CH_2	ICTL_IRQ_FS_CH_1	ICTL_IRQ_FS_CH_0
	Reset value	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
0x34	IRQ_FINALSTAT_H	ICTL_IRQ_FS_CH_63	ICTL_IRQ_FS_CH_62	ICTL_IRQ_FS_CH_61	ICTL_IRQ_FS_CH_60	ICTL_IRQ_FS_CH_59	ICTL_IRQ_FS_CH_58	ICTL_IRQ_FS_CH_57	ICTL_IRQ_FS_CH_56	ICTL_IRQ_FS_CH_55	ICTL_IRQ_FS_CH_54	ICTL_IRQ_FS_CH_53	ICTL_IRQ_FS_CH_52	ICTL_IRQ_FS_CH_51	ICTL_IRQ_FS_CH_50	ICTL_IRQ_FS_CH_49	ICTL_IRQ_FS_CH_48	ICTL_IRQ_FS_CH_47	ICTL_IRQ_FS_CH_46	ICTL_IRQ_FS_CH_45	ICTL_IRQ_FS_CH_44	ICTL_IRQ_FS_CH_43	ICTL_IRQ_FS_CH_42	ICTL_IRQ_FS_CH_41	ICTL_IRQ_FS_CH_40	ICTL_IRQ_FS_CH_39	ICTL_IRQ_FS_CH_38	ICTL_IRQ_FS_CH_37	ICTL_IRQ_FS_CH_36	ICTL_IRQ_FS_CH_35	ICTL_IRQ_FS_CH_34	ICTL_IRQ_FS_CH_33	ICTL_IRQ_FS_CH_32
	Reset value	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
0xC0	FIQ_INTEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ICTL_FIQ_EN_CH_3	ICTL_FIQ_EN_CH_2	ICTL_FIQ_EN_CH_1	ICTL_FIQ_EN_CH_0
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC4	FIQ_INTMASK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ICTL_FIQ_MSK_CH_3	ICTL_FIQ_MSK_CH_2	ICTL_FIQ_MSK_CH_1	ICTL_FIQ_MSK_CH_0
	Reset value	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
0xC8	FIQ_INTFORCE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ICTL_FIQ_FRC_CH_3	ICTL_FIQ_FRC_CH_2	ICTL_FIQ_FRC_CH_1	ICTL_FIQ_FRC_CH_0
	Reset value	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
0xCC	FIQ_RSTAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ICTL_FIQ_RS_CH_3	ICTL_FIQ_RS_CH_2	ICTL_FIQ_RS_CH_1	ICTL_FIQ_RS_CH_0
	Reset value	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
0xD0	FIQ_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ICTL_FIQ_S_CH_3	ICTL_FIQ_S_CH_2	ICTL_FIQ_S_CH_1	ICTL_FIQ_S_CH_0
	Reset value	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
0xD4	FIQ_FINALSTAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ICTL_FIQ_FS_CH_3	ICTL_FIQ_FS_CH_2	ICTL_FIQ_FS_CH_1	ICTL_FIQ_FS_CH_0
	Reset value	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
0xD8	IRQ_PLEVEL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IICTL_RQ_PLEVEL[3:0]			
	Reset value	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

13 Cyclic redundancy check calculation (CRC)

13.1 Introduction

The CRC (Cyclic Redundancy Check) algorithm generates a checksum, or a short fixed-length value, based on the data being transmitted.

The CRC algorithm works by treating the data being transmitted as a polynomial and dividing it by a predetermined polynomial known as the generator polynomial. The remainder of this division operation is then used as the checksum. The receiver performs the same division operation using the received data and the same generator polynomial and compares the resulting remainder with the received checksum. If they match, the data is assumed to be error-free, otherwise, the data is considered to be corrupted.

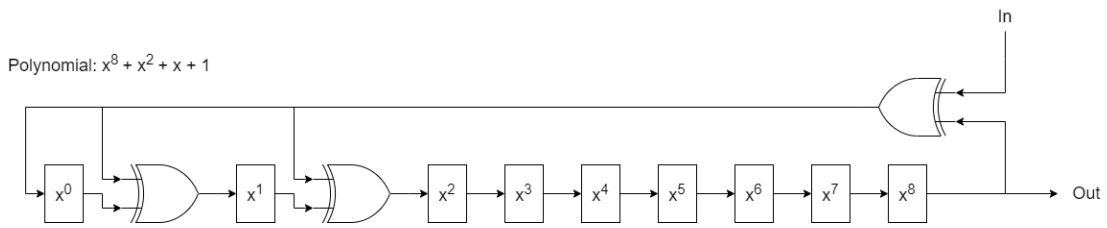


Figure 23 Architectural representation of CRC-8

13.2 CRC main features

CRC module supports below four polynomials:

1. CRC32 (MSB 0x04C11DB7)
2. CRC16 (MSB 0x1021)
3. CRC16 (MSB 0x8005)
4. CRC8 (MSB 0x07)

Input and output data can be reflected. Output data may choose XOR mask.

13.3 CRC functional description

CRC calculation performs the following steps:

Write to CRC_CFG.POLY to select polynomial.
Write to CRC_RESULT to set initial value.
Write to CRC_CFG.REFLECT to set input and output data reflect.
Write to CRC_CFG.XOREN and CRC_XOR to enable and set output XOR mask.
Write input data to CRC_DATA. If multiple date calculates CRC, repeat step 5.
Read CRC result from CRC_RESULT.

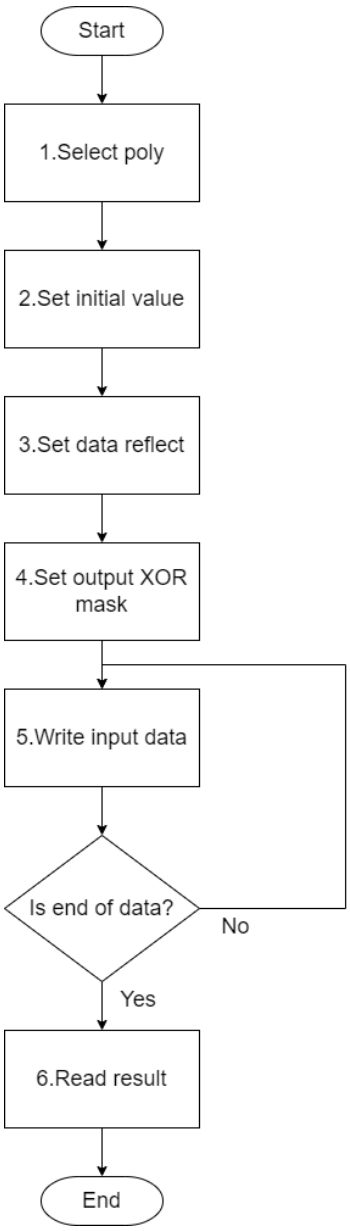


Figure 24 CRC calculation

13.4 CRC registers

13.4.1 CRC_CFG

Base address: 0x8005_0000
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:6>	Reserved	R/W
<5:4>	**CFG_POLY[1:0]**: CRC Polynomial selection 2'b00: CRC32 (0x41c11db7) 2'b01: CRC8 (0x07) 2'b10: CRC16 (0x1021) 2'b11: CRC16 (0x8005)	R/W
<3:2>	Reserved	R/W
<1>	**CFG_XOR_EN**: XOR the CRC value with the XOR mask in 0x0C register 1'b0: Non-XOR output 1'b1: XOR output	R/W
<0>	**CFG_REFLECT**: Reflect Input or output 1'b0: Don't Reflect Input and Output, normal 1'b1: Reflected Input, and Reflected Output	R/W

13.4.2 CRC_DATA

Base address : 0x8005_0000
Address offset: 0x04
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**DATA_SRC_DATA_3[7:0]**: CRC Source Data, Byte 3	R/W
<23:16>	**DATA_SRC_DATA_2[7:0]**: CRC Source Data, Byte 2	R/W
<15:8>	**DATA_SRC_DATA_1[7:0]**: CRC Source Data, Byte 1	R/W
<7:0>	**DATA_SRC_DATA_0[7:0]**: CRC Source Data, Byte 0	R/W

13.4.3 CRC_RESULT

Base address : 0x8005_0000
Address offset: 0x08
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**RESULT_RES_INIT_3[7:0]**: CRC Results or CRC Initial Value, Byte 3.	R/W
<23:16>	**RESULT_RES_INIT_2[7:0]**: CRC Results or CRC Initial Value, Byte 2.	R/W
<15:8>	**RESULT_RES_INIT_1[7:0]**: CRC Results or CRC Initial Value, Byte 1.	R/W
<7:0>	**RESULT_RES_INIT_0[7:0]**: CRC Results or CRC Initial Value, Byte 0.	R/W

13.4.4 CRC_XOR_VALUE

Base address : 0x8005_0000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**XOR_MSK_3[7:0]**: CRC Results XOR Mask, Byte 3.	R/W
<23:16>	**XOR_MSK_2[7:0]**: CRC Results XOR Mask, Byte 2.	R/W
<15:8>	**XOR_MSK_1[7:0]**: CRC Results XOR Mask, Byte 1.	R/W
<7:0>	**XOR_MSK_0[7:0]**: CRC Results XOR Mask, Byte 0.	R/W

13.4.5 CRC register map

The base address of the CRC registers is 0x8005_0000.
The following table gives the CRC register map and reset values.

Table 31 CRC register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_CFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CFG_POLY[1:0]		Res	Res	CFG_XOR_EN	CFG_REFLECT
	Reset value	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
0x04	CRC_DATA	SRC_DATA_3[7:0]								SRC_DATA_2[7:0]								SRC_DATA_1[7:0]								SRC_DATA_0[7:0]							
	Reset value	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
0x08	CRC_RESULT	RES_INIT_3[7:0]								RES_INIT_2[7:0]								RES_INIT_1[7:0]								RES_INIT_0[7:0]							
	Reset value	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
0x0C	CRC_XOR_VALUE	XOR_MSK_3[7:0]								XOR_MSK_2[7:0]								XOR_MSK_1[7:0]								XOR_MSK_0[7:0]							
	Reset value	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

14 Motor Driver Unit (MDU)

14.1 Introduction

XT32H serial motor driver unit includes ADC, DAC, Comparator, PGA. The architecture of Main analog blocks is shown in Figure 25. XT32H motor driver unit and advanced timer allow microcontroller (MCU) to control the speed and direction of electric motors.

14.2 MDU main features

- Configuration ADC conversion sequence
- Multiple ADC conversion trigger
- Support DMA transfer of ADC conversion results

14.3 MDU signal descriptions

The MDU signals are shown in the following table.

Table 32 MDU signal descriptions

Signal	Description	I/O
ADC_CH1_P	PGA1 non-inverting input	I
ADC_CH1_N	PGA1 inverting input	I
ADC_CH2_P	PGA2 non-inverting input	I
ADC_CH2_N	PGA2 inverting input	I
ADC_CH3_P	PGA3 non-inverting input	I
ADC_CH3_N	PGA3 inverting input	I
ADC_CH4_P	PGA4 non-inverting input	I
ADC_CH4_N	PGA5 inverting input	I
ADC_CH5	ADC channel source	I
ADC_CH6	ADC channel source	I
ADC_CH7	ADC channel source	I
ADC_CH8	ADC channel source	I
ADC_CH9	ADC channel source	I
ADC_CH10	ADC channel source	I
ADC_CH11	ADC channel source	I
ADC_CH12	ADC channel source	I
ADC_CH13	ADC channel source	I
ADC_CH14	ADC channel source	I
ADC_CH15	ADC channel source	I

ADC_CH16	ADC channel source	I
ADC_CH17	ADC channel source	I
ADC_CH18	ADC channel source	I
ADC_CH19	ADC channel source	I
ADC_CH24	ADC channel source	I
ADC_CH25	ADC channel source	I
ADC_CH26	ADC channel source	I
ADC_CH27	ADC channel source	I
ADC_CH28	ADC channel source	I
ADC_CH29	ADC channel source	I
ADC_CH30	ADC channel source	I
ADC_CH31	ADC channel source	I
ADC_CH32	ADC channel source	I
ADC_CH33	ADC channel source	I
ADC_CH34	ADC channel source	I
ADC_CH35	ADC channel source	I
ADC_CH36	ADC channel source	I
ADC_CH37	ADC channel source	I
ADC_CH38	ADC channel source	I
ADC_CH39	ADC channel source	I
ADC_CH40	ADC channel source	I
ADC_CH41	ADC channel source	I
ADC_CH42	ADC channel source	I
ADC_CH43	ADC channel source	I
ADC_CH44	ADC channel source	I
COMP1_P	Comparator 1 non-inverting input	I
COMP1_N	Comparator 1 inverting input	I
COMP2_P	Comparator 2 non-inverting input	I
COMP2_N	Comparator 2 inverting input	I
COMP3_P	Comparator 3 non-inverting input	I
COMP3_N	Comparator 3 inverting input	I
COMP4_P	Comparator 4 non-inverting input	I
COMP4_N	Comparator 4 inverting input	I

14.4 MDU functional description

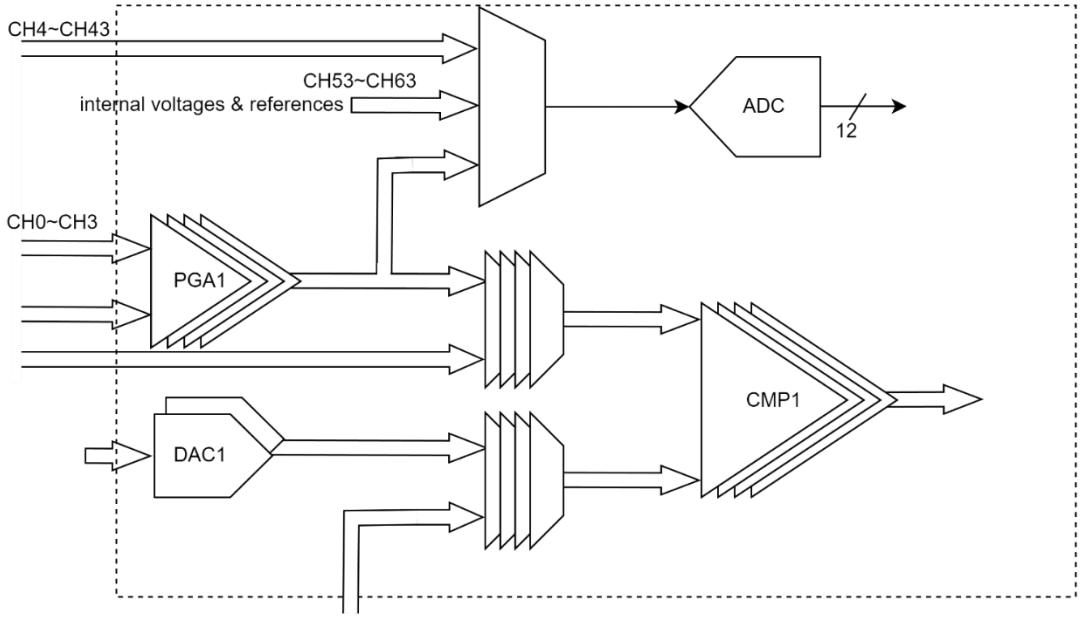


Figure 25 MDU block

14.4.1 ADC

14.4.1.1 Introduction

The 12-bit ADC is a successive approximation analog-to-digital converter.
It has up to 64 input channels which include 4 PGA channels for external differential signals, 40 channels for external single-ended signals, and 10 internal channels for internal signals and references measurements. A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored in a left-aligned or right-aligned 16-bit data register. The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined upper or lower thresholds.
[block diagram]

14.4.1.2 ADC output characteristics

The ADC output code (without calibration) can be calculated as

$$D_{out} = \left(V_{in} - \frac{vdd}{2}\right) \cdot \frac{Gain_{ADC}}{2.2} \cdot 4096 + 2048 \qquad \text{(Equation 14-1)}$$

where D_{out} is ADC raw output code, V_{in} is ADC input voltage, vdd is the chip supply voltage, $Gain_{ADC}$ is the gain factor of ADC.
From Equation 14-1， the ADC output code is linearly related to ADC input voltage.

Table 14-2 lists the $Gain_{ADC}$ values implemented in the XT32H series and their targeted supply voltage ranges.

Table 14-2 ADC supply voltage and suggested Gain factors

vdd	≤ 5.565V	≤ 5.00V	≤ 3.30V	≤ 2.50V
$Gain_{ADC}$	0.3953	0.4400	0.6667	0.8799

When the supply voltage is exactly at 5.565/5/3.3/2.5V together with its corresponding $Gain_{ADC}$ setting, Equation 14-1 can be simplified to

$$D_{out} = \frac{V_{in}}{vdd} \cdot 4096 \qquad \text{(Equation 14-2)}$$

The ADC output code becomes perfectly proportional to the input voltage without any offset.

The ADC conversion after calibration can also achieve an ADC output proportional to the input voltage, as Equation 14-2. Details of the calibration process is shown in Section 1.4.1.7.

14.4.1.3 ADC groups and channels definitions

XT32H supports multiple ADC functional modes/operation. The following defines the terminology and concepts related to the ADC:

1. ADC ports – 44 ADC input ports. The ADC pin channels are mapped to ADC channels by setting COMP_CFG2.PGAx_xSrc[1:0], refer to table 12.
2. ADC channels – 64 ADC working channels. Four channels are synchronized sampling channels. The others are single-ended channels. Among the 49 channels, 39 channels support mapping from external inputs to the ADC channels. The other 10 channels map the internal inputs directly to the ADC channels. Please refer to table 13 for details. The ADC channels are defined in Group channels by setting INJGRP_CHCR_x. ADCCH_ID_x[5:0] and REGGRP_CHCR_x.ADCCH_ID_x[7:0].
3. Group channels – ADC conversion group channels. There are 4 injection channels and 32 regular channels. The ADC channels are mapped to the respective group channels. Group channels are grouped to form Groups by setting INJGRP_CHCR_x. GRP_ID_x[2:0], REGGRP_CHMAP[31:0] and REGGRP_SGRP_CHMAP_x[31:0].
4. Groups – ADC conversion groups. There are 4 injection groups, 1 continuous conversion regular group and 8 discontinuous regular groups. The groups can include multiple group channels.

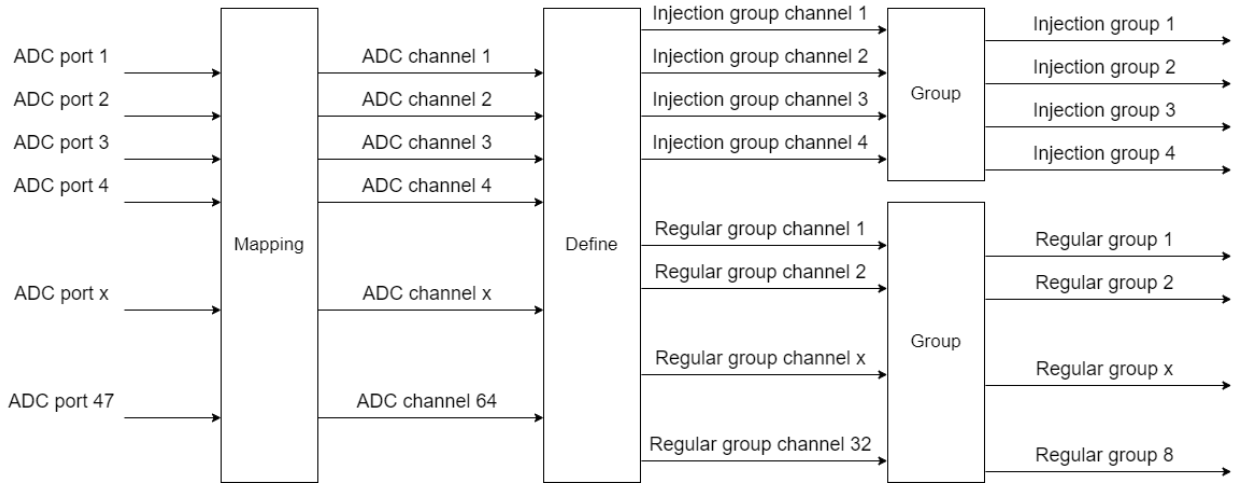


Figure 26 ADC channels and groups mapping

Notes:

1. When ADC channel 1/2/3/4 is configured as differential-ended input, the corresponding PGA1/2/3/4 must be enabled.
2. When ADC channel 1/2/3/4 is configured as differential-ended input, they must be mapped to two ADC pin channels.
3. When ADC channel 1/2/3/4 is configured as differential-ended input, they operate as a synchronized sampling channels.
4. When differential-ended channels and single-ended channels are allocated to injection group, the differential-ended channels and the single-ended should be allocated to different sub-groups.

Table 33 ADC channels mapping.

ADC channel	Description	IO	64pin	48pin	32pin	Port Config IOCFG_IO_x _CFG	PGA P Source CMPCFG3_ PGAx_PSRC	PGA N Source CMPCFG3_ PGAx_NSRC
1	Single-ended	52	56	32	29	Cfg 8		
1	Differential-ended (Positive)	52	56	32	29	Cfg 0	PGA1_PSRC = 01	
1	Differential-ended (Positive)	45	49	25	-	Cfg 0	PGA1_PSRC =10	
1	Differential-ended (Positive)	33	37			Cfg 0	PGA1_PSRC =11	
1	Differential-ended	51	55	31	28	Cfg 0		PGA1_NSRC = 10

ADC channel	Description	IO	64pin	48pin	32pin	Port Config IOCFG_IO_x_CFG	PGA P Source CMPCFG3_PGAx_PSRC	PGA N Source CMPCFG3_PGAx_NSRC
	(Negative)							
1	Differential-ended (Negative)	34	38			Cfg 0		PGA1_NSRC = 11
2	Single-ended	50	54	30	27	Cfg 8		
2	Differential-ended (Positive)	50	54	30	27	Cfg 0	PGA2_PSRC = 01	
2	Differential-ended (Positive)	46	50	27		Cfg 0	PGA2_PSRC = 10	
2	Differential-ended (Positive)	31	35			Cfg 0	PGA2_PSRC = 11	
2	Differential-ended (Negative)	49	53	29	26			PGA2_NSRC = 10
2	Differential-ended (Negative)	32	36			Cfg 0		PGA2_NSRC = 11
3	Single-ended	58	62	35	31	Cfg 8		
3	Differential-ended (Positive)	58	62	35	31	Cfg 0	PGA3_PSRC = 01	
3	Differential-ended (Positive)	47	51	27		Cfg 0	PGA3_PSRC = 10	
3	Differential-ended (Positive)	26	30			Cfg 0	PGA3_PSRC = 11	
3	Differential-ended (Negative)	57	61	34	30	Cfg 0		PGA3_NSRC = 10
3	Differential-ended (Negative)	27	31			Cfg 0		PGA3_NSRC = 11
4	Single-	42	46	22	24	Cfg 8		

ADC channel	Description	IO	64pin	48pin	32pin	Port Config IOCFG_IO_x_CFG	PGA P Source CMPCFG3_PGAx_PSRC	PGA N Source CMPCFG3_PGAx_NSRC
	ended							
4	Differential-ended (Positive)	42	46	22	24	Cfg 0	PGA4_PSRC = 11	
4	Differential-ended (Positive)	61	1			Cfg 0	PGA4_PSRC = 01	
4	Differential-ended (Positive)	48	52	28		Cfg 0	PGA4_PSRC = 10	
4	Differential-ended (Negative)	62	2			Cfg 0		PGA4_NSRC = 10
4	Differential-ended (Negative)	41	45	21	23	Cfg 0		PGA4_NSRC = 11
5	Single-ended	40	44	20	22	Cfg 8		
6	Single-ended	37	41	17	19	Cfg 8		
7	Single-ended	30	34	14	16	Cfg 8		
8	Single-ended	60	64	36		Cfg 8		
9	Single-ended	53	57	33		Cfg 8		
10	Single-ended	21	25	7		Cfg 8		
11	Single-ended	1	3	37	32	Cfg 8		
12	Single-ended	20	24	6		Cfg 8		
13	Single-ended	34	38			Cfg 8		
14	Single-ended	33	37			Cfg 8		
15	Single-ended	32	36			Cfg 8		
16	Single-	31	35			Cfg 8		

ADC channel	Description	IO	64pin	48pin	32pin	Port Config IOCFG_IO_x _CFG	PGA P Source CMPCFG3_ PGAx_PSRC	PGA N Source CMPCFG3_ PGAx_NSRC
	ended							
17	Single-ended	27	31			Cfg 8		
18	Single-ended	26	30			Cfg 8		
19	Single-ended	62	2			Cfg 8		
20	Single-ended	61	1			Cfg 8		
25	Single-ended	51	55	31	28	Cfg 8		
26	Single-ended	49	53	29	26	Cfg 8		
27	Single-ended	57	61	34	30	Cfg 8		
28	Single-ended	41	45	21	23	Cfg 8		
29	Single-ended	22	26	8		Cfg 8		
30	Single-ended	23	27	9		Cfg 8		
31	Single-ended	24	28	10		Cfg 8		
32	Single-ended	16	20	4		Cfg 8		
33	Single-ended	15	19	3	11	Cfg 8		
34	Single-ended	14	18	2	10	Cfg 8		
35	Single-ended	13	17	1	9	Cfg 8		
36	Single-ended	2	4			Cfg 8		
37	Single-ended	18	22			Cfg 8		
38	Single-ended	19	23			Cfg 8		
39	Single-ended	25	29	11	13	Cfg 8		

ADC channel	Description	IO	64pin	48pin	32pin	Port Config IOCFG_IO_x _CFG	PGA P Source CMPCFG3_ PGAx_PSRC	PGA N Source CMPCFG3_ PGAx_NSRC
40	Single-ended	43	47	23		Cfg 8		
41	Single-ended	44	48	24	25	Cfg 8		
42	Single-ended	56	60			Cfg 8		
43	Single-ended	59	63			Cfg 8		
44	Single-ended	17	21	5	12	Cfg 8		

Table 34 ADC channel definition

Bit	Index	ADC channel
000000	1	1 PN
000001	2	2 PN
000010	3	3 PN
000011	4	4 PN
000100	5	5
000101	6	6
000110	7	7
000111	8	8
001000	9	9
001001	10	10
001010	11	11
001011	12	12
001100	13	13
001101	14	14
001110	15	15
001111	16	16
010000	17	17
010001	18	18
010010	19	19
010011	20	20
010100	21	21 (no used)
010101	22	22 (no used)
010110	23	23 (no used)
010111	24	24 (no used)

011000	25	25
011001	26	26
011010	27	27
011011	28	28
011100	29	29
011101	30	30
011110	31	31
011111	32	32
100000	33	33
100001	34	34
100010	35	35
100011	36	36
100100	37	37
100101	38	38
100110	39	39
100111	40	40
101000	41	41
101001	42	42
101010	43	43
101011	44	44
101100	45	45 (no used)
101101	46	46 (no used)
101110	47	47 (no used)
101111	48	48 (no used)
110000	49	49 (no used)
110001	50	50 (no used)
110010	51	51 (no used)
110011	52	52 (no used)
110100	53	53 (no used)
110101	54	Vbg
110110	55	Vts
110111	56	Vbs
111000	57	0.2*AVDD
111001	58	0.3*AVDD
111010	59	0.4*AVDD
111011	60	0.5*AVDD
111100	61	0.6*AVDD
111101	62	0.7*AVDD
111110	63	0.8*AVDD
111111	64	reserved

14.4.1.4 ADC Injection conversion

ADC controller supports up to 4 injection groups which may include up to 4 injection channels.
At least one injection channel must be assigned to an injection group with up to a maximum of 4 injection channels can be assigned to the same injection group. However, once a particular injection channel is assigned to an injection group, it cannot be further assigned to another injection group. Each injection channel may map to any one of the ADC channels. It is possible to map a single ADC channel to multiple injection channels.
A specific event trigger can be assigned to each injection group. It is allowable to have different groups share the same event trigger.
The ADC performs data conversion for the corresponding injection group once the trigger is detected. If multiple injection groups are defined, they will be triggered independently.
Injection group has higher priority than regular group. If an injection group is triggered while a regular group is sampling or performing data conversion, the injection group would take over immediately and the regular group operations would be aborted. Once the injection group operation is completed, the data conversion process in the regular group will be restarted.

14.4.1.5 ADC regular conversion

ADC controller supports regular groups which may include up to 32 regular channels. The regular channels may map to the same ADC channel or different ADC channels.

14.4.1.5.1 Single conversion

In single conversion mode, ADC controller only support one regular group which is regular group 1. The ADC performs a single data conversion after receiving the trigger. When MDU_ADC_CFG.ADCCFG_MCONVMODE[7] is 1 and MDU_REGGRP_TRGCFG. REGGRP_OPMODE[31:30] is 0, the ADC runs in single conversion mode.

14.4.1.5.2 Continuous conversion

When MDU_ADC_CFG.ADCCFG_MCONVMODE[7] is 1 and MDU_REGGRP_TRGCFG. REGGRP_OPMODE[31:30] is 1, the ADC runs in continuous conversion mode.
The ADC controllers supports only one regular group, regular group 1, in continuous conversion mode. This regular group can include up to 32 regular channels which can be the same or different ADC channels. The following requirements must be satisfied in order to run the ADC in continuous conversion mode.

1. Configure regular group 1.
2. Trigger mode should be rising edge or falling edge.
3. Trigger signal is invalid.
4. The conversion is controlled by conversion start (MDU_REGGRP_TRGCFG[28].REGGRP_CON_STRT is 1) .

14.4.1.5.3 Discontinuous conversion

When MDU_ADC_CFG.ADCCFG_MCONVMODE[7] is 1 and MDU_REGGRP_TRGCFG. REGGRP_OPMODE[31:30] is 2, the ADC run in discontinuous conversion mode.
In discontinuous conversion mode, the ADC controller supports up to 8 regular groups which may include up to 32 regular channels. Only a total of 32 regular channels can be supported among all the groups. A regular channel may map to the same ADC channel or different ADC channels. Each regular group has an individual trigger and different groups may share the same trigger. Please note however that the ADC would perform data conversion for one regular group only even if multiple groups use the same trigger upon a trigger event. When multiple regular groups are defined, the groups would be triggered in a queue. The trigger of the remaining groups in the queue would be ignored if the previous group did not complete its conversion. This means that the user should carefully match the sequence of the trigger queue with the regular group queue.

14.4.1.6 ADC controller examples

14.4.1.6.1 Injection conversion with interrupt and DMA (TBD)

The following steps configure the ADC to perform injection conversion:

1. Configure DMA to read the ADC results.
2. Configure ADC core in single conversion mode (MDU_ADC_CFG. ADC_SCONV = 1)
3. Configure ADC control including discharge phase enable and cycles, PGA pre-sampling mode and cycles, reset pulse width, output data alignment and resolution.
4. Enable regular DMA handshaking
5. Enable regular group
6. Set regular group number and group channels mapping
7. Power on ADC core
8. Clear interrupt flags, set interrupt polarity and enable interrupts
9. Configure regular group with channels mapping trigger source and trigger mode
10. Configure regular channels with group identification, PGA pre-sampling cycles, ADC channel and AWD.
11. Enable ADC controller
12. Configure PGA including enable, miller, channel source
13. Enable PGA channel
14. Configure ADC input ports
15. Start conversion
16. Wait for the regular EOS interrupt. Then clear the interrupt flag and perform further processing. DMA should also be reading the ADC conversion results concurrently.

Notes: According to the DMA controller, user should assign group channels to continuous

group channels beginning at group channel 1.

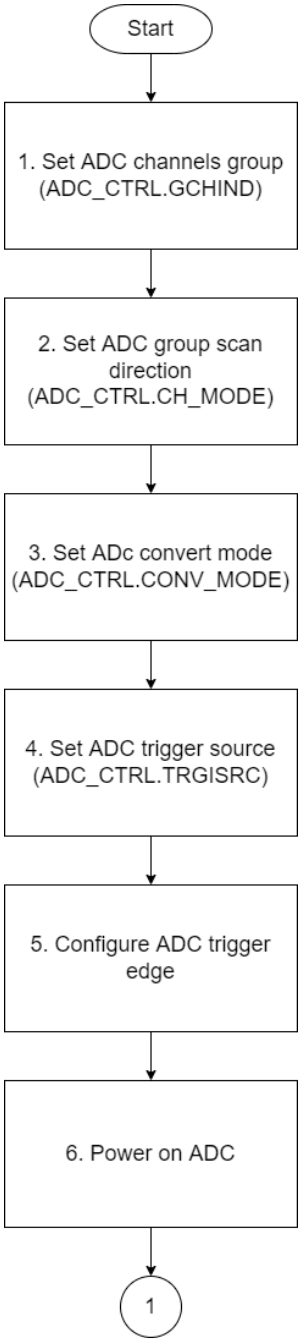


Figure 27 ADC routing-1

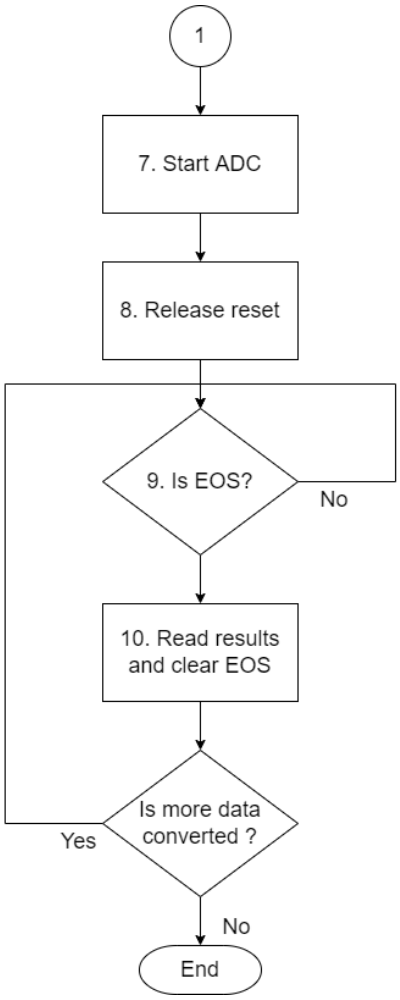


Figure 28 ADC routing-2

For NTO Chips,

14.4.1.6.2 Regular conversion with interrupt and DMA (TBD)

The following steps configure the ADC to perform regular conversion:

1. Configure DMA to read the ADC results.
2. Configure ADC core in single conversion mode (MDU_ADC_CFG. ADC_SCONV = 1)
3. Configure ADC control including discharge phase enable and cycles, PGA pre-sampling mode and cycles, reset pulse width, output data alignment and resolution.
4. Enable regular DMA handshaking
5. Enable regular group
6. Set regular group number
7. Set regular conversion mode
8. Power on ADC core
9. Clear interrupt flags, set interrupt polarity and enable interrupts
10. Configure regular group with trigger source and trigger mode

11. Configure regular channels with group identification, PGA pre-sampling cycles, ADC channel and AWD.
12. Enable ADC controller
13. Configure PGA including enable, miller, channel source
14. Enable PGA channel
15. Configure ADC input ports
16. Start conversion
17. Wait for the regular EOS interrupt. Then clear the interrupt flag and perform further processing. DMA should also be reading the ADC conversion results concurrently.

Notes: According to the DMA controller, user should assign group channels to continuous group channels beginning at group channel 1.

14.4.1.6.3 Manual mode (TBD)

14.4.1.7 ADC code calibration and correction

Due to the discrete ADC gain settings designed for a certain supply voltage, process variation and temperatures change, the ADC output code may not be perfectly proportional to its input voltage. ADC code software calibration and hardware correction are implemented in XT32H series to achieve an accurate result.

The correction of ADC output code is calculated as Equation 14-3.

$$D_{CAL,X} = k \cdot D_X + D_{OS} \qquad \text{(Equation 14-3)}$$

where $D_{CAL,X}$ is the ADC code after correction, D_X is the ADC output raw code, k is the ADC gain mis-match constant and D_{OS} is the code offset.

During the calibration process, the k and D_{OS} values are calculated and stored in the registers. The code correction procedure is implemented in ADC hardware.

The calibration process is shown below:

1. Set ADC input channel to internal reference $\alpha \cdot V_{DD}$ (CH56/57). Capture the corresponding ADC output D_α.
2. Set ADC input channel to internal reference $\beta \cdot V_{DD}$ (CH62/61). Capture the corresponding ADC output D_β.
3. Calculate slope constant as $k = \frac{\alpha-\beta}{D_\alpha - D_\beta} \cdot 4096$
4. Calculate offset constant as $D_{OS} = \alpha \cdot 4096 - k \cdot D_\alpha$

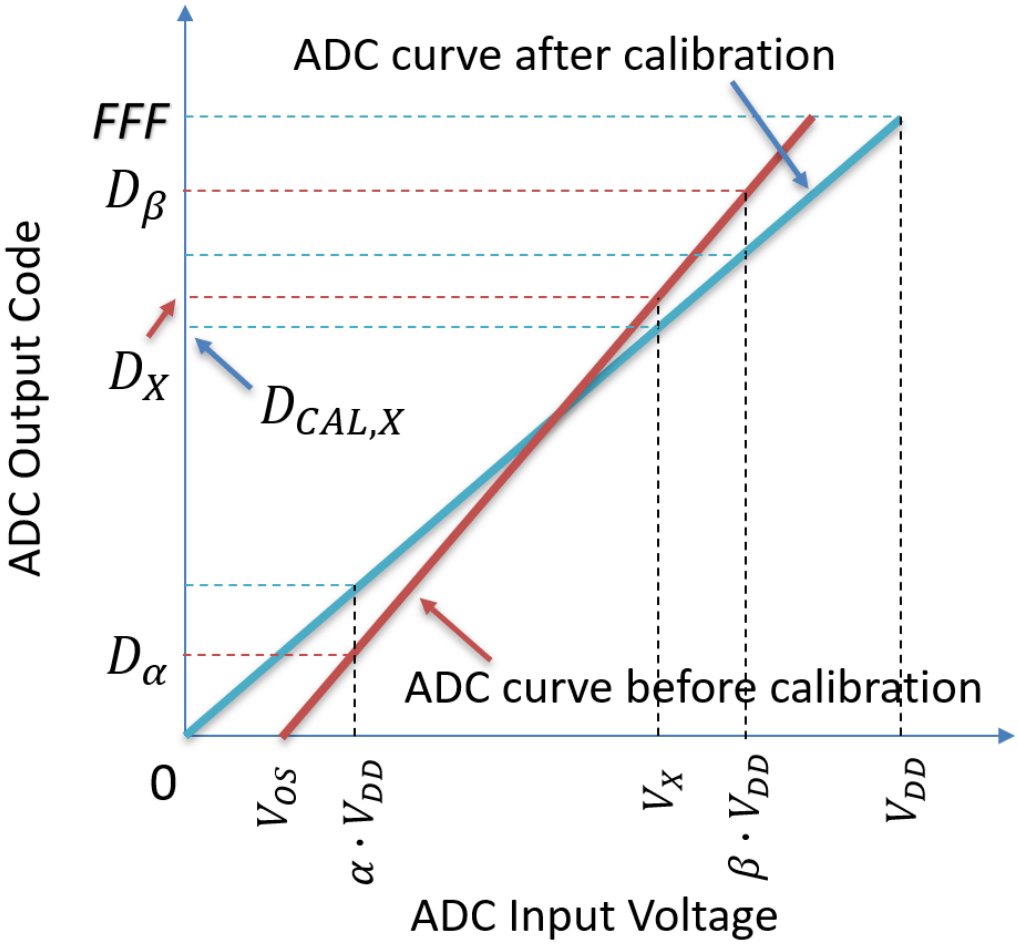


Figure 29 ADC output code calibration and correction

[process description]

The calibration procedure is illustrated as the flow chart below.

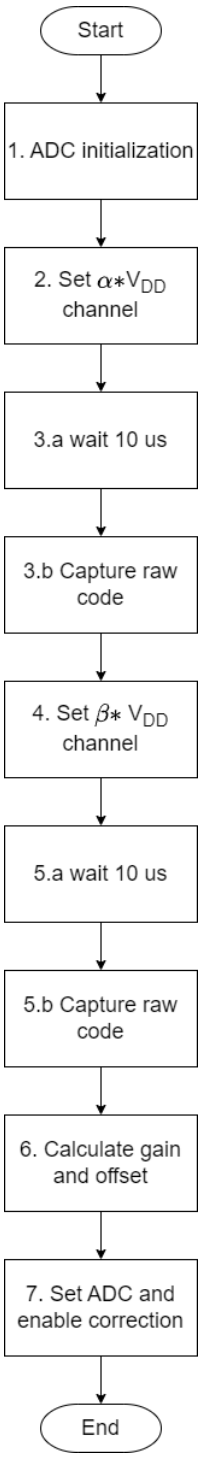


Figure 30 ADC calibration

Note: If ADC used calibration channel, calibration should be processed before ADC was initialized, since calibration will occupy ADC channel and group.

14.4.2 Comparator

XT32H serial has 4 comparators. Signal input for comparison can be from an external input pin or from the PGA output. The other input of the comparator can be from an external input pin, or from the internal DAC outputs. Hysteresis of the comparators can be set to 0, 10mV, 20mV and 30mV. Polarity of the comparator outputs can be inverted by setting the control bit.
[block]
[example]

14.4.3 PGA

XT32H serial has 4 sets of PGAs, which accept differential inputs from the chip external input pin. Input difference is amplified and outputted with a DC offset, as shown in Equation 14-3 and Figure **.

$$V_{PGA_OUT} = Gain_{PGA} \cdot \left(V_{PGA_P} - V_{PGA_N}\right) + V_{PGA_OS} \qquad \text{(Equation 14-3)}$$

where V_{PGA_OUT} is the PGA output, V_{PGAP}, V_{PGAN} are the input voltages to PGA inverting and non-inverting input, $Gain_{PGA}$ is the PGA gain and V_{PGA_OS} is the output offset.

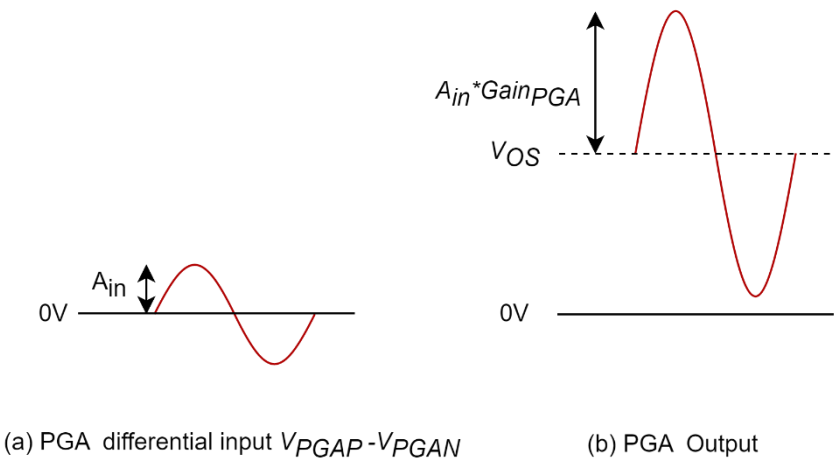


Figure 31 Illustration of PGA input and output

The gain of each PGA can be set from 2 to 15, and V_{PGA_OS} is usually set to half of the supply voltage. The PGA output voltage can be sampled by the ADC for data conversion, and/or sensed by the corresponding comparator for voltage comparison.

14.4.4 DAC

XT32H serial has 2 8-bit DACs. Reference voltage of DAC can be set to 1.2V, 2.4V, 3.6V and power supply. The DAC output can be routed to the comparators either directly or through an analog buffer. The DAC output can be used as a threshold for voltage comparisons.
DAC output voltage is calculated as:

$$V_{DAC} = \frac{D_{DAC_IN}}{2^N} \cdot V_{DAC_REF} \qquad \text{(Equation 14-4)}$$

14.5 MDU registers

14.5.1 MDU_PDCTRL

Base address : 0x4000_2800
Address offset: 0x00
Type: Read/Write
Reset value: 0x0011_1111

Bit	Description	Type
<31:29>	Reserved	R/W
<28>	**PDCTRL_TEMPSPD**: Temperature sensor power down control 1'b0: Power on 1'b1: Power down	R/W
<27:24 >	**PDCTRL_PGAENx**: PGA x (x=1 to 4) enable 1'b0: PGA x disable 1'b1: PGA x enable	R/W
<23:22>	Reserved	R/W
<21>	**PDCTRL_ADCBUF_EN**: ADC reference buffer enable 1'b0: Disable 1'b1: Enable	R/W
<20>	**PDCTRL_ADCPD**: Power down control of ADC 1'b0: Power on 1'b1: Power down	R/W
<19:17>	Reserved	R/W
<16>	**PDCTRL_CMP4PD**: Power down control of comparator 4 1'b0: Power on 1'b1: Power down	R/W
<15:13>	Reserved	R/W
<12>	**PDCTRL_CMP3PD**: Power down control of comparator 3 1'b0: Power on 1'b1: Power down	R/W
<11:9>	Reserved	R/W

Bit	Description	Type
<8>	**PDCTRL_CMP2PD**: Power down control of comparator 2 1'b0: Power on 1'b1: Power down	R/W
<7:5>	Reserved	R/W
<4>	**PDCTRL_CMP1PD**: Power down control of comparator 1 1'b0: Power on 1'b1: Power down	R/W
<3:1>	Reserved	R/W
<0>	**PDCTRL_DACPD**: Power down control of DAC 1'b0: Power on 1'b1: Power down	R/W

14.5.2 MDU_DAC_CFG

Base address : 0x4000_2800
Address offset: 0x04
Type: Read/Write
Reset value: 0x0110_0000

Bit	Description	Type
<31:30>	**DACCFG_DAC2_TSTM**: Signal selection from dacout2 2'b00 – dacout2 2'b 01 – adc_com (vrefp for adc) 2'b 10 – ana_ldo 2'b 11 – ull_ldo	R/W
<29:28>	**DACCFG_DAC1_TSTM**: Signal selection from dacout1 2'b 00 – dacout1 2'b 01 – vref_tst 2'b 10 – core_ldo 2'b 11 – flash_ldo	R/W
<27:25>	Reserved	R/W
<24>	**DACCFG_DAC2BBP**: DAC 2 buffer bypass control 1'b0: use buffer 1'b1: bypass buffer	R/W
<23:21>	Reserved	R/W
<20>	**DACCFG_DAC1BBP**: DAC 1 buffer bypass control 1'b0: use buffer 1'b1: bypass buffer	R/W
<19:18>	Reserved	R/W
<17:16>	**DACCFG_REFSEL[1:0]**: DAC Reference selection 2'b 00 – VREF=1.2V 2'b 01 – VREF=2.4V	R/W

Bit	Description	Type
	2’b 10 – VREF=3.6V 2’b 11 – VREF=VDD	
<15:8>	**DACCFG_DAC2ADD[7:0]**: DAC 2 add value	R/W
<7:0>	**DACCFG_DAC1ADD[7:0]**: DAC 1 add value	R/W

14.5.3 MDU_PGA_CFG

Base address : 0x4000_2800
Address offset: 0x08
Type: Read/Write
Reset value: 0x001F_4444

Bit	Description	Type
<31:21>	Reserved	R/W
<20>	**PGACFG_ADC_VCM_HVDD**: PGA offset voltage setting 1'b0: PGA offset voltage is set by ADC_VCM_TRIM 1'b1: PGA offset voltage is set to half of VDD.	R/W
<19:16>	**PGACFG_ADC_VCM_TRIM[3:0]**: PGA offset voltage setting when ADC_VCM_HVDD= 1'b0(it is suggested to set VCM to half of VDD) 4'b0000: VCM=1.0V 4'b0001: VCM=1.1V 4'b0010: VCM=1.2V … 4'b1111: VCM=2.5V	R/W
<15>	Reserved	R/W
<14:12>	**PGACFG_PGA4GAIN[2:0]**: PGA Gain 4 Settings are the same as PGA_GAIN1.	R/W
<11>	Reserved	R/W
<10:8>	**PGACFG_PGA3GAIN[2:0]**: PGA Gain 3 Settings are the same as PGA_GAIN1.	R/W
<7>	Reserved	R/W
<6:4>	**PGACFG_PGA2GAIN[2:0]**: PGA Gain 2 Settings are the same as PGA_GAIN1.	R/W
<3>	Reserved	R/W
<2:0>	**PGACFG_PGA1GAIN[2:0]**: PGA Gain 1 3'b000: gain=2 3'b001: gain=3 3'b010: gain=4 3'b011: gain=5 3'b100: gain=7 3'b101: gain=9 3'b110: gain=11 3'b111: gain=15	R/W

14.5.4 MDU_ADC_CFG

Base address : 0x4000_2800
Address offset: 0x0C
Type: Read/Write
Reset value: 0x3AA6_00F5

Bit	Description	Type
<31>	**ADCCFG_ADCOUTALIGN**: ADC output data alignment 1'b0: Least Significant Byte (LSB) 1'b1: Most Significant Byte (MSB)	R/W
<30:29>	**ADCCFG_ADCTSAMPLE[1:0]**: ADC sampling time control 2'b00: Sampling time equals to 1 clock cycles 2'b01: Sampling time equals to 3 clock cycles 2'b10: Sampling time equals to 7 clock cycles 2'b11: Sampling time equals to 15 clock cycles	R/W
<28>	**ADCCFG_AMPMILLER**: Miller capacitor control for pre-amp. '1' 1'b0: 0.9pF 1'b1: 1.7pF	R/W
<27:26>	**ADCCFG_PGAMILLER4[1:0]**: Miller capacitor control for PGA4. 2'b00:2.4pF (for gain=2) 2'b01:1.8pF 2'b10:1.2pF 2'b11:0.6pF (for gain=15)	R/W
<25:24>	**ADCCFG_PGAMILLER3[1:0]**: Miller capacitor control for PGA3. Settings are the same as PGA_MILLER3.	R/W
<23:22>	**ADCCFG_PGAMILLER2[1:0]**: Miller capacitor control for PGA2. Settings are the same as PGA_MILLER3.	R/W
<21:20>	**ADCCFG_PGAMILLER1[1:0]**: Miller capacitor control for PGA1. Settings are the same as PGA_MILLER3.	R/W
<19:18>	**ADCCFG_AMPGAIN[1:0]**: Gain for ADC pre-amplifier. Value is set according to VDD. 2'b00: VDD=5.565V 2'b01: VDD=5.0V 2'b10: VDD=3.3V 2'b11: VDD=2.5V	R/W
<17:16>	**ADCCFG_ADCDATAW[1:0]**: Configure ADC resolution 2'b00: 8-bit ADC output 2'b01: 10-bit ADC output 2'b10: 12-bit ADC output (default) 2'b11: N/A	R/W
<15>	**ADCCFG_MSAMPEN**: PGA sample clock pulse enable for manual control	R/W

	mode. 1’b0: Disable 1’b1: Enable Note: When ADC operates in auto mode, this bit has no effect.	
<14:12>	Reserved	R/W
<11:8>	**ADCCFG_RSTBDLY[3:0]**: rstc delay count, default is 0. 4’b0000: …	R/W
<7>	**ADCCFG_MCONVMODE**: ADC conversion mode 1’b0: ADC continuous conversion 1’b1: ADC single conversion	R/W
<6>	**ADCCFG_ADCC_EN**: ADC auto control enable 1’b0: Manual control 1’b1: Auto control	R/W
<5:0>	**ADCCFG_MCHSLCT[5:0]**: ADC channel selection. Refer to table 33. 6’b000000: channel 0 … Note: when ADC operates in auto mode, the bits has no effect.	R/W

14.5.5 MDU_COMP_CFG1

Base address : 0x4000_2800
Address offset: 0x10
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:29>	Reserved	R/W
<28>	**CMPCFG1_PSRC_4**: Comparator 4, non-inverting input source selection 1'b0: Comparator 4 non-inverting input from external IO 1'b1: Comparator 4 non-inverting input from internal pgaout4	R/W
<27:26>	Reserved	R/W
<25:24>	**CMPCFG1_NSRC_4[1:0]**: Comparator 4, inverting input source selection 2'b00: Comparator 4 inverting input from external IO 2'b01: Reserved 2'b10: Comparator 4 inverting input from internal dacout1 2'b11: Comparator 4 inverting input from external dacout2	R/W
<23:21>	Reserved	R/W
<20>	**CMPCFG1_PSRC_3**: Comparator 3, non-inverting input source selection 1'b0: Comparator 3 non-inverting input from external IO 1'b1: Comparator 3 non-inverting input from internal pgaout3	R/W
<19:18>	Reserved	R/W
<17:16>	**CMPCFG1_NSRC_3[1:0]**: Comparator 3, inverting input source selection 2'b00: Comparator 3 inverting input from external IO 2'b01: Reserved 2'b10: Comparator 3 inverting input from internal dacout1 2'b11: Comparator 3 inverting input from external dacout2	R/W
<15:13>	Reserved	R/W
<12>	**CMPCFG1_PSRC_2**: Comparator 2, non-inverting input source selection 1'b0: Comparator 2 non-inverting input from external IO 1'b1: Comparator 2 non-inverting input from internal pgaout2	R/W
<11:10>	Reserved	R/W
<9:8>	**CMPCFG1_NSRC_2[1:0]**: Comparator 2, inverting input source selection 2'b00: Comparator 2 inverting input from external IO 2'b01: Reserved 2'b10: Comparator 2 inverting input from internal dacout1 2'b11: Comparator 2 inverting input from external dacout2	R/W
<7:5>	Reserved	R/W
<4>	**CMPCFG1_PSRC_1**: Comparator 1, non-inverting input source selection 1'b0: Comparator 1 non-inverting input from external IO 1'b1: Comparator 1 non-inverting input from internal pgaout1	R/W

<3:2>	Reserved	R/W
<1:0>	**CMPCFG1_NSRC_1[1:0]**: Comparator 1, inverting input source selection 2'b00: Comparator 1 inverting input from external IO 2'b01: Reserved 2'b10: Comparator 1 inverting input from internal dacout1 2'b11: Comparator 1 inverting input from external dacout2	R/W

14.5.6 MDU_COMP_CFG2

Base address : 0x4000_2800
Address offset: 0x14
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	Reserved	R/W
<30:28>	**CMPCFG2_SPD_4[2:0]**: comp4_spd<2:0> 3'b000: Low speed … 3'b111: High speed	R/W
<27:26>	Reserved	R/W
<25:24>	**CMPCFG2_HYS_4[1:0]**: comp4_hys<1:0> 2'b00: No hysteresis 2'b01: 10mV hysteresis 2'b10: 20mV hysteresis 2'b11: 30mV hysteresis	R/W
<23>	Reserved	R/W
<22:20>	**CMPCFG2_SPD_3[2:0]**: comp3_spd<2:0> 3'b000: Low speed … 3'b111: High speed	R/W
<19:18>	Reserved	R/W
<17:16>	**CMPCFG2_HYS_3[1:0]**: comp3_hys<1:0> 2'b00: No hysteresis 2'b01: 10mV hysteresis 2'b10: 20mV hysteresis 2'b11: 30mV hysteresis	R/W
<15>	Reserved	R/W
<14:12>	**CMPCFG2_SPD_2[2:0]**: comp2_spd<2:0> 3'b000: Low speed … 3'b111: High speed	R/W
<11:10>	Reserved	R/W

Bit	Description	Type
<9:8>	**CMPCFG2_HYS_2[1:0]**: comp2_hys<1:0> 2'b00: No hysteresis 2'b01: 10mV hysteresis 2'b10: 20mV hysteresis 2'b11: 30mV hysteresis	R/W
<7>	Reserved	R/W
<6:4>	**CMPCFG2_SPD_1[2:0]**: comp1_spd<2:0> 3'b000: Low speed … 3'b111: High speed	R/W
<3:2>	Reserved	R/W
<1:0>	**CMPCFG2_HYS_1[1:0]**: comp1_hys<1:0> 2'b00: No hysteresis 2'b01: 10mV hysteresis 2'b10: 20mV hysteresis 2'b11: 30mV hysteresis	R/W

14.5.7 MDU_COMP_CFG3

Base address : 0x4000_2800
Address offset: 0x18
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CMPCFG3_PGA4_NSRC[1:0]**: PGA4 inverting terminal source selection 2'b0x: No input 2'b10: Channel 18 2'b11: Channel 27	R/W
<29:28>	**CMPCFG3_PGA4_PSRC [1:0]**: PGA4 non-inverting terminal source selection 2'b00: No input 2'b01: Channel 19 2'b10: Channel 23 2'b11: Channel 3	R/W
<27:26>	**CMPCFG3_PGA3_NSRC[1:0]**: PGA3 inverting terminal source selection 2'b0x: No input 2'b10: Channel 26 2'b11: Channel 16	R/W
<25:24>	**CMPCFG3_PGA3_PSRC[1:0]**: PGA3 non-inverting terminal source selection 2'b00: No input 2'b01: Channel 2 2'b10: Channel 22	R/W

Bit	Description	Type
	2’b11: Channel 17	
<23:22>	**CMPCFG3_PGA2_NSRC[1:0]**: PGA2 inverting terminal source selection 2’b0x: No input 2’b10: Channel 25 2’b11: Channel 14	R/W
<21:20>	**CMPCFG3_PGA2_PSRC[1:0]**: PGA2 non-inverting terminal source selection 2’b00: No input 2’b01: Channel 1 2’b10: Channel 21 2’b11: Channel 15	R/W
<19:18>	**CMPCFG3_PGA1_NSRC[1:0]**: PGA1 inverting terminal source selection 2’b0x: no input; 2’b10: channel 24; 2’b11: channel 12	R/W
<17:16>	**CMPCFG3_PGA1_PSRC[1:0]**: PGA1 non-inverting terminal source selection 2’b00: No input 2’b01: Channel 0 2’b10: Channel 20 2’b11: Channel 13	R/W
<15:14>	Reserved	R/W
<13>	**CMPCFG3_BPFLG**: bypass comparator interrupt flags 1’b0: Do not bypass flag 1’b1: Bypass flag	R/W
<12>	**CMPCFG3_CMP4_FTRG**: Comparator 4 interrupt trigger 1’b0: No trigger 1’b1: Falling edge trigger interrupt	R/W
<11>	**CMPCFG3_CMP4_RTRG**: Comparator 4 interrupt trigger 1’b0: No trigger 1’b1: Rising edge trigger interrupt	R/W
<10>	**CMPCFG3_CMP3_FTRG**: Comparator 3 interrupt trigger 1’b0: No trigger 1’b1: Falling edge trigger interrupt	R/W
<9>	**CMPCFG3_CMP3_RTRG**: Comparator 3 interrupt trigger 1’b0: No trigger 1’b1: Rising edge trigger interrupt	R/W
<8>	**CMPCFG3_CMP2_FTRG**: Comparator 2 interrupt trigger 1’b0: No trigger 1’b1: Falling edge trigger interrupt	R/W
<7>	**CMPCFG3_CMP2_RTRG**: Comparator 2 interrupt trigger 1’b0: No trigger 1’b1: Rising edge trigger interrupt	R/W

Bit	Description	Type
<6>	**CMPCFG3_CMP1_FTRG**: Comparator 1 interrupt trigger 1'b0: No trigger 1'b1: Falling edge trigger interrupt	R/W
<5>	**CMPCFG3_CMP1_RTRG**: Comparator 1 interrupt trigger 1'b0: No trigger 1'b1: Rising edge trigger interrupt	R/W
<4>	**CMPCFG3_OFLT**: Comparator output filtering or bypass. 1'b0: One clock cycle width glitch will be removed, the clock frequency can be configured by GPIO_CLK_DIV 1'b1: Bypass mode. No glitch will be removed.	R/W
<3:0>	**CMPCFG3_OPOL_x**: Comparator index x (x=4 to 1) output polarity 1'b0: Normal 1'b1: Inverted	R/W

14.5.8 MDU_ADC_CALIB

Base address : 0x4000_2800
Address offset: 0x28
Type: Read/Write
Reset value: 0x0000_4000

Bit	Description	Type
<31>	**ADC_CALIB_CODE_COR_EN**: ADC correction enable. Default is enabled. 1'b0: Disable 1'b1: Enable	R/W
<30>	Reserved	R/W
<29:16>	**ADC_CALIB_CODE_OFFSET[13:0]**: ADC correction code gain. Default value is 0x2000.	R/W
<14:0>	**ADC_CALIB_CODE_GAIN[14:0]**: ADC correction code offset. Default value is 0.	R/W

14.5.9 MDU_DMA_CFG

Base address : 0x4000_2800
Address offset: 0x2C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:2>	Reserved	R/W
<1>	**REG_DMA_EN**: Regular group conversion DMA transfer enable 1'b0: Disable 1'b1: Enable	R/W
<0>	**INJ_DMA_EN**: Injection group conversion DMA transfer enable 1'b0: Disable 1'b1: Enable	R/W

14.5.10 MDU_MDU_CFG

Base address: 0x4000_2800
Address offset: 0x30
Type: Read/Write
Reset value: 0xF104_6600

Bit	Description	Type
<31>	**REG_EOS_EN**: Regular group EOS interrupt enable 1'b0: Disable 1'b1: Enable	R/W
<30>	**REG_EOC_EN**: Regular channel EOC interrupt enable 1'b0: Disable 1'b1: Enable	R/W
<29>	**INJ_EOS_EN**: Injection group EOS interrupt enable 1'b0: Disable 1'b1: Enable	R/W
<28>	**INJ_EOC_EN**: Injection channel EOC interrupt enable 1'b0: Disable 1'b1: Enable	R/W
<27>	**REG_INT_POL**: Regular interrupt polarity 1'b0: High level interrupt 1'b1: Low level interrupt	R/W
<26>	**INJ_INT_POL**: Injection interrupt polarity 1'b0: High level interrupt 1'b1: Low level interrupt	R/W
<25:22>	**RSTB_PLS[3:0]**: Define the ADC reset pulse width (n+1). The default value is 0.	R/W

	4’b0000: 1 ADC clock cycle 4’b0001: 2 ADC clock cycles 4’b0010: 3 ADC clock cycles 4’b0011: 4 ADC clock cycles … 4’bxxxx: n+1 ADC clock cycles …	
<21:16>	**SMPL_CYC[5:0]**: Define the cycle numbers of the PGA pre-sampling when it is used in regular group conversion. （N+1） Default: 6’h4 6’b000000: 1 ADC clock cycle 6’b000001: 2 ADC clock cycles … 6’bxxxxxx: n+1 ADC clock cycles …	R/W
<15>	**ADCC_EN**: adc_enable digital part. ADC controller enable 1’b0: Disable 2’b1: Enable	R/W
<14>	**SW_TRG**: adc_soft_trigger 1’b0: ADC soft trigger is set to 0 1’b1: ADC soft trigger is set to 1	R/W
<13>	**RST**:adc_reset (for ADC digital state machine etc.) 1’b0: Reset 1’b1: Release reset	R/W
<12>	**STRT**: adc_start 1’b0: Stop ADC 1’b1: Start ADC	R/W
<11>	**SMP_MODE**: Select the sync pre-sampling cycle time for channel 0~3 in injection conversion Default: 1’b0 1’b0: Use the same pre-sample cycle numbers from mdu_cfg 1’b1: Use dedicated pre-sample cycle numbers in inj_group_define_x	R/W
<10>	**REGGRP_EN**: enable/disable the regular group conversion Default: 1’b0 1’b0: Disable regular group conversion 1’b1: Enable regular group conversion	R/W
<9>	**INJGRP_EN**: enable/disable the injection group conversion Default: 1’b0 1’b0: Disable injection group conversion 1’b1: Enable injection group conversion	R/W
<8>	**DSCG_EN**: the discharge phase enable, default: 1’b0	R/W

	1’b0: No discharge phase 1’b1: Insert discharge phase before each conversion	
<7:0>	**DSCH_PHS[7:0]**: indicates the cycles of the inserted discharge phase before each new conversion. Default: 0x00	R/W

Note:

1. For differential-ended ADC channels, PGA pre-sampling cycles should be 0x3F.
2. Reset pulse width should be more than 2 us. For example, when ADC work in 32MHz, the reset pulse width should be 64 ADC cycles, the value in **RSTB_PLS[3:0]** is 63.

14.5.11 MDU_INJGRP_TRG

Base address: 0x4000_2800
Address offset: 0x34
Type: Read/Write
Reset value: 0x0100_0021

Bit	Description	Type
<31:30>	Reserved	R/W
<29:28>	~~**INJGRP_OPMODE[1:0]**: injection group operation mode~~ ~~2'b00: single scan mode~~ ~~2'b01: continuous scan mode~~ ~~2'b1x: discontinuous scan mode~~	R/W
<27:24>	**INJGRP_SGRP_NUM[3:0]**: define how many sub-group will be converted by discontinuous mode. 4'b000: 4'b001: one sub-group 4'b010: two sub-groups 4'b011: three sub-groups 4'b100: four sub-groups	R/W
<23:22>	**INJGRP_TRGCFG_4[1:0]**: rising/falling edge trigger for injection sub-group 4 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<21:20>	**INJGRP_TRGCFG_3[1:0]**: rising/falling edge trigger for injection sub-group 3 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<19:18>	**INJGRP_TRGCFG_2[1:0]**: rising/falling edge trigger for injection sub-group 2 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<17:16>	**INJGRP_TRGCFG_1[1:0]**: rising/falling edge trigger for injection sub-group 1 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<15:12>	**INJGRP_TRGSRC_4[3:0]**: trigger source selection for injection sub-group 4 4'h0: No trigger (disabled the sub-group 4)	R/W

	4’h1: ADC trigger source 1 … 4’hF: ADC trigger source 15	
<11:8>	**INJGRP_TRGSRC_3[3:0]**: trigger source selection for injection sub-group 3 4’h0: No trigger (disabled the sub-group 3) 4’h1: ADC trigger source 1 … 4’hF: ADC trigger source 15	R/W
<7:4>	**INJGRP_TRGSRC_2[3:0]**: trigger source selection for injection sub-group 2 4’h0: No trigger (disabled the sub-group 2) 4’h1: ADC trigger source 1 … 4’hF: ADC trigger source 15	R/W
<3:0>	**INJGRP_TRGSRC_1[3:0]**: trigger source selection for injection sub-group 1 4’h0: No trigger (disabled the sub-group 1) 4’h1: ADC trigger source 1 … 4’hF: ADC trigger source 15	R/W

Table 35 Trigger sources

Index	Trigger Source
0	No trigger
1	Advanced Timer 1 Channel 4 output
2	Advanced Timer 1 TRGO2 output
3	Advanced Timer 2 Channel 4 output
4	Advanced Timer 2 TRGO2 output
5	General Timer 1 Channel 4 output
6	General Timer 1 TRGO2 output
7	General Timer 2 Channel 4 output
8	General Timer 2 TRGO2 output
9	General Timer 3 Channel 4 output
10	General Timer 3 TRGO2 output
11	General Timer 4 Channel 4 output
12	General Timer 4 TRGO2 output
13	External signal from Port C.24
14	Basic Timer 11 output
15	Software trigger

14.5.12 MDU_REGGRP_ TRG

Base address: 0x4000_2800
Address offset: 0x38
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:28>	**REGGRP_TRGSRC_8[3:0]**: trigger source selection for regular sub-group 8 4'h0: No trigger (disabled the sub-group 8) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<27:24>	**REGGRP_TRGSRC_7[3:0]**: trigger source selection for regular sub-group 7 4'h0: No trigger (disabled the sub-group 7) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<23:20>	**REGGRP_TRGSRC_6[3:0]**: trigger source selection for regular sub-group 6 4'h0: No trigger (disabled the sub-group 6) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<19:16>	**REGGRP_TRGSRC_5[3:0]**: trigger source selection for regular sub-group 5 4'h0: No trigger (disabled the sub-group 5) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<15:12>	**REGGRP_TRGSRC_4[3:0]**: trigger source selection for regular sub-group 4 4'h0: No trigger (disabled the sub-group 4) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<11:8>	**REGGRP_TRGSRC_3[3:0]**: trigger source selection for regular sub-group 3 4'h0: No trigger (disabled the sub-group 3) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<7:4>	**REGGRP_TRGSRC_2[3:0]**: trigger source selection for regular sub-group 2 4'h0: No trigger (disabled the sub-group 2) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W
<3:0>	**REGGRP_TRGSRC_1[3:0]**: trigger source selection for regular sub-group 1 4'h0: No trigger (disabled the sub-group 1) 4'h1: ADC trigger source 1 … 4'hF: ADC trigger source 15	R/W

14.5.13 MDU_REGGRP_TRGCFG

Base address: 0x4000_2800
Address offset: 0x3C
Type: Read/Write
Reset value: 0x0000_00A2

Bit	Description	Type
<31:30>	**REGGRP_OPMODE[1:0]**: define regular group operation mode 2'b00: Single scan mode 2'b01: Continuous scan mode 2'b1x: Discontinuous scan mode	R/W
<29>	Reserved	R/W
<28>	**REGGRP_CON_STRT**: start group continuous scan conversion. 1'b0: Stop the continuous group conversion 1'b1: Start the continuous group conversion	R/W
<27:24>	Reserved	R/W
<23:22>	**REGGRP_TRGCFG_8[1:0]**: rising/falling edge trigger for regular sub-group 8 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<21:20>	**REGGRP_TRGCFG_7[1:0]**: rising/falling edge trigger for regular sub-group 7 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<19:18>	**REGGRP_TRGCFG_6[1:0]**: rising/falling edge trigger for regular sub-group 6 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<17:16>	**REGGRP_TRGCFG_5[1:0]**: rising/falling edge trigger for regular sub-group 5 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<15:14>	**REGGRP_TRGCFG_4[1:0]**: rising/falling edge trigger for regular sub-group 4 2'b00: No trigger 2'b01: Rising edge trigger 2'b10: Falling edge trigger 2'b11: Rising and falling edge trigger	R/W
<13:12>	**REGGRP_TRGCFG_3[1:0]**: rising/falling edge trigger for regular sub-group 3	R/W

	2’b00: No trigger 2’b01: Rising edge trigger 2’b10: Falling edge trigger 2’b11: Rising and falling edge trigger	
<11:10>	**REGGRP_TRGCFG_2[1:0]**: rising/falling edge trigger for regular sub-group 2 2’b00: No trigger 2’b01: Rising edge trigger 2’b10: Falling edge trigger 2’b11: Rising and falling edge trigger	R/W
<9:8>	**REGGRP_TRGCFG_1[1:0]**: rising/falling edge trigger for regular sub-group 1 2’b00: No trigger 2’b01: Rising edge trigger 2’b10: Falling edge trigger 2’b11: Rising and falling edge trigger	R/W
<7:4>	Reserved	R/W
<3:0>	**REGGRP_GRPNUM[3:0]**: define how many sub-group will be converted by discontinuous mode in sequence. 4’b0000: 4’b0001: one sub-group 4’b0010: two sub-groups 4’b0011: three sub-groups 4’b0100: four sub-groups 4’b0101: five sub-group 4’b0110: six sub-groups 4’b0111: seven sub-groups 4’b1000: eight sub-groups	R/W

14.5.14 MDU_INJGRP_CHCR_1

Base address : 0x4000_2800
Address offset: 0x40
Type: Read/Write
Reset value: 0x0003_00A2

Bit	Description	Type
<31:26>	Reserved	R/W
<25:24>	**INJGRP_CH_AWD_ID[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<23:16>	**SMPL_CYC[7:0]**: Programmed sample cycle numbers for this injection channel. pre-sample cycle number is N+1	R/W
<15:10>	Reserved	R/W
<9:4>	**ADC_CHID[5:0]**: ADC channel ID which is assigned to this injection sub-group 6'b000000: Channel 1, synchronized sampling 6'b000001: Channel 2, synchronized sampling 6'b000010: Channel 3, synchronized sampling 6'b000011: Channel 4, synchronized sampling others: Other single-ended channels	R/W
<3>	Reserved	R/W
<2:0>	**INJGRP_CH_SGRP_ID[2:0]**: injection sub-group ID 3'b000: Not in scan conversion sequence 3'b001: Define this channel in sub-group 1 3'b010: Define this channel in sub-group 2 3'b011: Define this channel in sub-group 3 3'b100: Define this channel in sub-group 4	R/W

14.5.15 MDU_INJGRP_CHCR_2

Base address: 0x4000_2800
Address offset: 0x44
Type: Read/Write
Reset value: 0x0001_00B2

Bit	Description	Type
<31:26>	Reserved	R/W
<25:24>	**INJGRP_CH_AWD_ID[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<23:16>	**SMPL_CYC[7:0]**: Programmed sample cycle numbers for this injection channel. pre-sample cycle number is N+1	R/W
<15:10>	Reserved	R/W
<9:4>	**ADC_CHID[5:0]**: ADC channel ID which is assigned to this injection sub-group 6'b000000: Channel 1, synchronized sampling 6'b000001: Channel 2, synchronized sampling 6'b000010: Channel 3, synchronized sampling 6'b000011: Channel 4, synchronized sampling others: Other single-ended channels	R/W
<3>	Reserved	R/W
<2:0>	**INJGRP_CH_SGRP_ID[2:0]**: injection sub-group ID 3'b000: Not in scan conversion sequence 3'b001: Define this channel in sub-group 1 3'b010: Define this channel in sub-group 2 3'b011: Define this channel in sub-group 3 3'b100: Define this channel in sub-group 4	R/W

14.5.16 MDU_INJGRP_CHCR_3

Base address: 0x4000_2800
Address offset: 0x48
Type: Read/Write
Reset value: 0x0002_00C2

Bit	Description	Type
<31:26>	Reserved	R/W
<25:24>	**INJGRP_CH_AWD_ID[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<23:16>	**SMPL_CYC[7:0]**: Programmed sample cycle numbers for this injection channel. pre-sample cycle number is N+1	R/W
<15:10>	Reserved	R/W
<9:4>	**ADC_CHID[5:0]**: ADC channel ID which is assigned to this injection sub-group 6'b000000: Channel 1, synchronized sampling 6'b000001: Channel 2, synchronized sampling 6'b000010: Channel 3, synchronized sampling 6'b000011: Channel 4, synchronized sampling others: Other single-ended channels	R/W
<3>	Reserved	R/W
<2:0>	**INJGRP_CH_SGRP_ID[2:0]**: injection sub-group ID 3'b000: Not in scan conversion sequence 3'b001: Define this channel in sub-group 1 3'b010: Define this channel in sub-group 2 3'b011: Define this channel in sub-group 3 3'b100: Define this channel in sub-group 4	R/W

14.5.17 MDU_INJGRP_CHCR_4

Base address: 0x4000_2800
Address offset: 0x4C
Type: Read/Write
Reset value: 0x0004_00D2

Bit	Description	Type
<31:26>	Reserved	R/W
<25:24>	**INJGRP_CH_AWD_ID[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<23:16>	**SMPL_CYC[7:0]**: Programmed sample cycle numbers for this injection channel. pre-sample cycle number is N+1	R/W
<15:10>	Reserved	R/W
<9:4>	**ADC_CHID[5:0]**: ADC channel ID which is assigned to this injection sub-group 6'b000000: Channel 1, synchronized sampling 6'b000001: Channel 2, synchronized sampling 6'b000010: Channel 3, synchronized sampling 6'b000011: Channel 4, synchronized sampling others: Other single-ended channels	R/W
<3>	Reserved	R/W
<2:0>	**INJGRP_CH_SGRP_ID[2:0]**: injection sub-group ID 3'b000: Not in scan conversion sequence 3'b001: Define this channel in sub-group 1 3'b010: Define this channel in sub-group 2 3'b011: Define this channel in sub-group 3 3'b100: Define this channel in sub-group 4	R/W

14.5.18 MDU_REGGRP_CHMAP

Base address: 0x4000_2800
Address offset: 0x50
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular groups. They can be operating in continuous mode or discontinuous mode. 1'b1: Specify that the index channel is in conversion group 1'b0: Specify that the index channel is not in conversion group	R/W

14.5.19 MDU_REGGRP_SGRP_CHMAP_1

Base address: 0x4000_2800
Address offset: 0x54
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 1. 1'b1: Specify that the index channel is in discontinuous group 1 1'b0: Specify that the index channel is not in discontinuous group 1	R/W

14.5.20 MDU_REGGRP_SGRP_CHMAP_2

Base address: 0x4000_2800
Address offset: 0x58
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 2. 1'b1: Specify that the index channel is in discontinuous group 2 1'b0: Specify that the index channel is not in discontinuous group 2	R/W

14.5.21 MDU_REGGRP_SGRP_CHMAP_3

Base address: 0x4000_2800
Address offset: 0x5C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 3. 1'b1: Specify that the index channel is in discontinuous group 3 1'b0: Specify that the index channel is not in discontinuous group 3	R/W

14.5.22 MDU_REGGRP_SGRP_CHMAP_4

Base address: 0x4000_2800
Address offset: 0x60
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 4. 1'b1: Specify that the index channel is in discontinuous group 4 1'b0: Specify that the index channel is not in discontinuous group 4	R/W

14.5.23 MDU_REGGRP_SGRP_CHMAP_5

Base address: 0x4000_2800
Address offset: 0x64
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 5. 1'b1: Specify that the index channel is in discontinuous group 5 1'b0: Specify that the index channel is not in discontinuous group 5	R/W

14.5.24 MDU_REGGRP_SGRP_CHMAP_6

Base address: 0x4000_2800
Address offset: 0x68
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 6. 1'b1: Specify that the index channel is in discontinuous group 6 1'b0: Specify that the index channel is not in discontinuous group 6	R/W

14.5.25 MDU_REGGRP_SGRP_CHMAP_7

Base address: 0x4000_2800
Address offset: 0x6C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 7. 1'b1: Specify that the index channel is in discontinuous group 7 1'b0: Specify that the index channel is not in discontinuous group 7	R/W

14.5.26 MDU_REGGRP_SGRP_CHMAP_8

Base address: 0x4000_2800
Address offset: 0x70
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**MAP[31:0]**: define 31 group channels in regular discontinuous group 8. 1'b1: Specify that the index channel is in discontinuous group 8 1'b0: Specify that the index channel is not in discontinuous group 8	R/W

14.5.27 MDU_REGGRP_CHCR_1

Base address: 0x4000_2800
Address offset: 0x74
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_4[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_4[5:0]**: Channel ID number which is assigned to regular group channel 4	R/W
<23:22>	**CH_AWD_ID_3[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_3[5:0]**: Channel ID number which is assigned to regular group channel 3	R/W
<15:14>	**CH_AWD_ID_2[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_2[5:0]**: Channel ID number which is assigned to regular group channel 2	R/W
<7:6>	**CH_AWD_ID_1[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_1[5:0]**: Channel ID number which is assigned to regular group channel 1	R/W

14.5.28 MDU_REGGRP_CHCR_2

Base address: 0x4000_2800
Address offset: 0x78
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_8[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_8[5:0]**: Channel ID number which is assigned to regular group channel 8	R/W
<23:22>	**CH_AWD_ID_7[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_7[5:0]**: Channel ID number which is assigned to regular group channel 7	R/W
<15:14>	**CH_AWD_ID_6[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_6[5:0]**: Channel ID number which is assigned to regular group channel 6	R/W
<7:6>	**CH_AWD_ID_5[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_5[5:0]**: Channel ID number which is assigned to regular group channel 5	R/W

14.5.29 MDU_REGGRP_CHCR_3

Base address: 0x4000_2800
Address offset: 0x7C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_12[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_12[5:0]**: Channel ID number which is assigned to regular group channel 12	R/W
<23:22>	**CH_AWD_ID_11[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_11[5:0]**: Channel ID number which is assigned to regular group channel 11	R/W
<15:14>	**CH_AWD_ID_10[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_10[5:0]**: Channel ID number which is assigned to regular group channel 10	R/W
<7:6>	**CH_AWD_ID_9[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_9[5:0]**: Channel ID number which is assigned to regular group channel 9	R/W

14.5.30 MDU_REGGRP_CHCR_4

Base address: 0x4000_2800
Address offset: 0x80
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_16[1:0]**: define AWD 2'b00: no analog watch-dog 2'b01: analog watch-dog 1 2'b10: analog watch-dog 2 2'b11: analog watch-dog 3	R/W
<29:24>	**CH_ID_16[5:0]**: Channel ID number which is assigned to regular group channel 16	R/W
<23:22>	**CH_AWD_ID_15[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_15[5:0]**: Channel ID number which is assigned to regular group channel 15	R/W
<15:14>	**CH_AWD_ID_14[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_14[5:0]**: Channel ID number which is assigned to regular group channel 14	R/W
<7:6>	**CH_AWD_ID_13[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_13[5:0]**: Channel ID number which is assigned to regular group channel 13	R/W

14.5.31 MDU_REGGRP_CHCR_5

Base address: 0x4000_2800
Address offset: 0x84
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_20[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_20[5:0]**: Channel ID number which is assigned to regular group channel 20	R/W
<23:22>	**CH_AWD_ID_19[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_19[5:0]**: Channel ID number which is assigned to regular group channel 19	R/W
<15:14>	**CH_AWD_ID_18[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_18[5:0]**: Channel ID number which is assigned to regular group channel 18	R/W
<7:6>	**CH_AWD_ID_17[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_17[5:0]**: Channel ID number which is assigned to regular group channel 17	R/W

14.5.32 MDU_REGGRP_CHCR_6

Base address: 0x4000_2800
Address offset: 0x88
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_24[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_24[5:0]**: Channel ID number which is assigned to regular group channel 24	R/W
<23:22>	**CH_AWD_ID_23[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_23[5:0]**: Channel ID number which is assigned to regular group channel 23	R/W
<15:14>	**CH_AWD_ID_22[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_22[5:0]**: Channel ID number which is assigned to regular group channel 22	R/W
<7:6>	**CH_AWD_ID_21[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_21[5:0]**: Channel ID number which is assigned to regular group channel 21	R/W

14.5.33 MDU_REGGRP_CHCR_7

Base address: 0x4000_2800
Address offset: 0x8C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_28[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_28[5:0]**: Channel ID number which is assigned to regular group channel 28	R/W
<23:22>	**CH_AWD_ID_27[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_27[5:0]**: Channel ID number which is assigned to regular group channel 27	R/W
<15:14>	**CH_AWD_ID_26[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_26[5:0]**: Channel ID number which is assigned to regular group channel 26	R/W
<7:6>	**CH_AWD_ID_25[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_25[5:0]**: Channel ID number which is assigned to regular group channel 25	R/W

14.5.34 MDU_REGGRP_CHCR_8

Base address: 0x4000_2800
Address offset: 0x90
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	**CH_AWD_ID_32[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<29:24>	**CH_ID_32[5:0]**: Channel ID number which is assigned to regular group channel 32	R/W
<23:22>	**CH_AWD_ID_31[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<21:16>	**CH_ID_31[5:0]**: Channel ID number which is assigned to regular group channel 31	R/W
<15:14>	**CH_AWD_ID_30[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<13:8>	**CH_ID_30[5:0]**: Channel ID number which is assigned to regular group channel 30	R/W
<7:6>	**CH_AWD_ID_29[1:0]**: define AWD 2'b00: No analog watch-dog 2'b01: Analog watch-dog 1 2'b10: Analog watch-dog 2 2'b11: Analog watch-dog 3	R/W
<5:0>	**CH_ID_29[5:0]**: Channel ID number which is assigned to regular group channel 29	R/W

14.5.35 MDU_INT_STATUS_1

Base address: 0x4000_2800
Address offset: 0x94
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	**CMP4**: comparator 4 output interrupt flag. Write 0 to clear flag 1'b0: No comparator output transition 1'b1: Comparator output transition	R/W0
<30>	**CMP3**: comparator 3 output interrupt flag. Write 0 to clear flag 1'b0: No comparator output transition 1'b1: Comparator output transition	R/W0
<29>	**CMP2**: comparator 2 output interrupt flag. Write 0 to clear flag 1'b0: No comparator output transition 1'b1: Comparator output transition	R/W0
<28>	**CMP1**: comparator 1 output interrupt flag. Write 0 to clear flag 1'b0: No comparator output transition 1'b1: Comparator output transition	R/W0
<27:22>	Reserved	R/W
<21>	**REGGRP_OW**: Over write error flag of regular group. Write 0 to clear flag. 1'b0: No error 1'b1: Result data register over write error	R/W0
<20>	**INJGRP_OW**: Over write error flag of injection group. Write 0 to clear flag. 1'b0: No error 1'b1: Result data register over write error	R/W0
<19>	**INJ_OW_4**: Over write error flag of injection group channel 4. Write 0 to clear flag. 1'b0: No error 1'b1: Result data register over write error	R/W0
<18>	**INJ_OW_3**: Over write error flag of injection group channel 3. Write 0 to clear flag. 1'b0: No error 1'b1: Result data register over write error	R/W0
<17>	**INJ_OW_2**: Over write error flag of injection group channel 2. Write 0 to clear flag. 1'b0: No error 1'b1: Result data register over write error	R/W0
<16>	**INJ_OW_1**: Over write error flag of injection group channel 1. Write 0 to clear flag. 1'b0: No error	R/W0

	1’b1: Result data register over write error	
<15:12>	Reserved	R/W
<11>	**INJ_AWD_4**: AWD flag for injection group channel 4. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<10>	**INJ_AWD_3**: AWD flag for injection group channel 3. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<9>	**INJ_AWD_2**: AWD flag for injection group channel 2. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<8>	**INJ_AWD_1**: AWD flag for injection group channel 1. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<7:6>	Reserved	R/W
<5>	**REG_EOS**: interrupt EOS flag of regular group. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<4>	**INJ_EOS**: interrupt EOS flag of injection group. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<3>	**INJ_EOC_4**: interrupt EOC flag of injection group channel 4. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<2>	**INJ_EOC_3**: interrupt EOC flag of injection group channel 3. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<1>	**INJ_EOC_2**: interrupt EOC flag of injection group channel 2. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W0
<0>	**INJ_EOC_1**: interrupt EOC flag of injection group channel 1. Write 0 to clear flag. 1’b0: No interrupt request generated 1’b1: Interrupt request generated	R/W

14.5.36 MDU_INT_STATUS_2

Base address: 0x4000_2800
Address offset: 0x98
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**REG_OW_x[15:0]**: Overwrite error flag of regular group channel x [1 to 16]. Write 0 to clear flag 1'b0: Overwrite error flag not set 1'b1: Overwrite error flag set	R/W0
<15:0>	**REG_EOC_x[15:0]**: EOC flag of regular group channel x [1 to 16]. Write 0 to clear flag 1'b0: EOC flag not set 1'b1: EOC flag set	R/W0

14.5.37 MDU_INT_STATUS_3

Base address: 0x4000_2800
Address offset: 0x9C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**REG_OW_x[15:0]**: Overwrite error flag of regular group channel x [17 to 32]. Write 0 to clear flag 1'b0: Overwrite error flag not set 1'b1: Overwrite error flag set	R/W0
<15:0>	**REG_EOC_x[15:0]**: EOC flag of regular group channel x [17 to 32]. Write 0 to clear flag 1'b0: EOC flag not set 1'b1: EOC flag set	R/W0

14.5.38 MDU_INJ_RESULT_1

Base address: 0x4000_2800
Address offset: 0xA0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH2[15:0]**: ADC result of channel in injection group channel 2	R
<15:0>	**RSLT_CH1[15:0]**: ADC result of channel in injection group channel 1	R

14.5.39 MDU_INJ_RESULT_2

Base address: 0x4000_2800
Address offset: 0xA4
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH4[15:0]**: ADC result of channel in injection group channel 4	R
<15:0>	**RSLT_CH3[15:0]**: ADC result of channel in injection group channel 3	R

14.5.40 MDU_REG_RESULT_1

Base address: 0x4000_2800
Address offset: 0xA8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH2[15:0]**: ADC result of channel in regular group channel 2	R
<15:0>	**RSLT_CH1[15:0]**: ADC result of channel in regular group channel 1	R

14.5.41 MDU_REG_RESULT_2

Base address: 0x4000_2800
Address offset: 0xAC
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH4[15:0]**: ADC result of channel in regular group channel 4	R
<15:0>	**RSLT_CH3[15:0]**: ADC result of channel in regular group channel 3	R

14.5.42 MDU_REG_RESULT_3

Base address: 0x4000_2800
Address offset: 0xB0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH6[15:0]**: ADC result of channel in regular group channel 6	R
<15:0>	**RSLT_CH5[15:0]**: ADC result of channel in regular group channel 5	R

14.5.43 MDU_REG_RESULT_4

Base address: 0x4000_2800
Address offset: 0xB4
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH8[15:0]**: ADC result of channel in regular group channel 8	R
<15:0>	**RSLT_CH7[15:0]**: ADC result of channel in regular group channel 7	R

14.5.44 MDU_REG_RESULT_5

Base address: 0x4000_2800
Address offset: 0xB8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH10[15:0]**: ADC result of channel in regular group channel 10	R
<15:0>	**RSLT_CH9[15:0]**: ADC result of channel in regular group channel 9	R

14.5.45 MDU_REG_RESULT_6

Base address: 0x4000_2800
Address offset: 0xBC
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH12[15:0]**: ADC result of channel in regular group channel 12	R
<15:0>	**RSLT_CH11[15:0]**: ADC result of channel in regular group channel 11	R

14.5.46 MDU_REG_RESULT_7

Base address: 0x4000_2800
Address offset: 0xC0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH14[15:0]**: ADC result of channel in regular group channel 14	R
<15:0>	**RSLT_CH13[15:0]**: ADC result of channel in regular group channel 13	R

14.5.47 MDU_REG_RESULT_8

Base address: 0x4000_2800
Address offset: 0xC4
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH16[15:0]**: ADC result of channel in regular group channel 16	R
<15:0>	**RSLT_CH15[15:0]**: ADC result of channel in regular group channel 15	R

14.5.48 MDU_REG_RESULT_9

Base address: 0x4000_2800
Address offset: 0xC8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH18[15:0]**: ADC result of channel in regular group channel 18	R
<15:0>	**RSLT_CH17[15:0]**: ADC result of channel in regular group channel 17	R

14.5.49 MDU_REG_RESULT_10

Base address: 0x4000_2800
Address offset: 0xCC
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH20[15:0]**: ADC result of channel in regular group channel 20	R
<15:0>	**RSLT_CH19[15:0]**: ADC result of channel in regular group channel 19	R

14.5.50 MDU_REG_RESULT_11

Base address: 0x4000_2800
Address offset: 0xD0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH22[15:0]**: ADC result of channel in regular group channel 22	R
<15:0>	**RSLT_CH21[15:0]**: ADC result of channel in regular group channel 21	R

14.5.51 MDU_REG_RESULT_12

Base address: 0x4000_2800
Address offset: 0xD4
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH24[15:0]**: ADC result of channel in regular group channel 24	R
<15:0>	**RSLT_CH23[15:0]**: ADC result of channel in regular group channel 23	R

14.5.52 MDU_REG_RESULT_13

Base address: 0x4000_2800
Address offset: 0xD8
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH26[15:0]**: ADC result of channel in regular group channel 26	R
<15:0>	**RSLT_CH25[15:0]**: ADC result of channel in regular group channel 25	R

14.5.53 MDU_REG_RESULT_14

Base address: 0x4000_2800
Address offset: 0xDC
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH28[15:0]**: ADC result of channel in regular group channel 28	R
<15:0>	**RSLT_CH27[15:0]**: ADC result of channel in regular group channel 27	R

14.5.54 MDU_REG_RESULT_15

Base address: 0x4000_2800
Address offset: 0xE0
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH30[15:0]**: ADC result of channel in regular group channel 30	R
<15:0>	**RSLT_CH29[15:0]**: ADC result of channel in regular group channel 29	R

14.5.55 MDU_REG_RESULT_16

Base address: 0x4000_2800
Address offset: 0xE4
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**RSLT_CH32[15:0]**: ADC result of channel in regular group channel 32	R
<15:0>	**RSLT_CH31[15:0]**: ADC result of channel in regular group channel 31	R

14.5.56 MDU_AWD_CR_1

Base address: 0x4000_2800
Address offset: 0xF0
Type: Read/Write
Reset value: 0x0FFF_0000

Bit	Description	Type
<31:30>	Reserved	R/W
<29:28>	**AWD_CR_CFG_1[1:0]**: analog watch-dog 1 monitor zone configuration 2'b00: Do not generate flag 2'b01: ADC out code is lower than the low threshold 2'b10: ADC out code is between the low and high threshold 2'b11: ADC out code is higher than the high threshold	R/W
<27:16>	**AWD_THR_HIGH_1[11:0]**: analog watch-dog 1 high threshold value	R/W
<15:12>	Reserved	R/W
<11:0>	**AWD_THR_LOW_1[11:0]**: analog watch-dog 1 low threshold value	R/W

14.5.57 MDU_AWD_CR_2

Base address: 0x4000_2800
Address offset: 0xF4
Type: Read/Write
Reset value: 0x0FFF_0000

Bit	Description	Type
<31:30>	Reserved	R/W
<29:28>	**AWD_CR_CFG_2[1:0]**: analog watch-dog 2 monitor zone configuration 2'b00: Do not generate flag 2'b01: ADC out code is lower than the low threshold 2'b10: ADC out code is between the low and high threshold 2'b11: ADC out code is higher than the high threshold	R/W
<27:16>	**AWD_THR_HIGH_2[11:0]**: analog watch-dog 2 high threshold value	R/W
<15:12>	Reserved	R/W
<11:0>	**AWD_THR_LOW_2[11:0]**: analog watch-dog 2 low threshold value	R/W

14.5.58 MDU_AWD_CR_3

Base address: 0x4000_2800
Address offset: 0xF8
Type: Read/Write
Reset value: 0x0FFF_0000

Bit	Description	Type
<31:30>	Reserved	R/W
<29:28>	**AWD_CR_CFG_3[1:0]**: analog watch-dog 3 monitor zone configuration 2'b00: Do not generate flag 2'b01: ADC out code is lower than the low threshold 2'b10: ADC out code is between the low and high threshold 2'b11: ADC out code is higher than the high threshold	R/W
<27:16>	**AWD_THR_HIGH_3[11:0]**: analog watch-dog 3 high threshold value	R/W
<15:12>	Reserved	R/W
<11:0>	**AWD_THR_LOW_3[11:0]**: analog watch-dog 3 low threshold value	R/W

14.5.59 MDU_INT_STATUS_4

Base address: 0x4000_2800
Address offset: 0xFC
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**REG_AWD_x[31:0]**: analog watch-dog interrupt flag of regular group channel [1 to 32]. Write 0 to clear flag. 1'b0: AWD interrupt flag not set 1'b1: AWD interrupt flag set	R/W0

14.5.60 MDU register map

The base address of the MDU registers is 0x4000_2800.
The following table gives the MDU register map and reset values.

Table 36 MDU register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	MDU_PDCTRL	Res	Res	Res	PDCTRL_TEMPSPD	PDCTRL_PGAEN4	PDCTRL_PGAEN3	PDCTRL_PGAEN2	PDCTRL_PGAEN1	Res	Res	PDCTRL_ADCBUF_EN	PDCTRL_ADCPD	Res	Res	Res	PDCTRL_CMP4PD	Res	Res	Res	PDCTRL_CMP3PD	Res	Res	Res	PDCTRL_CMP2PD	Res	Res	Res	PDCTRL_CMP1PD	Res	Res	Res	PDCTRL_DACPD
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
0x04	MDU_DAC_CFG		DACCFG_DAC2_TSTM[1:0]		DACCFG_DAC1_TSTM[1:0]	Res	Res	Res	DACCFG_DAC2BBP	Res	Res	Res	DACCFG_DAC1BBP	Res	Res	DACCFG_REFSEL[1:0]		DACCFG_DAC2ADD[7:0]								DACCFG_DAC1ADD[7:0]							
	Reset value	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	MDU_PGA_CFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PGACFG_ADC_VCM_HVDD				PGACFG_ADC_VCM_TRIM[3:0]	Res			PGACFG_PGA4GAIN[2:0]	Res			PGACFG_PGA3GAIN[2:0]	Res			PGACFG_PGA2GAIN[2:0]	Res			PGACFG_PGA1GAIN[2:0]
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x0C	MDU_ADC_CFG	ADCCFG_ADCOUTALIGN		ADCCFG_ADCTSMPLE[1:0]	ADCCFG_AMPMILLER	ADCCFG_PGA_MILLER4[1:0]		ADCCFG_PGA_MILLER3[1:0]		ADCCFG_PGA_MILLER2[1:0]			ADCCFG_PGAMILLER1[1:0]		ADCCFG_AMPGAIN[1:0]		ADCCFG_ADCDATAW[1:0]	ADCCFG_MSAMP_EN	Res	Res	Res					ADCCFG_RSTDLY[3:0]	ADCCFG_MCONVMODE	ADCCFG_ADCC_EN					ADCCFG_MCHSLCT[5:0]
	Reset value	0	0	1	1	1	0	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1
0x10	MDU_COMP_CFG1	Res	Res	Res	CMPCFG1_PSRC_4	Res	Res		CMPCFG1_NSRC_4 [1:0]	Res	Res	Res	CMPCFG1_PSRC_3	Res	Res		CMPCFG1_NSRC_3 [1:0]	Res	Res	Res	CMPCFG1_PSRC_2	Res	Res		CMPCFG1_NSRC_2[1:0]	Res	Res	Res	CMPCFG1_PSRC_1	Res	Res		CMPCFG1_NSRC_1[1:0]
	Reset value	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
0x14	MDU_COMP_CFG2	Res			CMPCFG2_SPD_4[2:0]	Res	Res		CMPCFG2_HYS_4[1:0]	Res			CMPCFG2_SPD_3[2:0]	Res	Res		CMPCFG2_HYS_3[1:0]	Res			CMPCFG2_SPD_2[2:0]	Res	Res		CMPCFG2_HYS_2[1:0]	Res			CMPCFG2_SPD_2[2:0]	Res	Res		CMPCFG2_HYS_1[1:0]
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18	MDU_COMP_CFG3	CMPCFG3_PGA4_NSRC[1:0]		CMPCFG3_PGA4_PSRC[1:0]		CMPCFG3_PGA3_NSRC[1:0]		CMPCFG3_PGA3_PSRC[1:0]		CMPCFG3_PGA2_NSRC[1:0]		CMPCFG3_PGA2_PSRC[1:0]		CMPCFG3_PGA1_NSRC[1:0]		CMPCFG3_PGA1_PSRC[1:0]		Res	Res	CMPCGF3_CMP_BPFLG	CMPCFG3_CMP4_FTRG	CMPCFG3_CMP4_RTRG	CMPCFG3_CMP3_FTRG	CMPCFG3_CMP3_RTRG	CMPCFG3_CMP2_FTRG	CMPCFG3_CMP2_RTRG	CMPCFG3_CMP1_FTRG	CMPCFG3_CMP1_RTRG	CMPCFG3_PRLT	CMPCFG3_OPOL[3:0]			
	Reset value	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
0x28	ADC_CALIB	ADCCALIB_COR_EN	Res	ADCCALIB_CODE_OFFSET[13:0]														Res	ADCCALIB_CODE_GAIN[14:0]														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	MDU_DMACFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REG_DMA_EN	INJ_DMA_EN
	Reset value	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
0x30	MDU_CFG	REG_EOS_EN	REG_EOC_EN	INJ_EOS_EN	INJ_EOC_EN	REG_INT_POL	INJ_INT_POL	RST_PW[3:0]				SMPL_CYC[5:0]						ADCC_EN	ADC_SWTRG	ADC_RST	ADC_STRT	SMPL_MODE	REGGRP_EN	INJGRP_EN	DISCHARGE_EN	DISC_DLY[7:0]							
	Reset value	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0
0x34	MDU_INJGRP_TRG	Res	Res	INJ_OPMODE[1:0]		REG_DISC_GMODE[3:0]				INJGRP_TRGCFG_4[1:0]		INJGRP_TRGCFG_3[1:0]		INJGRP_TRGCFG_2[1:0]		INJGRP_TRGCFG_1[1:0]		INJGRP_TRGSCR_4[3:0]				INJGRP_TRGSCR_3[3:0]				INJGRP_TRGSCR_2[3:0]				INJGRP_TRGSCR_1[3:0]			
	Reset value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
0x38	ADC_REGGRP_TRG	REGGRP_TRGSCR_8[3:0]				REGGRP_TRGSCR_7[3:0]				REGGRP_TRGSCR_6[3:0]				REGGRP_TRGSCR_5[3:0]				REGGRP_TRGSCR_4[3:0]				REGGRP_TRGSCR_3[3:0]				REGGRP_TRGSCR_2[3:0]				REGGRP_TRGSCR_1[3:0]			
	Reset value	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
0x3C	ADC_REFGRP_TRG CFG		REG_OPMODE[1:0]	Res	REGGRP_CSTRT	Res	Res	Res	Res	REGGRP_TRGCFG_8[1:0]		REGGRP_TRGCFG_7[1:0]		REGGRP_TRGCFG_6[1:0]		REGGRP_TRGCFG_5[1:0]		REGGRP_TRGCFG_4[1:0]		REGGRP_TRGCFG_3[1:0]		REGGRP_TRGCFG_2[1:0]		REGGRP_TRGCFG_1[1:0]		Res	Res	Res	Res	REGGRP_CFG[3:0]			
	Reset value	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	1	0	0	0	1	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	ADC_INJ_CHCR_1	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_AWD_ID_1[1:0]									INJGRP_CHCR_SMP_CYC_1[7:0]	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_ADCCH_ID_1[5:0]						Res	INJGRP_CHCR_GRP_ID_1[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
0x44	ADC_INJ_CHCR_2	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_AWD_ID_2[1:0]									INJGRP_CHCR_SMP_CYC_2[7:0]	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_ADCCH_ID_2[5:0]						Res	INJGRP_CHCR_GRP_ID_2[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0
0x48	ADC_INJ_CHCR_3	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_AWD_ID_3[1:0]									INJGRP_CHCR_SMP_CYC_3[7:0]	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_ADCCH_ID_3[5:0]						Res	INJGRP_CHCR_GRP_ID_3[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
0x4C	ADC_INJ_CHCR_4	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_AWD_ID_4[1:0]									INJGRP_CHCR_SMP_CYC_4[7:0]	Res	Res	Res	Res	Res	Res	INJGRP_CHCR_ADCCH_ID_4[5:0]						Res	INJGRP_CHCR_GRP_ID_4[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0
0x50	ADC_REG_CHMAP	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x54	ADC_REG_DC_CHMAP_1	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x58	ADC_REG_DC_CHMAP_2	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	ADC_REG_DC_CHMAP_3	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x60	ADC_REG_DC_CHMAP_4	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x64	ADC_REG_DC_CHMAP_5	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x68	ADC_REG_DC_CHMAP_6	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x6C	ADC_REG_DC_CHMAP_7	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x70	ADC_REG_DC_CHMAP_8	REG_CH_EN_32	REG_CH_EN_31	REG_CH_EN_30	REG_CH_EN_29	REG_CH_EN_28	REG_CH_EN_27	REG_CH_EN_26	REG_CH_EN_25	REG_CH_EN_24	REG_CH_EN_23	REG_CH_EN_22	REG_CH_EN_21	REG_CH_EN_20	REG_CH_EN_19	REG_CH_EN_18	REG_CH_EN_17	REG_CH_EN_16	REG_CH_EN_15	REG_CH_EN_14	REG_CH_EN_13	REG_CH_EN_12	REG_CH_EN_11	REG_CH_EN_10	REG_CH_EN_9	REG_CH_EN_8	REG_CH_EN_7	REG_CH_EN_6	REG_CH_EN_5	REG_CH_EN_4	REG_CH_EN_3	REG_CH_EN_2	REG_CH_EN_1
	Reset value	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
0x74	ADC_REG_CHCR_1	CH_ID_4[7:0]								CH_ID_3[7:0]								CH_ID_2[7:0]								CH_ID_1[7:0]							
	Reset value	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
0x78	ADC_REG_CHCR_2	CH_ID_8[7:0]								CH_ID_7[7:0]								CH_ID_6[7:0]								CH_ID_5[7:0]							
	Reset value	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
0x7C	ADC_REG_CHCR_3	CH_ID_12[7:0]								CH_ID_11[7:0]								CH_ID_10[7:0]								CH_ID_9[7:0]							
	Reset value	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
0x80	ADC_REG_CHCR_4	CH_ID_16[7:0]								CH_ID_15[7:0]								CH_ID_14[7:0]								CH_ID_13[7:0]							
	Reset value	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
0x84	ADC_REG_CHCR_5	CH_ID_20[7:0]								CH_ID_19[7:0]								CH_ID_18[7:0]								CH_ID_17[7:0]							
	Reset value	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
0x88	ADC_REG_CHCR_6	CH_ID_24[7:0]								CH_ID_23[7:0]								CH_ID_22[7:0]								CH_ID_21[7:0]							
	Reset value	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
0x8C	ADC_REG_CHCR_7	CH_ID_28[7:0]								CH_ID_27[7:0]								CH_ID_26[7:0]								CH_ID_25[7:0]							
	Reset value	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
0x90	ADC_REG_CHCR_8	CH_ID_32[7:0]								CH_ID_31[7:0]								CH_ID_30[7:0]								CH_ID_29[7:0]							
	Reset value	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
0x94	ADC_INT_STATUS_1	CMP4	CMP3	CMP2	CMP1	Res	Res	Res	Res	Res	Res	REGGRP_OW	INJGRP_OW	INJ_OW_4	INJ_OW_3	INJ_OW_2	INJ_OW_1	Res	Res	Res	Res	INJ_AWD_4	INJ_AWD_3	INJ_AWD_2	INJ_AWD_1	Res	Res	REG_EOS	INJ_EOS	INJ_EOC_4	INJ_EOC_3	INJ_EOC_2	INJ_EOC_1
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x98	ADC_INT_STATUS_2	REG_OW_16	REG_OW_15	REG_OW_14	REG_OW_13	REG_OW_12	REG_OW_11	REG_OW_10	REG_OW_9	REG_OW_8	REG_OW_7	REG_OW_6	REG_OW_5	REG_OW_4	REG_OW_3	REG_OW_2	REG_OW_1	REG_EOC_16	REG_EOC_15	REG_EOC_14	REG_EOC_13	REG_EOC_12	REG_EOC_11	REG_EOC_10	REG_EOC_9	REG_EOC_8	REG_EOC_7	REG_EOC_6	REG_EOC_5	REG_EOC_4	REG_EOC_3	REG_EOC_2	REG_EOC_1
	Reset value	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
0x9C	ADC_INT_STATUS_3	REG_OW_32	REG_OW_31	REG_OW_30	REG_OW_29	REG_OW_28	REG_OW_27	REG_OW_26	REG_OW_25	REG_OW_24	REG_OW_23	REG_OW_22	REG_OW_21	REG_OW_20	REG_OW_19	REG_OW_18	REG_OW_17	REG_EOC_32	REG_EOC_31	REG_EOC_30	REG_EOC_29	REG_EOC_28	REG_EOC_27	REG_EOC_26	REG_EOC_25	REG_EOC_24	REG_EOC_23	REG_EOC_22	REG_EOC_21	REG_EOC_20	REG_EOC_19	REG_EOC_18	REG_EOC_17
	Reset value	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
0xA0	ADC_INJ_RESULT_1	RSLT_CH_2[15:0]																RSLT_CH_1[15:0]															
	Reset value	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
0xA4	ADC_INJ_RESULT_2	RSLT_CH_4[15:0]																RSLT_CH_3[15:0]															
	Reset value	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
0xA8	ADC_REG_RESULT_1	RSLT_CH_2[15:0]																RSLT_CH_1[15:0]															
	Reset value	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
0xAC	ADC_REG_RESULT_2	RSLT_CH_4[15:0]																RSLT_CH_3[15:0]															
	Reset value	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
0xB0	ADC_REG_RESULT_3	RSLT_CH_6[15:0]																RSLT_CH_5[15:0]															
	Reset value	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
0xB4	ADC_REG_RESULT_4	RSLT_CH_8[15:0]																RSLT_CH_7[15:0]															
	Reset value	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
0xB8	ADC_REG_RESULT_5	RSLT_CH_10[15:0]																RSLT_CH_9[15:0]															
	Reset value	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
0xBC	ADC_REG_RESULT_6	RSLT_CH_12[15:0]																RSLT_CH_11[15:0]															
	Reset value	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
0xC0	ADC_REG_RESULT_7	RSLT_CH_14[15:0]																RSLT_CH_31[15:0]															
	Reset value	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
0xC4	ADC_REG_RESULT_8	RSLT_CH_16[15:0]																RSLT_CH_15[15:0]															
	Reset value	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
0xC8	ADC_REG_RESULT_9	RSLT_CH_18[15:0]																RSLT_CH_17[15:0]															
	Reset value	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
0xCC	ADC_REG_RESULT_10	RSLT_CH_20[15:0]																RSLT_CH_19[15:0]															
	Reset value	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
0xD0	ADC_REG_RESULT_11	RSLT_CH_22[15:0]																RSLT_CH_21[15:0]															
	Reset value	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
0xD4	ADC_REG_RESULT_12	RSLT_CH_24[15:0]																RSLT_CH_23[15:0]															
	Reset value	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
0xD8	ADC_REG_RESULT_13	RSLT_CH_26[15:0]																RSLT_CH_25[15:0]															
	Reset value	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
0xDC	ADC_REG_RESULT_14	RSLT_CH_28[15:0]																RSLT_CH_27[15:0]															
	Reset value	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
0xE0	ADC_REG_RESULT_15	RSLT_CH_30[15:0]																RSLT_CH_29[15:0]															
	Reset value	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
0xE4	ADC_REG_RESULT_16	RSLT_CH_32[15:0]																RSLT_CH_31[15:0]															
	Reset value	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
0xF0	ADC_AWD_CR_1	Res	Res	AWD_CR_CFG_1[1:0]													AWD_THR_HIGH_1[11:0]	Res	Res	Res	Res	AWD_THR_LOW_1[11:0]											
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xF4	ADC_AWD_CR_2	Res	Res	AWD_CR_CFG_2[1:0]													AWD_THR_HIGH_2[11:0]																AWD_THR_LOW_2[11:0]
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF8	ADC_AWD_CR_3	Res	Res	AWD_CR_CFG_3[1:0]													AWD_THR_HIGH_3[11:0]																AWD_THR_LOW_3[11:0]
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFC	ADC_INT_STATUS_4	REG_AWD_31	REG_AWD_30	REG_AWD_29	REG_AWD_28	REG_AWD_27	REG_AWD_26	REG_AWD_25	REG_AWD_24	REG_AWD_23	REG_AWD_22	REG_AWD_21	REG_AWD_20	REG_AWD_19	REG_AWD_18	REG_AWD_17	REG_AWD_16	REG_AWD_15	REG_AWD_14	REG_AWD_13	REG_AWD_12	REG_AWD_11	REG_AWD_10	REG_AWD_9	REG_AWD_8	REG_AWD_7	REG_AWD_6	REG_AWD_5	REG_AWD_4	REG_AWD_3	REG_AWD_2	REG_AWD_1	REG_AWD_0
	Reset value	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

15 Advanced timer (TIMA)

15.1 Introduction

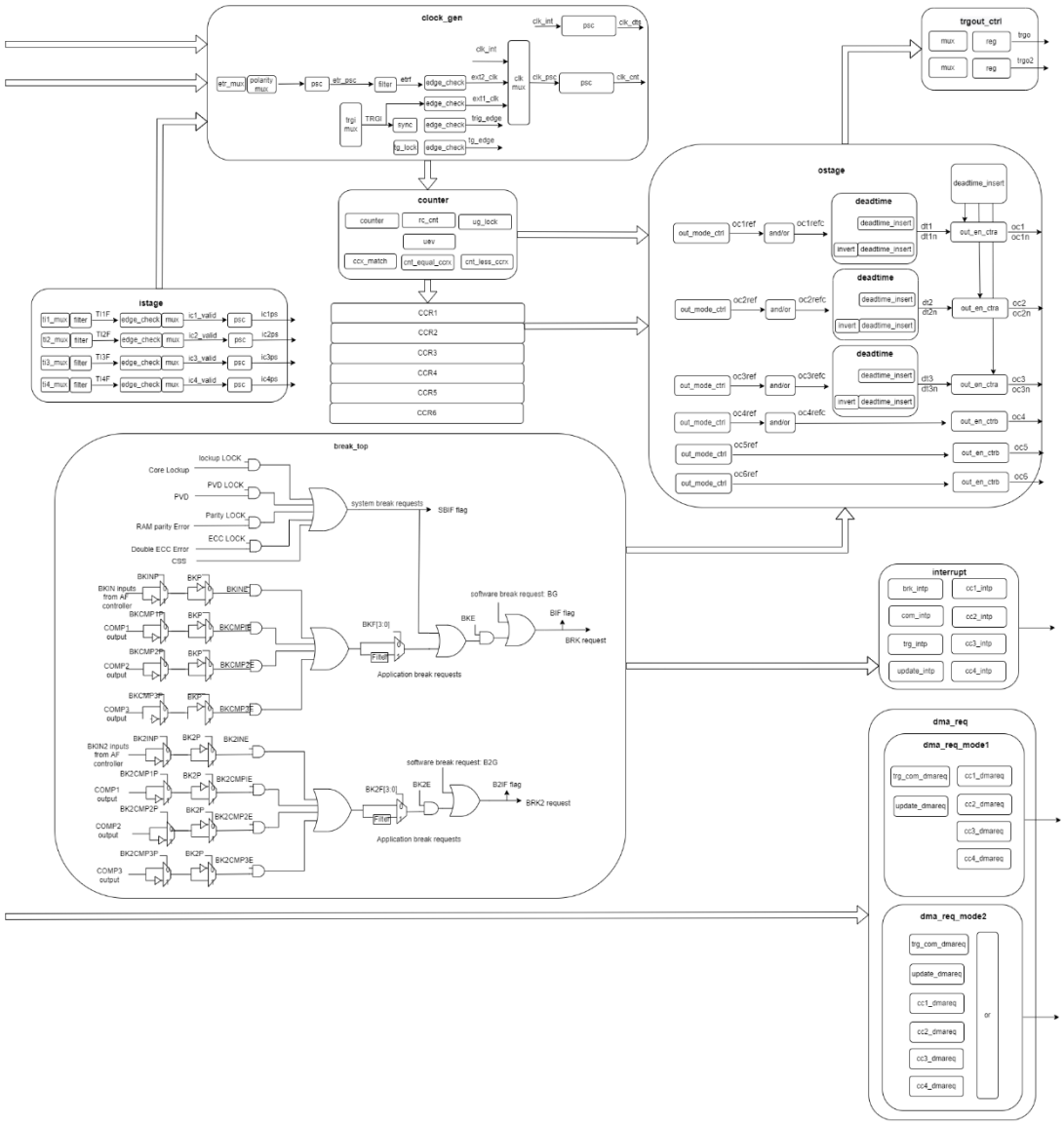


Figure 32 Advanced timer block diagram

15.2 TIMA main features

- 16-bit up, down, up and down auto-load counter
- 16-bit programming prescaler
- Up to 6 independent channels for:
  - Input capture (excluding channel 5 and channel 6)
  - Output compare
  - PWM generation (edge and center-aligned)
  - One-pulse mode output
- Complementary outputs with programmable dead-time
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Repetition counter to update the timer registers only after a given number of cycles of the counter
- 2 bidirectional break inputs
- Supports incremental encoder and hall-sensor circuitry for positioning perposes
- Interrupts generator
- Configure by DMA
- Trigger input for external clock

15.3 TIMA signal descriptions

The advanced timer signals are shown in the following table.

Table 37 TIMA signal descriptions

Signal	Description	I/O
ETR	External trigger input	I
CH1_IN	Channel 1 input	I
CH2_IN	Channel 2 input	I
CH3_IN	Channel 3 input	I
CH4_IN	Channel 4 input	I
BRK_IN	Break I input	I
BRK2_IN	Break 2 input	I
CH1_OUT_P	Channel 1 positive output	O
CH1_OUT_N	Channel 1 negitive output	O
CH2_OUT_P	Channel 2 positive output	O
CH2_OUT_N	Channel 2 negitive output	O
CH3_OUT_P	Channel 3 positive output	O
CH3_OUT_N	Channel 3 negitive output	O

CH4_OUT	Channel 4 output	O
BRK_OUT	Break 1 output	O
BRK2_OUT	Break 2 output	O

15.4 TIMA function description

15.4.1 Basic Timer

Timer may be set as basic timer and also need to be configured basic timer for advanced features. The basic timer features are:

1. The prescaler used to divided timer clock.
2. Counter mode which are count up, count down and count aligned center.
3. The period of counter.
4. The divider for dead-time and sampling clock
5. The repetition of counter
6. Auto-reload mode

Using flow,

1. Configure counter mode, clock division, auto-reload preload, auto-load value, prescaler value, repetition counter value for Timer.
2. Write EGR_UG to reset timer
3. Configure clock source for Timer
4. Configure Timer to master mode and set TRGO source
5. Enable update interrupt
6. Enable counter

15.4.2 Slave mode configurate

15.4.3 Master mode configurate

15.4.4 Break configurate

15.4.4.1 Break control

15.4.4.2 Bi-direction break

15.4.5 Output mode

15.4.5.1 Output compare

When timer channel works as output compare, the following characters would ne configurated:

1. Output mode
2. Compare value
3. Output polarity
4. Complementary output polarity
5. Output fast mode
6. Output pin state in idle state

7. Complementary output pin state in idle state

15.4.5.2 PWM output

When timer channel works as PWM output, the characters which would be configurated are the same as output compare.

Using flow,

1. Configure clock source and divider for TIMA by ICG
2. Configure counter mode, clock division, auto-reload preload, auto-load value, prescaler value, repetition counter value for Timer.
3. Write EGR_UG to reset timer
4. Configure Timer to master mode and set TRGO and TRGO2 source
5. Configure channels with output compare mode, output polarity, output N polarity, output and output N idle state, capture compare value, enable channel preload, and output enable.
6. Set break state, polarity filter, input features, run mode off state, idle mode off state for timer
7. Set deadtime for timer
8. Enable MOE (main output enable)
9. Enable timer

15.4.5.3 One pulse mode

Using flow,

1. Select input source (TIM_ISCT)
2. Mapping input signal (TIM_CHMDx)
3. Configure signal polarity
4. Select trigger (TIM_SLVMD.TS)
5. Set in one pulse mode (TIM_SLVMD.SMS)

15.4.5.4 Commutation control

15.4.6 Input mode

15.4.6.1 Input capture

Using flow,

1. Configure clock source and divider for TIMA by ICG
2. Configure counter mode, clock division, auto-reload preload, auto-load value, prescaler value, repetition counter value for Timer.
3. Write EGR_UG to reset timer
4. Configure Timer to master mode and set TRGO and TRGO2 source
5. Configure channels with input filter, polarity, ptrscaler, and enable
6. Enable CCx interrupts
7. Enable input capture channel
8. Enable counter

15.4.6.2 PWM intput

15.4.6.3 Encoder interface

When timer configure as channel 1 input edges, channel 2 input edges or both channel 1 and channel 2 edges, encoder interface may be selected by set SLVMD.SMS[16,2:0]. When SMS = '0001', the counter is counting on channel 2 input edges only. When SMS = '0010' the counter is counting on channel 1 input edges only and when SMS = '0111' the counter is counting both channel 1 input and channel 2 input edges.

Select channel 1 input and channel 2 input polarity by programming the CHEP.CC1P[1] and CHEP.CC2P[5]. Note: CHEP.CC1NP[3] and CHEP.CC2NP[7] must be kept low.

When needed, the input filer of channel 1 (CHMD1.IC1F[7:4]) and channel 2 (CHMD1.IC2F[15:12]) can be programmed.

Notes: the external clock mode 2 (SLVMD.ECE[14] = 0) must not be selected together with encode mode.

The two inputs are used to interface to a quadrature encoder (refer to the following table).

Table 38 Counting direction

Edge (SMS)	Level on opposite signal (CC1P/CC2P)	Channel 1 signal		Channel 2 signal	
		Rising	Falling	Rising	Falling
Counting on channel 1	High	Down	Up	No Count	No Count
	Low	Up	Down	No Count	No Count
Counting on channel 2	High	No Count	No Count	Up	Down
	Low	No Count	No Count	Down	Up
Counting on channel 1 and 2	High	Down	Up	Up	Down
	Low	Up	Down	Down	Up

Using flow,

1. Mapping channel 1 and channel 2 to encoder interface.
   Configure
   TIM_CHMD1.CC1S and TIM_CHMD1.CC2S,
   TIM_CHMD1.IC1F and TIM_CHMD1.IC2F (optional).
2. Configure input signal polarity.
   Configure
   TIM_CHEP.CC1P and TIM_CHEP.CC2P,
   TIM_CHEP.CC1NP = 0 and TIM)CHEP.CC2NP = 0.
3. Set encoder mode.
   Program
   TIM_SLVMD.SMS = 1/2/3

4. Enable counter (TIM_CTRL1.CEN = 1)

15.4.6.4 Hall sensor interface

Using flow,

1. Connect Hall inputs to channel 1, 2, and 3 input.
2. Configure to hall sensor interface mode: the timer capture the 3 channel inputs connected through a XOR to channel 1 input.

   Program

   TIM_CTRL2.TIS[7] = 1.
3. configure timer.

   Configure

   Set TIM_PRD to the max value,

   Set TIM_PSCR to get a maximum counter period longer than the time between 2 changes on the sensors,

   TIM_SLVMD.SMS[16,2:0] = '0100' (run in rest mode),

   TIM_SLVMD.TS[21:20.6:4] = '0100' (trigger by channel 1 input edge detector).
4. Configure channel 1 in capture mode.

   Configure

   TIM_CHMD1.CCIS[1:0] = '11' (capture signal is TRC),

   When needed, the input filer of channel 1 (CHMD1.IC1F[7:4]) can be programmed.
5. Used in output mode to generate a pulse which triggers another timer to driver motor. This pulse must be a positive pulse and generate after a programmed delay (in output compare or PWM mode). This pulse sent by TRGO.

   Example:

   Configure channel 2 in PWM 2 mode with delay.

   TIM_CHMD1.OC2M[24,14:12] = '111', TIM_CHMD1.CC2S[9:8] = '00'.

   Select OC2REF as trigger output on TRGO.

   TIM_CTRL2.MMS[6:4] = '101'

15.4.7 UIF bit remapping

The TIM_CTRL1.UIFREMAP[11] can enable a continuous copy of the Update Interrupt Flag (TIM_STATUS.UIF[0]) into the timer's counter bit (TIM_CNTR.UIFCP[31]). This allows both the counter value and a potential roll-over condition signaled by the UIFCP flag to be read in an atomic way.

15.4.8 Timer synchronization

Configure TRGO source

15.4.9 ADC synchronization

Configure TRGO2 source

15.4.10 Clock source selection

15.4.11 DMA control

15.4.12 Motor control

15.4.12.1 Sensorless motor control

Using flow,

1. Initialize
   a. MDU initialize
   b. TIMA initialize
   c. Motor control initialize
      i. Set system tick
      ii. Set pad
      iii. MCBoot

15.4.12.2 Six-step drive

Using flow,

1. Configure complementary output channels
2. Enable preload for OCxM, CCxE and CCxNE
3. Configure all channels for the next step.
4. COM event can be triggered by software or hardware (TRGI)

15.4.12.3 Speed control

TBD

15.1 TIMA registers

15.1.1 TIM_CTRL1 (Timer control register 1)

Base address: 0x4000_1000
Address offset: 0x00
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:12>	Reserved.	R/W
<11>	**UIFREMAP**: UIF status bit remapping 1'b0: No remapping. UIF status bit is not copied to TIM_CNTR bit 31. 1'b1: Remapping enabled. UIF status bit is copied to TIM_CNTR bit 31.	R/W
<10>	Reserved	R/W
<9:8>	**CKD[1:0]**: Clock division. This bits indicates the division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock (t_{DTS}) used by the dead-time generators and digital filters (ETR, TIx): 2'b00: $t_{DTS}=t_{CK_INT}$ 2'b01: $t_{DTS}=2*t_{CK_INT}$ 2'b10: $t_{DTS}=4*t_{CK_INT}$ 2'b11: Reserved Note: $t_{DTS}=1/f_{DTS}$, $t_{CK_INT}=1/f_{CK_INT}$.	R/W
<7>	**ARPE**: Auto-reload preload enable 1'b0: ARR is not buffering 1'b1: ARR is buffered	R/W
<6:5>	**CMS[1:0]**: Center-aligned mode selection 2'b00: Edge-aligned mode. The counter counts up or down depending on the direction bit (DIR) . 2'b01: Center-aligned mode 1. The counter counts up and down alternatively. The output compare CCx interrupt flag of channels configured in output (CCxs=00 in TIM_CHMDx) are set when the counter is counting down. 2'b10: Center-aligned mode 2. The counter counts up and down alternatively. The output compare CCx interrupt flag of channels configured in output (CCxs=00 in TIM_CHMDx) are set when the counter is counting up. 2'b11: Center-aligned mode 3. The counter counts up and down alternatively. The output compare CCx interrupt flag of channels configured in output (CCxs=00 in TIM_CHMDx) are set both when the counter is counting down or up. Note: Switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN=1) is not allowed.	R/W
<4>	**DIR**: Direction.	R/W

Bit	Description	Type
	1'b0: Counter used as upcounter 2'b1: Counter used as downcounter Note: This bit is read only when the timer is configured in Center-aligned mode or Encoder mode.	
<3>	**OPM**: One pulse mode 1'b0: Counter is not stopped at update event 1'b1: Counter stops counting at the next update event (clearing the bit CEN)	R/W
<2>	**URS**: Update request source This bit is set and cleared by software to select the UEV event sources. 1'b0: Any of the following events generate an updateinterrupt ot DNA request fi enabled. These events can be: - Counter overflow/underflow - Setting the UG bit - Update generation through the slave mode controller 1'b1: Only counter overflow/underflow generates an update interrupt or DMA request id enabled.	R/W
<1>	**UDIS**: Update disable. This bit is set and cleared by software to enable/disable UEV event generation. 1'b0: UEV enabled. The Update (UEV) event is generated by one the following events: - Counter overflow/underflow - Setting the UG bit - Update generation through the slave mode controller Buffered registers are then loaded with their preload values. 1'b1: UEV disabled. The update event is not generated, shadow registers keep their value (ARR, PSC, CCRx). However the counter and the prescaler are reinitialized if the UG bit is set or if a hardware reset is received from the slave mode controller.	R/W
<0>	**CEN**: Counter enable 1'b0: Counter disabled 1'b1: Counter enabled Note: External clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.	R/W

15.1.2 TIM_CTRL2 (Timer control register 2)

Base address: 0x4000_1000
Address offset: 0x04
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:24>	Reserved.	R/W
<23:20>	**MMS2[3:0]**: Master mode selection 2 (TRGO2) This bits allow the information to be sent to ADC for synchronization (TRGO2) to be selected. The combination is as follows: 4’b0000: Reset - the UG bit from the TIM_EVT register is used as trigger output (TRGO2). If the reset is generated by the trigger input (slave mode controller configured in reset mode), the signal on TRGO2 is delayed compared to the actual reset. 4’b0001: Enable - the Counter Enavle signal CNT_EN is used as trigger output (TRGO2). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The Counter Enable signal is generated by a logic AND between the CEN control bit and the trigger input when configured in Gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO2, except if the Master/Slave mode is selected. (see MSM bit description in TIM_SLVMD register) 4’b0010: Update – the update event is selected as trigger output (TRGO2). For instance, a master timer can then be used as prescaler for a slave timer. 4’b0011: Compare pulse – the trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or compare match occurs (TRGO2). 4’b0100: Compare - OC1REFC signal is used as trigger output (TRGO2). 4’b0101: Compare – OC2REFC signal is used as trigger output (TRGO2). 4’b0110: Compare – OC3REFC signal is used as trigger output (TRGO2). 4’b0111: Compare – OC4REFC signal is used as trigger output (TRGO2). 4’b1000: Compare – OC5REFC signal is used as trigger output (TRGO2). 4’b1001: Compare – OC6REFC signal is used as trigger output (TRGO2). 4’b1010: Compare pulse – OC4REFC rising or falling edges generate pulsed on TRGO2. 4’b1011: Compare pulse – OC6REFC rising or falling edges generate pulsed on TRGO2. 4’b1100: Compare pulse – OC4REFC or OC6REFC rising edges generate pulsed on TRGO2. 4’b1101: Compare pulse – OC4REFC rising or OC6REFC falling edges generate pulsed on TRGO2. 4’b1110: Compare pulse – OC6REFC or OC6REFC rising edges generate pulsed on TRGO2. 4’b1111: Compare pulse – OC5REFC rising or OC6REFC falling edges generate pulsed on TRGO2. Note: the clock of the slave timer or ADC must ne enabled prior to receive events from the master timer, and must not be changed on-the-fly while triggers are received from the master timer.	R/W

Bit	Description	Type
<19>	Reserved	R/W
<18>	**OIS6**: OC6 output Refer OC1	R/W
<17>	Reserved	R/W
<16>	**OIS5**: OC5 output Refer OC1	R/W
<15>	Reserved	R/W
<14>	**OIS4**: OC4 output Refer OC1	R/W
<13>	**OIS3N**: OC3N output Refer OC1N	R/W
<12>	**OIS3**: OC3 output Refer OC1	R/W
<11>	**OIS2N**: OC2N output Refer to OC2N	R/W
<10>	**OIS2**: OC2 output Refer to OC1	R/W
<9>	**OIS1N**: OC1N output 1'b0: When MOE=0, OC1N=0 (after dead-time) 1'b1: When MOE=0, OC1N=1 (after dead-time)	R/W
<8>	**OIS1**: OC1 output 1'b0: When MOE=0, OC1=0 (after dead-time) 1'b1: When MOE=0, OC1=1 (after dead-time)	R/W
<7>	**TI1S**: TI1 selection 1'b0: TIMx_CH1 pin is connected to TI1 input 2'b1: TIMx_CH1, CH2 and CH3 pins are connected to TI1 input (XOR combination)	R/W
<6:4>	**MMS[2:0]**: Master mode selection (TRGO) These bits allow selected information to be sent in master mode to slave timers for synchronization (TRGO). The combination is as follows: 3'b000: Reset- the UG bit from the TIM_EVT register is used as trigger output (TRGO). If the reset is generated by the trigger input (slave mode controller configured in reset mode), the signal on TRGO is delayed compared to the actual reset. 3'b001: Enable - the Counter Enavle signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enabled. The Counter Enable signal is generated by a logic AND between the CEN control bit and the trigger input when configured in Gated mode. When the Counter Enable signal is controlled by the trigger input,	R/W

Bit	Description	Type
	there is a delay on TRGO, except if the Master/Slave mode is selected. (see MSM bit description in TIM_SLVMD register) 3'b010: Update – the update event is selected as trigger output (TRGO). For instance, a master timer can then be used as prescaler for a slave timer. 3'b011: Compare pulse – the trigger output sends a positive pulse when the CC1IF flag is to be set (even if it was already high), as soon as a capture or compare match occurs (TRGO). 3'b100: Compare – OC1REFC signal is used as trigger output (TRGO). 3'b101: Compare – OC2REFC signal is used as trigger output (TRGO). 3'b110: Compare – OC3REFC signal is used as trigger output (TRGO). 3'b111: Compare – OC4REFC signal is used as trigger output (TRGO). Note: the clock of the slave timer or ADC must ne enabled prior to receive events from the master timer, and must not be changed on-the-fly while triggers are received from the master timer.	
<3>	**CCDS**: Capture/compare DMA selection 1'b0: CCx DMA request sent when CCx event occurs 2'b1: CCx DMA requests send when update event occurs	R/W
<2>	**CCUS**: Capture/compare control update selection 1'b0: updated by setting the COMG bit only when capture/compare control bits are preload (CCPC=1) 1'b1: updated by setting the COMG bit or when an rising edge occurs on TRGI when capture/compare control bits are preload (CCPC=1) Note: This bit acts only on channels that have a complementary output.	R/W
<1>	Reserved.	R/W
<0>	**CCPC**: Capture/compare preloaded control 1' b0: CCxE, CCxNE and OCxM bits are not preloaded 1' b1: CCxE, CCxNE and OCxM bits are preloaded, after having been written, they are updated only when a commutation event (COM) occurs (COMG bit set or rising edge detected on TRGI, depending on the CCUS bit).	R/W

15.1.3 TIM_SLVMD (Timer slave mode control register)

Base address: 0x4000_1000
Address offset: 0x08
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:22>	Reserved.	R/W
<19:17>	Reserved	R/W
<15>	**ETP**: External trigger polarity	R/W

Bit	Description	Type
	This bit selected whether ETR or ~ETR is used for trigger operations 1'b0: ETR is non-inverted, active at high level or riding edge 1'b1: ETR is inverted, active at low level or falling edge	
<14>	**ECE**: External clock mode 2 enable This bit enables External clock mode 2. 1'b0: External clock mode 2 disabled 1'b1: External clock mode 2 enabled. The counter is clocked by any active edge on the ETRF signal. Note: Setting the ECE bit has the same effect as selecting external clock mode 1 with TRGI connected to ETRF (SMS =11 and TS =00111). It is possible to simultaneously use external clock mode 2 with the following slave modes: reset mode, fated mode and trigger mode. Nevertheless, TRGI must not be connected to ETRF in this case (TS bits must not be 00111). If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.	R/W
<13:12>	**ETPS[1:0]**: External trigger prescaler External trigger signal ETRF frequency must be at most ¼ of f_{CK_INT} frequency. A prescaler can be enabled to reduced ETRF frequency. It is useful when inputting fast external clocks. 2'b00: Prescaler off 2'b01: divided by 2 2'b10: divided by 4 2'b11: divided by 8	R/W
<11:8>	**ETF[3:0]**: External trigger filter These bits defines the frequency used to sample ETRP signal and the length of the digital filter applied to ETRP. The digital filter is made of an event counter in which N consecutive events are needed to validated a transition on the output: 4'b0000: no filter, $f_{SAMPLING}=f_{DTS}$ 4'b0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 4'b0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 4'b0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 4'b0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 4'b0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 4'b0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 4'b0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 4'b1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 4'b1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 4'b1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 4'b1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 4'b1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 4'b1101: $f_{SAMPLING}=f_{DTS}/32$, N=5	R/W

Bit	Description	Type
	4’b1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 4’b1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	
<7>	**MSM**: Master/Slave mode 1’b0: No action 1’b1:The effect of an event on trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.	R/W
<21:20, 6:4>	**TS[4:0]**: Trigger selection These bits selects the trigger input to be used to synchronize the counter. 5’b00000: Internal trigger 0 (ITR0) 5’b00001: Internal trigger 1 (ITR1) 5’b00010: Internal trigger 2 (ITR2) 5’b00011: Internal trigger 3 (ITR3) 5’b00100: TI1 edge detector (TI1F_ED) 5’b00101: Filtered timer input 1 (TI1FP1) 5’b00110: Filtered timer input 2 (TI2FP2) 5’b00111: External trigger input (ETRF) 5’b01000: Internal trigger 4 (ITR4) See Table TIM internal trigger connection for more details on ITRx meaning for each Timers. Note: These bits must be changed only when they are not used (e.g. when SMS=000) to avoid wrong edge detections at the transition.	R/W
<3>	**OCCS**: OCREF clear source selection This bit is used to select the OCREF clear source. 1’b0: OCREF_CLR_INT is connected to COMP1 or COMP2 output depending on TIM_AUX.OCREF_CLR 1’b1: OCREF_CLR_INT is connected to ETRF	R/W
<16,2:0>	**SMS[3:0]**: Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input. 4’b0000: Slave mode disable – if CEN=1 then the presaler is clocked directly by the internal clock. 4’b0001: Encode mode 1 – Counter counts up/down on TI1FP1 edge depending on TI2FP2 level. 4’b0010: Encode mode 2 – Counter counts up/down on TI2FP2 edge depending on TI1FP1 level. 4’b0011: Encode mode 3 – Counter counts up/down on both TI1FP1 and TI2FP2 edges depending on the level of the other input. 4’b0100: Reset mode – Rising edge of the selected trigger input (TRGI) reinitializes the counter and generates an update of the registers. 4’b0101: Gated mode – The counter clock is enabled when the trigger input	R/W

Bit	Description	Type
	(TRGI) id high. The counter stops (but is not reset) as soon as the trigger becomes low. Both start and stop pf the counter are controlled. 4'b0110: Trigger mode – The counter starts at a rising edge od the trigger input TRGI (but it is not reset). Only the start of counter is controlled/ 4'b0111: External clock mode – Rising edges of the selected trigger (TRGI) clock the counter. 4'b1000: Combined reset and trigger mode – Rising edges if the selected trigger (TRGI) reinitialized the counter, generates an update of the registers and starts the counter. Note: The gated mode must not be used if TI1IF_ED is selected as the trigger input (TS=00100). Indeed, TI1F_ED outputs 1 pulse for each transition on TI1F, whereas the gated mode checks the level of the trigger signal. Note: The clock of the slave peripherals (timer, ADC, …) receiving the TRGO or the TRGO2 signal must be enabled prior to receive events from the master timer, and the clock frequency (prescaler) must not be changed on-the-fly while triggers are received from the master timer.	

Table 39 TIM internal trigger connection

	internal trigger resource					
slave timer	ITR0	ITR1	ITR2	ITR3	ITR4	
Ad_Timer0	Ad_Timer1	GP_Timer0	Gp_Timer1	Gp_Timer2	Gp_Timer3	
Ad_Timer1	Ad_Timer0	GP_Timer0	Gp_Timer1	Gp_Timer2	Gp_Timer3	
Gp_Timer0	Ad_Timer0	Ad_Timer1	Gp_Timer1	Gp_Timer2	Gp_Timer3	
Gp_Timer1	Ad_Timer0	Ad_Timer1	GP_Timer0	Gp_Timer2	Gp_Timer3	
Gp_Timer2	Ad_Timer0	Ad_Timer1	GP_Timer0	Gp_Timer1	Gp_Timer3	
Gp_Timer3	Ad_Timer0	Ad_Timer1	GP_Timer0	Gp_Timer1	Gp_Timer2	
	slave					
master	Ad_Timer0	Ad_Timer1	Gp_Timer0	Gp_Timer1	Gp_Timer2	Gp_Timer3
Ad_Timer0	x	ADV1_ITR0	GP0_ITR0	GP1_ITR0	GP2_ITR0	GP3_ITR0
Ad_Timer1	ADV0_ITR0	x	GP0_ITR1	GP1_ITR1	GP2_ITR1	GP3_ITR1
Gp_Timer0	ADV0_ITR1	ADV1_ITR1	x	GP1_ITR2	GP2_ITR2	GP3_ITR2
Gp_Timer1	ADV0_ITR2	ADV1_ITR2	GP0_ITR2	x	GP2_ITR3	GP3_ITR3
Gp_Timer2	ADV0_ITR3	ADV1_ITR3	GP0_ITR3	GP1_ITR3	x	GP3_ITR4
Gp_Timer3	ADV0_ITR4	ADV1_ITR4	GP0_ITR4	GP1_ITR4	GP2_ITR4	x

15.1.4 TIM_REQEN (Timer DMA/interrupt enable register)

Base address: 0x4000_1000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:15>	Reserved.	R/W
<14>	**TDE**: Trigger DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<13>	**COMDE**: COM DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<12>	**CC4DE**: Capture/compare 4 DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<11>	**CC3DE**: Capture/compare 3 DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<10>	**CC2DE**: Capture/compare 2 DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<9>	**CC1DE**: Capture/compare 1 DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<8>	**UDE**: Update DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<7>	**BIE**: Break interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<6>	**TIE**: Trigger interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<5>	**COMIE**: COM interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<4>	**CC4IE**: Capture/compare 4 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<3>	**CC3IE**: Capture/compare 3 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<2>	**CC2IE**: Capture/compare 2 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W

Bit	Description	Type
<1>	**CC1IE**: Capture/compare 1 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<0>	**UIE**: Update interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W

15.1.5 TIM_STATUS (Timer status register)

Base address: 0x4000_1000
Address offset: 0x10
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:18>	Reserved.	R/W
<17>	**CC6IF**: Compare 6 interrupt flag Refer to CC1IF	R/W
<16>	**CC5IF**: Compare 5 interrupt flag Refer to CC1IF	R/W
<15:14>	Reserved	R/W
<13>	**SBIF**: System break interrupt flag 1'b0: No break event occurred 1'b1: Detect active signal on system break input.	R/W
<12>	**CC4OF**: Capture/compare 4 overcapture flag Refer to CC1OF	R/W
<11>	**CC3OF**: Capture/compare 3 overcapture flag Refer to CC1OF	R/W
<10>	**CC2OF**: Capture/compare 2 overcapture flag Refer to CC1OF	R/W
<9>	**CC1OF**: Capture/compare 1 overcapture flag 1'b0: No overcapture 1'b1: Capture has been detected when CC1IF has occurred.	R/W
<8>	**B2IF**: Break 2 interrupt flag 1'b0: No break event occurred 1'b1: Break event occurred when - Detect active signal on break 2 input when BK2E is set; - B2G is set.	R/W
<7>	**BIF**: Break interrupt flag 1'b0: No break event occurred	R/W

Bit	Description	Type
	1'b1: Break event occurred when - Detect active signal on break input when BKE is set; - Detect active signal on system break input when BKE is set; - BG is set.	
<6>	**TIF**: Trigger interrupt flag 1'b0: No trigger event occurred 1'b1: Trigger occurred.	R/W
<5>	**COMIF**: COM event interrupt flag 1'b0: No COM event occurred 1'b1: COM occurred.	R/W
<4>	**CC4IF**: Capture/compare 4 interrupt flag Refer to CC1IF	R/W
<3>	**CC3IF**: Capture/compare 3 interrupt flag Refer to CC1IF	R/W
<2>	**CC2IF**: Capture/compare 2 interrupt flag Refer to CC1IF	R/W
<1>	**CC1IF**: Capture/compare 1 interrupt flag 1'b0: No compare match or no input capture occurred 1'b1: A compare match or an input capture occurred.	R/W
<0>	**UIF**: Update interrupt flag 1'b0: No update occurred 1'b1: Update event occurred.	R/W

15.1.6 TIM_EVT (Timer event generation register)

Base address: 0x4000_1000
Address offset: 0x14
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:9>	Reserved.	R/W
<8>	**B2G**: Break 2 generation Refer to BG	R/W
<7>	**BG**: Break generation 1'b0: No action 1'b1: A break event is generated.	R/W
<6>	**TG**: Trigger generation 1'b0: No action	R/W

Bit	Description	Type
	1'b1: TIF is set.	
<5>	**COMG**: Capture/compare control update generation 1'b0: No action 1'b1: COM event is generated.	R/W
<4>	**CC4G**: Capture/compare 4 generation Refer to CC1G.	R/W
<3>	**CC3G**: Capture/compare 3 generation Refer to CC1G.	R/W
<2>	**CC2G**: Capture/compare 2 generation Refer to CC1G.	R/W
<1>	**CC1G**: Capture/compare 1 generation 1'b0: No action 1'b1: A capture or compare event is generated on channel.	R/W
<0>	**UG**: Update generation 1'b0: No action 1'b1: Generate a update.	R/W

15.1.7 TIM_CHMD1 (Timer compare mode register 1 - output)

Base address: 0x4000_1000
Address offset: 0x18
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:25>	Reserved.	R/W
<23:17>	Reserved.	R/W
<15>	**OC2CE**: Output compare 2 clear enable Refer to OC1CE	R/W
<24,14:12>	**OC2M[3:0]**: Output compare 2 mode Refer to OC1M[3:0]	R/W
<11>	**OC2PE**: Output compare 2 preload enable Refer to OC1PE	R/W
<10>	**OC2FE**: Output compare 2 fast enable Refer to OC2FE	R/W
<9:8>	**CC2S[1:0]**: Capture/compare 2 selection 2'b00: Channel 2 is output 2'b01: Channel 2 is input, IC2 is mapped on TI2 2'b00: Channel 2 is input, IC2 is mapped on TI1 2'b00: Channel 2 is input, IC2 is mapped on TRC.	R/W
<7>	**OC1CE**: Output compare 1 clear enable	R/W

Bit	Description	Type
	1’b0: No effect 1’b1: OCREF_CLR_INT signal is high to clear OC1REF.	
<16,6:4>	**OC1M[3:0]**: Output compare 1 mode 4’b0000: Frozen (used to generate a timing base) 4’b0001: Set channel 1 to active level on match to CCR1 4’b0010: Set channel 1 to inactive level on match to CCR1 4’b0011: Toggle 4’b0100: Force inactive level 4’b0101: Force active level 4’b0110: PWM mode 1. Channel 1 is active when CNT < CCR1 4’b0111: PWM mode 2. Channel 1 is active when CNT > CCR1 4’b1000: Retriggerrable OPM mode 1. In up-counting mode, channel 1 is active until a trigger event is detected. In down-counting mode, channel 1 is inactive until a trigger event is detected. 4’b1001: Retriggerrable OPM mode 2. In up-counting mode, channel 1 is inactive until a trigger event is detected. In down-counting mode, channel 1 is active until a trigger event is detected. 4’b1010: Reserved. 4’b1011: Reserved. 4’b1100: Combined PWM mode 1. OC1REF has the same behavior as in PWM mode 1. OC1REFC is logic OR between OC1REF and OC2REF. 4’b1101: Combined PWM mode 2. OC1REF has the same behavior as in PWM mode 2. OC1REFC is logic AND between OC1REF and OC2REF. 4’b1110: Asymmetric PWM mode 1. OC1REF has the same behavior as in PWM mode 1. OC1REFC outputs OC1REF in up-counting, OC2REF in down-counting. 4’b1111: Asymmetric PWM mode 2. OC1REF has the same behavior as in PWM mode 2. OC1REFC outputs OC1REF in up-counting, OC2REF in down-counting.	R/W
<3>	**OC1PE**: Output compare 1 preload enable 1’b0: Preload CCR1 disabled 1’b1: Preload CCR1 enabled	R/W
<2>	**OC1FE**: Output compare 1 fast enable 1’b0: CC1 behaves normally depending on counter and CCR1 values when the trigger is ON. 1’b1: An active edge on the trigger input acts like a compare match on CC1 output.	R/W
<1:0>	**CC1S[1:0]**: Capture/compare 1 selection 2’b00:	R/W

15.1.8 TIM_CHMD1 (Timer capture mode register 1 - input)

Base address: 0x4000_1000
Address offset: 0x18
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:12>	**IC2F[3:0]**: Input capture 2 filter Refer to IC1F[3:0]	R/W
<11:10>	**IC2PSC[1:0]**: Input capture 2 prescaler Refer to IC2PSC[1:0]	R/W
<9:8>	**CC2S[1:0]**: Capture/compare 2 selection Refer to CC1S[1:0]	R/W
<7:4>	**IC1F[3:0]**: Input capture 1 filter 4’b0000: No filter 4’b0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 4’b0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 4’b0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 4’b0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 4’b0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 4’b0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 4’b0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 4’b1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 4’b1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 4’b1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 4’b1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 4’b1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 4’b1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 4’b1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 4’b1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	R/W
<3:2>	**IC1PSC[1:0]**: Input capture 1 prescaler 2’b00: no prescaler 2’b01: Capture is done pre 2 events 2’b10: Capture is done pre 4 events 2’b11: Capture is done pre 8 events	R/W
<1:0>	**CC1S[1:0]**: Capture/compare 1 selection 2’b00: Channel 1 is output 2’b01: Channel 1 is input, IC1 is mapped on TI1 2’b10: Channel 1 is input, IC1 is mapped on TI2 2’b11: Channel 1 is input, IC1 is mapped on TRC	R/W

15.1.9 TIM_CHMD2 (Timer compare mode register 2 - output)

Base address: 0x4000_1000
Address offset: 0x1C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:25>	Reserved.	R/W
<23:17>	Reserved.	R/W
<15>	**OC4CE**: Output compare 4 clear enable Refer to OC1CE	R/W
<24,14:12>	**OC4M[3:0]**: Output compare 4 mode Refer to OC1M[3:0]	R/W
<11>	**OC4PE**: Output compare 4 preload enable Refer to OC1PE	R/W
<10>	**OC4FE**: Output compare 4 fast enable Refer to OC1FE	R/W
<9:8>	**CC4S[1:0]**: Capture/compare 4 selection Refer to CC1S[1:0]	R/W
<7>	**OC3CE**: Output compare 3 clear enable Refer to OC1CE	R/W
<16,6:4>	**OC3M[3:0]**: Output compare 3 mode. Refer to OC1M[3:0]	R/W
<3>	**OC3PE**: Output compare 3 preload enable Refer to OC1PE	R/W
<2>	**OC3FE**: Output compare 3 fast enable Refer to OC1FE	R/W
<1:0>	**CC3S[1:0]**: Capture/compare 3 selection Refer to CC1S[1:0]	R/W

15.1.10 TIM_CHMD2 (Timer capture mode register 2- input)

Base address: 0x4000_1000
Address offset: 0x1C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:12>	**IC4F[3:0]**: Input capture 4 filter Refer to IC1F[3:0]	R/W

Bit	Description	Type
<11:10>	**IC4PSC[1:0]**: Input capture 4 prescaler Refer to OC1PSC[1:0]	R/W
<9:8>	**CC4S[1:0]**: Capture/compare 4 selection Refer to CC1S[1:0]	R/W
<7:4>	**IC3F[3:0]**: Input capture 3 filter Refer to IC1F[3:0]	R/W
<3:2>	**IC3PSC[1:0]**: Input capture 3 prescaler Refer to IC1PSC[1:0]	R/W
<1:0>	**CC3S[1:0]**: Capture/compare 3 selection Refer to CC1S[1:0]	R/W

15.1.11 TIM_CHEP (Timer capture/compare enable register)

Base address: 0x4000_1000
Address offset: 0x20
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:22>	Reserved.	R/W
<21>	**CC6P**: Capture/compare 6 output polarity Refer to CC1P	R/W
<20>	**CC6E**: Capture/compare 6 output enable Refer to CC1E	R/W
<19:18>	Reserved	R/W
<17>	**CC5P**: Capture/compare 5 output polarity Refer to CC1P	R/W
<16>	**CC5E**: Capture/compare 5 output enable Refer to CC1E	R/W
<15>	**CC4NP**: Capture/compare 4 complementary output polarity Refer to CC1NP	R/W
<14>	Reserved	R/W
<13>	**CC4P**: Capture/compare 4 output polarity Refer to CC1P	R/W
<12>	**CC4E**: Capture/compare 4 output enable Refer to CC1E	R/W
<11>	**CC3NP**: Capture/compare 3 complementary output polarity Refer to CC1NP	R/W
<10>	**CC3NE**: Capture/compare 3 complementary output enable Refer to CC1NE	R/W
<9>	**CC3P**: Capture/compare 3 output polarity	R/W

Bit	Description	Type
	Refer to CC1P	
<8>	**CC3E**: Capture/compare 3 output enable Refer to CC1E	R/W
<7>	**CC2NP**: Capture/compare 2 complementary output polarity Refer to CC1NP	R/W
<6>	**CC2NE**: Capture/compare 2 complementary output enable Refer to CC1NE	R/W
<5>	**CC2P**: Capture/compare 2 output polarity Refer to CC1P	R/W
<4>	**CC2E**: Capture/compare 2 output enable Refer to CC1E	R/W
<3>	**CC1NP**: Capture/compare 1 complementary output polarity 1'b0: OC1N active high (output mode) 1'b1: OC1N active low (output mode) In input mode, this bit is conjunction with CC1P	R/W
<2>	**CC1NE**: Capture/compare 1 complementary output enable 1'b0: Off. OC1N is not active. 1'b1: On. OC1N is active.	R/W
<1>	**CC1P**: Capture/compare 1 output polarity 1'b0: OC1 active high (output mode)/Edge sensitivity selection (input mode) 1'b1: OC1 active low (output mode) /Edge sensitivity selection (input mode) In input mode, CC1NP=0, CC1P=0: non-inverted/rising edge. CC1NP=0, CC1P=1: Inverted/falling edge. CC1NP=1, CC1P=1: non-inverted/both edges. CC1NP=1, CC1P=0: reserved.	R/W
<0>	**CC1E**: Capture/compare 1 output enable 1'b0: Capture mode disabled/OC1 is not active 1'b1: Capture mode enabled/OC1 is active.	R/W

15.1.12 TIM_CNTR (Timer counter)

Base address: 0x4000_1000
Address offset: 0x24
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31>	**UIFCP**: Update flag copy	R/W
<30:16>	Reserved	R/W
<15:0>	**CNT[15:0]**: Counter value	R/W

15.1.13 TIM_PSCR (Timer prescaler)

Base address: 0x4000_1000
Address offset: 0x28
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**PSE[15:0]**: Prescaler value	R/W

15.1.14 TIM_PRD (Timer auto-load register)

Base address: 0x4000_1800
Address offset: 0x2C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**ARR[15:0]**: Auto-load value	R/W

15.1.15 TIM_REP (Timer repetition counter register)

Base address: 0x4000_1000
Address offset: 0x30
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**REP[15:0]**: Repetition counter value	R/W

15.1.16 TIM_CHR1 (Timer capture/compare register 1)

Base address: 0x4000_1000
Address offset: 0x34
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type

<31:16>	Reserved.	R/W
<15:0>	**CCR1[15:0]**: Capture/compare value for channel 1	R/W

15.1.17 TIM_CHR2 (Timer capture/compare register 2)

Base address: 0x4000_1000
Address offset: 0x38
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR2[15:0]**: Capture/compare value for channel 2	R/W

15.1.18 TIM_CHR3 (Timer capture/compare register 3)

Base address: 0x4000_1000
Address offset: 0x3C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR3[15:0]**: Capture/compare value for channel 3	R/W

15.1.19 TIM_CHR4 (Timer capture/compare register 4)

Base address: 0x4000_1000
Address offset: 0x40
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR4[15:0]**: Capture/compare value for channel 4	R/W

15.1.20 TIM_BKDT (Timer break and dead-time register)

Base address: 0x4000_1000
Address offset: 0x44
Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:30>	Reserved.	R/W
<29>	**BK2BID**: Break 2 bidirectional Refer to BKBID	R/W
<28>	**BKBID**: Break bidirectional 1'b0: Input mode 1'b1: Bidirectional mode	R/W
<27>	**BK2DSRM**: Break 2 disarm Refer to Break disarm	R/W
<26>	**BKDSRM**: Break disarm 1'b0: Break input is armed 1'b1: Break input is disarmed	R/W
<25>	**BK2P**: Break 2 polarity Refer to BKP	R/W
<24>	**BK2E**: Break 2 enable Refer to BKE	R/W
<23:20>	**BK2F[3:0]**: Break 2 filter Refer to BKF[3:0]	R/W
<19:16>	**BKF[3:0]**: Break filter 4'b0000: no filter, BRK acts as asynchronously 4'b0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 4'b0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 4'b0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 4'b0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 4'b0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 4'b0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 4'b0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 4'b1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 4'b1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 4'b1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 4'b1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 4'b1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 4'b1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 4'b1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 4'b1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	R/W
<15>	**MOE**: Main output enable 1'b0: OC and OCN output are disabled 1'b1: OC and OCN output are enabled	R/W
<14>	**AOE**: Automatic output enable 1'b0: Bit[16] can be set only by software 1'b1: Bit[16] can be ser by software or automatically at the next update event.	R/W

Bit	Description	Type
<13>	**BKP**: Break polarity 1'b0: Active low 1'b1: Active high	R/W
<12>	**BKE**: Break enable 1'b0: Disabled 1'b1: Enabled	R/W
<11>	**OSSR**: Off-state selection for run mode (Bit[16] =1) 1'b0: OC/OCN output are disabled when inactive 1'b1: OC/OCN output are enabled when inactive	R/W
<10>	**OSSI**: Off-state selection for idle mode (Bit[16] = 0) 1'b0: OC/OCN output are disabled when inactive 1'b1: OC/OCN output are enabled when inactive	R/W
<9:8>	**LOCK[1:0]**: Lock configuration 2'b00: lock off 2'b01: lock level 1, BKDT.bit[14], BKDT.bit[11], BKDT.bit[10], BKDT.bit[7:0],CTRL2.bit[18], CTRL2.bit[16], CTRL2.bit[14], CTRL2.bit[13], CTRL2.bit[12], CTRL2.bit[11], CTRL2.bit[10], CTRL2.bit[9], CTRL2.bit[8] are write protected 2'b10: lock level 2. Level 1 + CC polarity bits (CHEP) are write protected 2'b11: lock level 3. Level 2 + CC control bits (CHMDx) are write protected	R/W
<7:0>	**DT[7:0]**: Dead-time generator setup 8'b0xxxxxxx: bit[7:0] * t_{DTS} 8'b10xxxxxx: (64+bit[5:0])*2 * t_{DTS} 8'b110xxxxx: (32+bit[4:0])*8 * t_{DTS} 8'b111xxxxx: (32+bit[4:0])*16** t_{DTS}	R/W

15.1.21 TIM_DMAC (Timer DMA control register)

Base address: 0x4000_1000
Address offset: 0x48
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:17>	Reserved.	R/W
<16>	**DMA_RSTN**: DMA reset	R/W
<15:13>	Reserved	R/W
<12:8>	**DBL[4:0]**: DMA burst length	R/W
<7:5>	Reserved	R/W

<4:0>	**DBA[4:0]**: DMA base address	R/W

15.1.22 TIM_DMAI (Timer DMA information register)

Base address: 0x4000_1000
Address offset: 0x4C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**DMAB[31:0]**: DMA register for burst accessed.	R/W

15.1.23 TIM_AUX (Timer auxiliary register)

Base address: 0x4000_1000
Address offset: 0x50
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:2>	Reserved.	R/W
<1:0>	**OCREF_CLR[1:0]**: OCREF clear input source selection 2'b00: COMP1 output 2'b01: COMP2 output 2'b10: COMP3 output 2'b11: Reserved	R/W

15.1.24 TIM_CHMD3 (Timer compare mode register 3 - output)

Base address: 0x4000_1000
Address offset: 0x54
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:25>	Reserved.	R/W
<23:17>	Reserved	R/W
<15>	**OC6CE**: Output compare 6 clear enable Refer to OC1CE	R/W
<24,14:12>	**OC6M[3:0]**: Output compare 6 mode Refer to OC1M[3:0]	R/W
<11>	**OC6PE**: Output compare 6 preload enable Refer to OC1PE	R/W

<10>	**OC6FE**: Output compare 6 fast enable Refer to OC1FE	R/W
<9:8>	Reserved	R/W
<7>	**OC5CE**: Output compare 5 clear enable Refer to OC1CE	R/W
<16,6:4>	**OC5M[3:0]**: Output compare 5 mode Refer to OC1M[3:0]	R/W
<3>	**OC5PE**: Output compare 5 preload enable Refer to OC1PE	R/W
<2>	**OC5FE**: Output compare 5 fast enable Refer to OC1FE	R/W
<1:0>	Reserved	R/W

15.1.25 TIM_CHR5 (Timer capture/compare register 5)

Base address: 0x4000_1000
Address offset: 0x58
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31>	**GC5C3**: Group channel 5 and channel 3 1'b0: No effect of OC5REF on OC3REFC 1'b1: OC3REFC is the logical AND of OC3REFC and OC5REF	R/W
<30>	**GC5C2**: Group channel 5 and channel 2 1'b0: No effect of OC5REF on OC2REFC 1'b1: OC2REFC is the logical AND of OC2REFC and OC5REF	R/W
<29>	**GC5C1**: Group channel 5 and channel 1 1'b0: No effect of OC5REF on OC1REFC 1'b1: OC1REFC is the logical AND of OC1REFC and OC5REF	R/W
<28:16>	Reserved	R/W
<15:0>	**CCR5[15:0]**: Capture/Compare 5 value	R/W

15.1.26 TIM_CHR6 (Timer capture/compare register 6)

Base address: 0x4000_1000
Address offset: 0x5C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W

<15:0>	**CCR6[15:0]**: Capture/compare 6 value	R/W

15.1.27 TIM_BRK (Timer break control register)

Base address: 0x4000_1000
Address offset: 0x60
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:18>	Reserved.	R/W
<17:14>	**ETRSEL[3:0]**: ETR source selection 4'b0000: ETR legacy mode 4'b0001: COMP1 output 4'b0010: COMP2 output 4'b0011: 4'b0100: 4'b0101: 4'b0110: COMP3 output	R/W
<13>	Reserved	R/W
<12>	**BKCMP3P**: BRK COMP3 input polarity 1'b0: COMP3 input polarity is not inverted 1'b1: COMP3 input polarity in inverted	R/W
<11>	**BKCMP2P**: BRK COMP2 input polarity 1'b0: COMP2 input polarity is not inverted 1'b1: COMP2 input polarity in inverted	R/W
<10>	**BKCMP1P**: BRK COMP1 input polarity 1'b0: COMP1 input polarity is not inverted 1'b1: COMP1 input polarity in inverted	R/W
<9>	**BKINP**: BRK BKIN input polarity 1'b0: BKIN input polarity is not inverted 1'b1: BKIN input polarity in inverted	R/W
<8:4>	Reserved	R/W
<3>	**BKCMP3E**: BRK COMP3 enable 1'b0: COMP3 input disable 1'b1: COMP3 input enable	R/W
<2>	**BKCMP2E**: BRK COMP2 enable 1'b0: COMP2 input disable 1'b1: COMP2 input enable	R/W
<1>	**BKCMP1E**: BRK COMP1 enable 1'b0: COMP1 input disable 1'b1: COMP1 input enable	R/W

Bit	Description	Type
<0>	**BKINE**: BRK BKIN enable 1'b0: BKIN input disable 1'b1: BKIN input enable	R/W

15.1.28 TIM_BRK2 (Timer break control register 2)

Base address: 0x4000_1000
Address offset: 0x64
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:13>	Reserved.	R/W
<12>	**BK2CMP3P**: BRK2 COMP3 input polarity 1'b0: COMP3 input polarity is not inverted 1'b1: COMP3 input polarity in inverted	R/W
<11>	**BK2CMP2P**: BRK2 COMP2 input polarity 1'b0: COMP2 input polarity is not inverted 1'b1: COMP2 input polarity in inverted	R/W
<10>	**BK2CMP1P**: BRK2 COMP1 input polarity 1'b0: COMP1 input polarity is not inverted 1'b1: COMP1 input polarity in inverted	R/W
<9>	**BK2INP**: BRK2 BKIN2 input polarity 1'b0: BKIN2 input polarity is not inverted 1'b1: BKIN2 input polarity in inverted	R/W
<8:4>	Reserved	R/W
<3>	**BK2CMP3E**: BRK2 COMP3 enable 1'b0: COMP3 input disable 1'b1: COMP3 input enable	R/W
<2>	**BK2CMP2E**: BRK2 COMP2 enable 1'b0: COMP2 input disable 1'b1: COMP2 input enable	R/W
<1>	**BK2CMP1E**: BRK2 COMP1 enable 1'b0: COMP1 input disable 1'b1: COMP1 input enable	R/W
<0>	**BK2INE**: BRK2 BKIN2 enable 1'b0: BKIN2 input disable 1'b1: BKIN2 input enable	R/W

15.1.29 TIM_ISCLT (Timer input selection register)

Base address: 0x4000_1000
Address offset: 0x68
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:28>	Reserved.	R/W
<27:24>	**TI4SEL[3:0]**: Select TI4 input 4'b0000: TIM_CH4 input Others: Reserved	R/W
<23:20>	Reserved	R/W
<19:16>	**TI3SEL[3:0]**: Select TI3 input 4'b0000: TIM_CH3 input 4'b0001: COMP3 output Others: Reserved	R/W
<15:12>	Reserved	R/W
<11:8>	**TI2SEL[3:0]**: Select TI2 input 4'b0000: TIM_CH2 input 4'b0001: COMP2 output Others: Reserved	R/W
<7:4>	Reserved	R/W
<3:0>	**TI1SEL[3:0]**: Select TI1 input 4'b0000: TIM_CH1 input 4'b0001: COMP1 output Others: Reserved	R/W

15.1.30 TIM_DTR1 (Timer dead-time register 1)

Base address: 0x4000_1000
Address offset: 0x6C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:24>	**DTG2N[7:0]**: Dead-time generator setup, the same as DTG1P.	R/W
<23:16>	**DTG2P[7:0]**: Dead-time generator setup, the same as DTG1P.	R/W
<15:8>	**DTG1N[7:0]**: Dead-time generator setup, the same as DTG1P.	R/W
<7:0>	**DTG1P[7:0]**: Dead-time generator setup 8'b0xxxxxxx: bit[7:0] * t_{DTS} 8'b10xxxxxx: (64+bit[5:0])*2 * t_{DTS} 8'b110xxxxx: (32+bit[4:0])*8 * t_{DTS}	R/W

	8’b111xxxxx: (32+bit[4:0])*16** t_{DTS}	

15.1.31 TIM_DTR2 (Timer dead-time register 2)

Base address: 0x4000_1000
Address offset: 0x70
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved	R/W
<15:8>	**DTG3N[7:0]**: Dead-time generator setup, the same as DTG1P.	R/W
<7:0>	**DTG3P[7:0]**: Dead-time generator setup, the same as DTG1P.	R/W

15.1.32 TIMA register map

The base address of the TIMA registers is 0x4000_1000.
The following table gives the TIMA register map and reset values.

Table 40 TIMA register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIM_CTRL1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIFREMAP	Res	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value	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
0x04	TIM_CTRL2	Res	Res	Res	Res	Res	Res	Res	Res	MMS2[3:0]				Res	OIS6	Res	OIS5	Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res	CCPC
	Reset value	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
0x08	TIM_SLVMD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TS[4:3]		Res	Res	Res	SMS[3]	ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
	Reset value	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
0x0C	TIM_REQEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value	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
0x10	TIM_STATUS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6IF	CC5IF	Res	Res	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value	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
0x14	TIM_EVT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	Reset value	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
0x18	TIM_CHMD1 (Output)	Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	Reset value	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
	TIM_CHMD1 (Input)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
	Reset value	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
0x1C	TIM_CHMD2 (Output)	Res	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
	Reset value	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
	TIM_CHMD2 (Input)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
	Reset value	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
0x20	TIME_CHEP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6P	CC6E	Res	Res	CC5P	CC5E	CC4NP	Res	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value	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
0x24	TIM_CNTR	UIFCP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	Reset value	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
0x28	TIM_PSCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	Reset value	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
0x2C	TIM_PRD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	TIM_REP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REP[15:0]															
	Reset value	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
0x34	TIM_CHR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR1[15:0]															
	Reset value	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
0x38	TIM_CHR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR2[15:0]															
	Reset value	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
0x3C	TIM_CHR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR3[15:0]															
	Reset value	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
0x40	TIM_CHR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR4[15:0]															
	Reset value	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
0x44	TIM_BKDT	Res	Res	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DT[7:0]							
	Reset value	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
0x48	TIM_DMAC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMA_RSTN	Res	Res	Res	DBL[4:0]					Res	Res	Res	DBA[4:0]				
	Reset value	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
0x4C	TIM_DMAI	DMAB[31:0]																															
	Reset value	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
0x50	TIM_AUX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OCREF_CLR[1:0]	
	Reset value	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
0x54	TIM_CHMD3	Res	Res	Res	Res	Res	Res	Res	OC6M[3]	Res	Res	Res	Res	Res	Res	Res	OC5M[3]	OC6CE	OC6M[2:0]			OC6PE	OC6FE	Res	Res	OC5CE	OC5M[2:0]			OC5PE	OC5FE	Res	Res
	Reset value	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
0x58	TIM_CHR5	GC5C3	GC5C2	GC5C1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR5[15:0]															
	Reset value	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
0x5C	TIM_CHR6	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR6[15:0]															
	Reset value	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
0x60	TIM_BRK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ETRSEL[3:0]				Res	BKCMP3P	BKCMP2P	BKCMP1P	BKINP	Res	Res	Res	Res	Res	BKCMP3E	BKCMP2E	BKCMP1E	BKINE
	Reset value	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	1
0x64	TIM_BRK2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BK2CMP3P	BK2CMP2P	BK2CMP1P	BK2INP	Res	Res	Res	Res	Res	BK2CMP3E	BK2CMP2E	BK2CMP1E	BK2INE
	Reset value	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	1
0x68	TIM_ISLCT	Res	Res	Res	Res	TI4SEL[3:0]				Res	Res	Res	Res	TI3SEL[3:0]				Res	Res	Res	Res	TI2SEL[3:0]				Res	Res	Res	Res	TI1SEL[3:0]			
	Reset value	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
0x6C	TIM_DTR1	DTG2N[7:0]								DTG2P[7:0]								DTG1N[7:0]								DTG1P[7:0]							
	Reset value	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
0x70	TIM_DTR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DTG3N[7:0]							DTG3P[7:0]							

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

16 General-purpose timer (TIMG)

16.1 Introduction

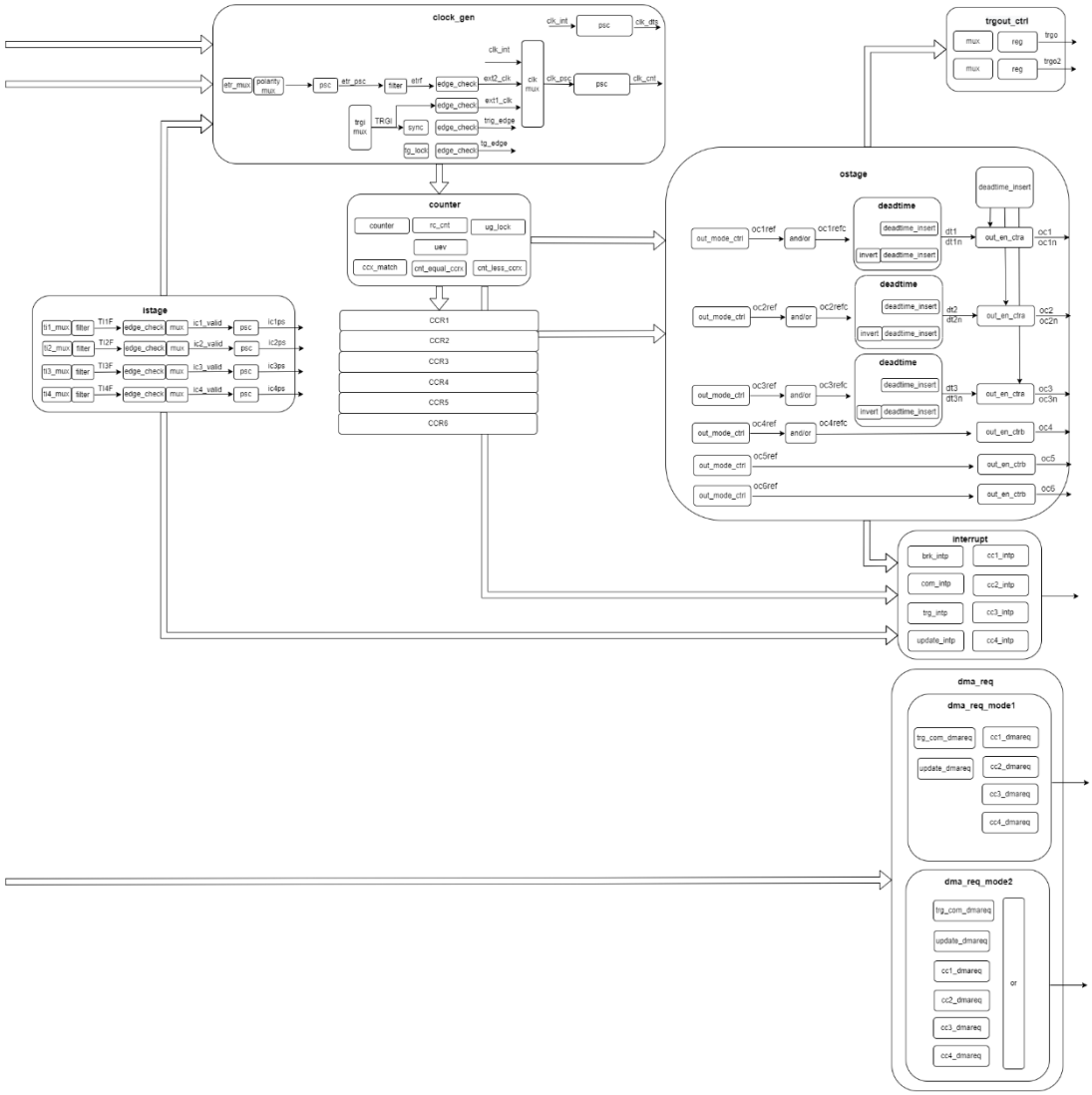


Figure 33 General-purpose timer block diagram

16.2 TIMG main features

- 16-bit up, down, up and down auto-load counter
- 16-bit programming prescaler
- Input capture
- Output compare
- PWM generation
- One-pulse mode output
- Repetition counter
- Interrupts generator
- Configure by DMA

16.3 TIMG signal descriptions

The general timer signals are shown in the following table.

Table 41 TIMG signal descriptions

Signal	Description	I/O
ETR	External trigger input	I
CH1_IN	Channel 1 input	I
CH2_IN	Channel 2 input	I
CH3_IN	Channel 3 input	I
CH4_IN	Channel 4 input	I
CH1_OUT_P	Channel 1 positive output	O
CH1_OUT_N	Channel 1 negitive output	O
CH2_OUT_P	Channel 2 positive output	O
CH2_OUT_N	Channel 2 negitive output	O
CH3_OUT_P	Channel 3 positive output	O
CH3_OUT_N	Channel 3 negitive output	O
CH4_OUT	Channel 4 output	O

16.4 TIMG functional description

Similar to TIMA

16.5 TIMG registers

16.5.1 TIM_CTRL1

Base address: 0x4000_1800
Address offset: 0x00
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:12>	Reserved.	R/W
<11>	**UIFREMAP**: UIF status bit remapping (copy to TIMx_CNT bit[31]) 1'b0: No remapping 1'b1: Remapping enabled	R/W
<10>	Reserved	R/W
<9:8>	**CKD[1:0]**: Clock division 2'b00: $t_{DTS}=t_{CK_INT}$ 2'b01: $t_{DTS}=2*t_{CK_INT}$ 2'b10: $t_{DTS}=4*t_{CK_INT}$ 2'b11: Reserved	R/W
<7>	**ARPE**: Auto-reload preload enable 1'b0: ARR is not buffering 1'b1: ARR is buffered	R/W
<6:5>	**CMS[1:0]**: Center-aligned mode 2'b00: Edge-aligned mode. 2'b01: Center-aligned mode 1. CCx interrupt flag are set in counting down 2'b10: Center-aligned mode 2. CCx interrupt flag are set in counting up 2'b11: Center-aligned mode 3. CCx interrupt flag are set in counting down or up	R/W
<4>	**DIR**: Direction 1'b0: Counter used as upcounter 2'b1: Counter used as downcounter	R/W
<3>	**OPM**: One pulse mode 1'b0: Counter is not stopped at update event 1'b1: Counter stops counting at the next update event	R/W
<2>	**URS**: Update request source 1'b0: generate update by - Counter overflow/underflow - Setting the UG bit - Update generation through the slave mode controller 1'b1: generate update by counter overflow/underflow	R/W

Bit	Description	Type
<1>	**UDIS**: Update disable 1'b0: UEV enabled 1'b1: UEV disabled	R/W
<0>	**CEN**: Counter enable 1'b0: Counter disabled 1'b1: Counter enabled	R/W

16.5.2 TIM_CTRL2

Base address: 0x4000_1800
Address offset: 0x04
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:24>	Reserved.	R/W
<23:20>	**MMS2[3:0]**: Master mode selection 2 (TRGO2) 4'b0000: Reset 4'b0001: Enable 4'b0010: Update 4'b0011: Compare pulse 4'b0100: Compare - OC1REFC 4'b0101: Compare – OC2REFC 4'b0110: Compare – OC3REFC 4'b0111: Compare – OC4REFC 4'b1000: Compare – OC5REFC 4'b1001: Compare – OC6REFC 4'b1010: Compare pulse – OC4REFC rising or falling edge 4'b1011: Compare pulse – OC6REFC rising or falling edge 4'b1100: Compare pulse – OC4REFC or OC6REFC rising edge 4'b1101: Compare pulse – OC4REFC rising or OC6REFC falling edge 4'b1110: Compare pulse – OC6REFC or OC6REFC rising edge 4'b1111: Compare pulse – OC5REFC rising or OC6REFC falling edge	R/W
<19>	Reserved	R/W
<18>	**OIS6**: OC6 output Refer OC1	R/W
<17>	Reserved	R/W
<16>	**OIS5**: OC5 output Refer OC1	R/W
<15>	Reserved	R/W

Bit	Description	Type
<14>	**OIS4**: OC4 output Refer OC1	R/W
<13>	**OIS3N**: OC3N output Refer OC1N	R/W
<12>	**OIS3**: OC3 output Refer OC1	R/W
<11>	**OIS2N**: OC2N output Refer to OC2N	R/W
<10>	**OIS2**: OC2 output Refer to OC1	R/W
<9>	**OIS1N**: OC1N output 1'b0: When MOE=0, OC1N=0 (after dead-time) 1'b1: When MOE=0, OC1N=1 (after dead-time)	R/W
<8>	**OIS1**: OC1 output 1'b0: When MOE=0, OC1=0 (after dead-time) 1'b1: When MOE=0, OC1=1 (after dead-time)	R/W
<7>	**TI1S**: TI1 selection 1'b0: TIMx_CH1 pin is connected to TI1 input 2'b1: TIMx_CH1, CH2 and CH3 pins are connected to TI1 input (XOR combination)	R/W
<6:4>	**MMS[2:0]**: Master mode selection (TRGO) 3'b000: Reset 3'b001: Enable 3'b010: Update 3'b011: Compare pulse 3'b100: Compare – OC1REFC 3'b101: Compare – OC2REFC 3'b110: Compare – OC3REFC 3'b111: Compare – OC4REFC	R/W
<3>	**CCDS**: Capture/compare DMA selection 1'b0: CCx DMA request sent when CCx event occurs 2'b1: CCx DMA requests send when update event occurs	R/W
<2>	**CCUS**: Capture/compare control update selection 1'b0: updated by setting the COMG bit only 1'b1: updated by setting the COMG bit or an rising edge occurs on TRGI	R/W
<1>	Reserved.	R/W
<0>	**CCPC**: Capture/compare preloaded control 1' b0: CCxE, CCxNE and OCxM bits are not preloaded 1' b1: CCxE, CCxNE and OCxM bits are preloaded	R/W

16.5.3 TIM_SLVMD

Base address: 0x4000_1800
Address offset: 0x08
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15>	**ETP**: External trigger polarity 1'b0: ETR is non-inverted, active at high level or riding edge 1'b1: ETR is inverted, active at low level or falling edge	R/W
<14>	**ECE**: External clock mode 2 enable 1'b0: Disable 1'b1: Enable	R/W
<13:12>	**ETPS[1:0]**: External trigger prescaler 2'b00: Prescaler off 2'b01: divider is 2 2'b10: divider is 4 2'b11: divider is 8	R/W
<11:8>	**ETF[3:0]**: External trigger filter 4'b0000: no filter, $f_{SAMPLING}=f_{DTS}$ 4'b0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 4'b0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 4'b0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 4'b0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 4'b0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 4'b0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 4'b0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 4'b1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 4'b1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 4'b1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 4'b1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 4'b1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 4'b1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 4'b1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 4'b1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	R/W
<7>	**MSM**: Master/Slave mode 1'b0: No action 1'b1:Synchronization between TRGI and TRGO.	R/W
<21:20, 6:4>	**TS[4:0]**: Trigger selection 5'b00000: Internal trigger 0 5'b00001: Internal trigger 1	R/W

Bit	Description	Type
	5’b00010: Internal trigger 2 5’b00011: Internal trigger 3 5’b00100: TI1 edge detector 5’b00101: Filtered timer input 1 5’b00110: Filtered timer input 2 5’b00111: External trigger input	
<3>	**OCCS**: OCREF clear source selection 1’b0: COMP1 or COMP2 output 1’b1: ETRF	R/W
<16,2:0>	**SMS[3:0]**: Slave mode selection 4’b0000: Slave mode disable 4’b0001: Encode mode 1 4’b0010: Encode mode 2 4’b0011: Encode mode 3 4’b0100: Reset mode 4’b0101: Gated mode 4’b0110: Trigger mode 4’b0111: External clock mode 4’b1000: Combined reset and trigger mode	R/W

16.5.4 TIM_REQEN

Base address: 0x4000_1800
Address offset: 0x0C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:15>	Reserved.	R/W
<14>	**TDE**: Trigger DMA request enable 1’b0: Request disabled 1’b1: Request enabled	R/W
<13>	**COMDE**: COM DMA request enable 1’b0: Request disabled 1’b1: Request enabled	R/W
<12>	**CC4DE**: Capture/compare 4 DMA request enable 1’b0: Request disabled 1’b1: Request enabled	R/W
<11>	**CC3DE**: Capture/compare 3 DMA request enable 1’b0: Request disabled 1’b1: Request enabled	R/W

Bit	Description	Type
<10>	**CC2DE**: Capture/compare 2 DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<9>	**CC1DE**: Capture/compare 1 DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<8>	**UDE**: Update DMA request enable 1'b0: Request disabled 1'b1: Request enabled	R/W
<7>	Reserved	R/W
<6>	**TIE**: Trigger interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<5>	**COMIE**: COM interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<4>	**CC4IE**: Capture/compare 4 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<3>	**CC3IE**: Capture/compare 3 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<2>	**CC2IE**: Capture/compare 2 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<1>	**CC1IE**: Capture/compare 1 interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W
<0>	**UIE**: Update interrupt enable 1'b0: Interrupt disabled 1'b1: Interrupt enabled	R/W

16.5.5 TIM_STATUS

Base address: 0x4000_1800
Address offset: 0x10
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:18>	Reserved.	R/W

Bit	Description	Type
<17>	**CC6IF**: Compare 6 interrupt flag Refer to CC1IF	R/W
<16>	**CC5IF**: Compare 5 interrupt flag Refer to CC1IF	R/W
<15:13>	Reserved	R/W
<12>	**CC4OF**: Capture/compare 4 overcapture flag Refer to CC1OF	R/W
<11>	**CC3OF**: Capture/compare 3 overcapture flag Refer to CC1OF	R/W
<10>	**CC2OF**: Capture/compare 2 overcapture flag Refer to CC1OF	R/W
<9>	**CC1OF**: Capture/compare 1 overcapture flag 1'b0: No overcapture 1'b1: Capture has been detected when CC1IF has occurred.	R/W
<8:7>	Reserved	R/W
<6>	**TIF**: Trigger interrupt flag 1'b0: No trigger event occurred 1'b1: Trigger occurred.	R/W
<5>	**COMIF**: COM event interrupt flag 1'b0: No COM event occurred 1'b1: COM occurred.	R/W
<4>	**CC4IF**: Capture/compare 4 interrupt flag Refer to CC1IF	R/W
<3>	**CC3IF**: Capture/compare 3 interrupt flag Refer to CC1IF	R/W
<2>	**CC2IF**: Capture/compare 2 interrupt flag Refer to CC1IF	R/W
<1>	**CC1IF**: Capture/compare 1 interrupt flag 1'b0: No compare match or no input capture occurred 1'b1: A compare match or an input capture occurred.	R/W
<0>	**UIF**: Update interrupt flag 1'b0: No update occurred 1'b1: Update event occurred.	R/W

16.5.6 TIM_EVT

Base address: 0x4000_1800
Address offset: 0x14
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:7>	Reserved.	R/W
<6>	**TG**: Trigger generation 1'b0: No action 1'b1: TIF is set.	R/W
<5>	**COMG**: Capture/compare control update generation 1'b0: No action 1'b1: COM event is generated.	R/W
<4>	**CC4G**: Capture/compare 4 generation Refer to CC1G.	R/W
<3>	**CC3G**: Capture/compare 3 generation Refer to CC1G.	R/W
<2>	**CC2G**: Capture/compare 2 generation Refer to CC1G.	R/W
<1>	**CC1G**: Capture/compare 1 generation 1'b0: No action 1'b1: A capture or compare event is generated on channel.	R/W
<0>	**UG**: Update generation 1'b0: No action 1'b1: Generate a update.	R/W

16.5.7 TIM_CHMD1 (Output)

Base address: 0x4000_1800
Address offset: 0x18
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:25>	Reserved.	R/W
<23:17>	Reserved.	R/W
<15>	**OC2CE**: Output compare 2 clear enable Refer to OC1CE	R/W
<24,14:12>	**OC2M[3:0]**: Output compare 2 mode Refer to OC1M[3:0]	R/W
<11>	**OC2PE**: Output compare 2 preload enable Refer to OC1PE	R/W
<10>	**OC2FE**: Output compare 2 fast enable Refer to OC2FE	R/W
<9:8>	**CC2S[1:0]**: Capture/compare 2 selection 2'b00: Channel 2 is output	R/W

Bit	Description	Type
	2’b01: Channel 2 is input, IC2 is mapped on TI2 2’b00: Channel 2 is input, IC2 is mapped on TI1 2’b00: Channel 2 is input, IC2 is mapped on TRC.	
<7>	**OC1CE**: Output compare 1 clear enable 1’b0: No effect 1’b1: OCREF_CLR_INT signal is high to clear OC1REF.	R/W
<16,6:4>	**OC1M[3:0]**: Output compare 1 mode 4’b0000: Frozen (used to generate a timing base) 4’b0001: Set channel 1 to active level on match to CCR1 4’b0010: Set channel 1 to inactive level on match to CCR1 4’b0011: Toggle 4’b0100: Force inactive level 4’b0101: Force active level 4’b0110: PWM mode 1. Channel 1 is active when CNT < CCR1 4’b0111: PWM mode 2. Channel 1 is active when CNT > CCR1 4’b1000: Retriggerrable OPM mode 1. In up-counting mode, channel 1 is active until a trigger event is detected. In down-counting mode, channel 1 is inactive until a trigger event is detected. 4’b1001: Retriggerrable OPM mode 2. In up-counting mode, channel 1 is inactive until a trigger event is detected. In down-counting mode, channel 1 is active until a trigger event is detected. 4’b1010: Reserved. 4’b1011: Reserved. 4’b1100: Combined PWM mode 1. OC1REF has the same behavior as in PWM mode 1. OC1REFC is logic OR between OC1REF and OC2REF. 4’b1101: Combined PWM mode 2. OC1REF has the same behavior as in PWM mode 2. OC1REFC is logic AND between OC1REF and OC2REF. 4’b1110: Asymmetric PWM mode 1. OC1REF has the same behavior as in PWM mode 1. OC1REFC outputs OC1REF in up-counting, OC2REF in down-counting. 4’b1111: Asymmetric PWM mode 2. OC1REF has the same behavior as in PWM mode 2. OC1REFC outputs OC1REF in up-counting, OC2REF in down-counting.	R/W
<3>	**OC1PE**: Output compare 1 preload enable 1’b0: Preload CCR1 disabled 1’b1: Preload CCR1 enabled	R/W
<2>	**OC1FE**: Output compare 1 fast enable 1’b0: CC1 behaves normally depending on counter and CCR1 values when the trigger is ON. 1’b1: An active edge on the trigger input acts like a compare match on CC1 output.	R/W

Bit	Description	Type
<1:0>	**CC1S[1:0]**: Capture/compare 1 selection 2'b00:	R/W

16.5.8 TIM_CHMD1 (Input)

Base address: 0x4000_1800
Address offset: 0x18
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:12>	**IC2F[3:0]**: Input capture 2 filter Refer to IC1F[3:0]	R/W
<11:10>	**IC2PSC[1:0]**: Input capture 2 prescaler Refer to IC2PSC[1:0]	R/W
<9:8>	**CC2S[1:0]**: Capture/compare 2 selection Refer to CC1S[1:0]	R/W
<7:4>	**IC1F[3:0]**: Input capture 1 filter 4'b0000: No filter 4'b0001: $f_{SAMPLING}=f_{CK_INT}$, N=2 4'b0010: $f_{SAMPLING}=f_{CK_INT}$, N=4 4'b0011: $f_{SAMPLING}=f_{CK_INT}$, N=8 4'b0100: $f_{SAMPLING}=f_{DTS}/2$, N=6 4'b0101: $f_{SAMPLING}=f_{DTS}/2$, N=8 4'b0110: $f_{SAMPLING}=f_{DTS}/4$, N=6 4'b0111: $f_{SAMPLING}=f_{DTS}/4$, N=8 4'b1000: $f_{SAMPLING}=f_{DTS}/8$, N=6 4'b1001: $f_{SAMPLING}=f_{DTS}/8$, N=8 4'b1010: $f_{SAMPLING}=f_{DTS}/16$, N=5 4'b1011: $f_{SAMPLING}=f_{DTS}/16$, N=6 4'b1100: $f_{SAMPLING}=f_{DTS}/16$, N=8 4'b1101: $f_{SAMPLING}=f_{DTS}/32$, N=5 4'b1110: $f_{SAMPLING}=f_{DTS}/32$, N=6 4'b1111: $f_{SAMPLING}=f_{DTS}/32$, N=8	R/W
<3:2>	**IC1PSC[1:0]**: Input capture 1 prescaler 2'b00: no prescaler 2'b01: Capture is done pre 2 events 2'b10: Capture is done pre 4 events 2'b11: Capture is done pre 8 events	R/W
<1:0>	**CC1S[1:0]**: Capture/compare 1 selection	R/W

Bit	Description	Type
	2'b00: Channel 1 is output 2'b01: Channel 1 is input, IC1 is mapped on TI1 2'b10: Channel 1 is input, IC1 is mapped on TI2 2'b11: Channel 1 is input, IC1 is mapped on TRC	

16.5.9 TIM_CHMD2 (Output)

Base address: 0x4000_1800
Address offset: 0x1C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:25>	Reserved.	R/W
<23:17>	Reserved.	R/W
<15>	**OC4CE**: Output compare 4 clear enable Refer to OC1CE	R/W
<24,14:12>	**OC4M[3:0]**: Output compare 4 mode Refer to OC1M[3:0]	R/W
<11>	**OC4PE**: Output compare 4 preload enable Refer to OC1PE	R/W
<10>	**OC4FE**: Output compare 4 fast enable Refer to OC1FE	R/W
<9:8>	**CC4S[1:0]**: Capture/compare 4 selection Refer to CC1S[1:0]	R/W
<7>	**OC3CE**: Output compare 3 clear enable Refer to OC1CE	R/W
<16,6:4>	**OC3M[3:0]**: Output compare 3 mode. Refer to OC1M[3:0]	R/W
<3>	**OC3PE**: Output compare 3 preload enable Refer to OC1PE	R/W
<2>	**OC3FE**: Output compare 3 fast enable Refer to OC1FE	R/W
<1:0>	**CC3S[1:0]**: Capture/compare 3 selection Refer to CC1S[1:0]	R/W

16.5.10 TIM_CHMD2 (Input)

Base address: 0x4000_1800
Address offset: 0x1C
Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:12>	**IC4F[3:0]**: Input capture 4 filter Refer to IC1F[3:0]	R/W
<11:10>	**IC4PSC[1:0]**: Input capture 4 prescaler Refer to OC1PSC[1:0]	R/W
<9:8>	**CC4S[1:0]**: Capture/compare 4 selection Refer to CC1S[1:0]	R/W
<7:4>	**IC3F[3:0]**: Input capture 3 filter Refer to IC1F[3:0]	R/W
<3:2>	**IC3PSC[1:0]**: Input capture 3 prescaler Refer to IC1PSC[1:0]	R/W
<1:0>	**CC3S[1:0]**: Capture/compare 3 selection Refer to CC1S[1:0]	R/W

16.5.11 TIM_CHEP

Base address: 0x4000_1800
Address offset: 0x20
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:22>	Reserved.	R/W
<21>	**CC6P**: Capture/compare 6 output polarity Refer to CC1P	R/W
<20>	**CC6E**: Capture/compare 6 output enable Refer to CC1E	R/W
<19:18>	Reserved	R/W
<17>	**CC5P**: Capture/compare 5 output polarity Refer to CC1P	R/W
<16>	**CC5E**: Capture/compare 5 output enable Refer to CC1E	R/W
<15>	**CC4NP**: Capture/compare 4 complementary output polarity Refer to CC1NP	R/W
<14>	Reserved	R/W
<13>	**CC4P**: Capture/compare 4 output polarity Refer to CC1P	R/W
<12>	**CC4E**: Capture/compare 4 output enable Refer to CC1E	R/W

Bit	Description	Type
<11>	**CC3NP**: Capture/compare 3 complementary output polarity Refer to CC1NP	R/W
<10>	**CC3NE**: Capture/compare 3 complementary output enable Refer to CC1NE	R/W
<9>	**CC3P**: Capture/compare 3 output polarity Refer to CC1P	R/W
<8>	**CC3E**: Capture/compare 3 output enable Refer to CC1E	R/W
<7>	**CC2NP**: Capture/compare 2 complementary output polarity Refer to CC1NP	R/W
<6>	**CC2NE**: Capture/compare 2 complementary output enable Refer to CC1NE	R/W
<5>	**CC2P**: Capture/compare 2 output polarity Refer to CC1P	R/W
<4>	**CC2E**: Capture/compare 2 output enable Refer to CC1E	R/W
<3>	**CC1NP**: Capture/compare 1 complementary output polarity 1'b0: OC1N active high (output mode) 1'b1: OC1N active low (output mode) In input mode, this bit is conjunction with CC1P	R/W
<2>	**CC1NE**: Capture/compare 1 complementary output enable 1'b0: Off. OC1N is not active. 1'b1: On. OC1N is active.	R/W
<1>	**CC1P**: Capture/compare 1 output polarity 1'b0: OC1 active high (output mode)/Edge sensitivity selection (input mode) 1'b1: OC1 active low (output mode) /Edge sensitivity selection (input mode) In input mode, CC1NP=0, CC1P=0: non-inverted/rising edge. CC1NP=0, CC1P=1: Inverted/falling edge. CC1NP=1, CC1P=1: non-inverted/both edges. CC1NP=1, CC1P=0: reserved.	R/W
<0>	**CC1E**: Capture/compare 1 output enable 1'b0: Capture mode disabled/OC1 is not active 1'b1: Capture mode enabled/OC1 is active.	R/W

16.5.12 TIM_CNTR

Base address: 0x4000_1800
Address offset: 0x24
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31>	**UIFCP**: Update flag copy	R/W
<30:16>	Reserved	R/W
<15:0>	**CNT[15:0]**: Counter value	R/W

16.5.13 TIM_PSCR

Base address: 0x4000_1800
Address offset: 0x28
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**PSE[15:0]**: Prescaler value	R/W

16.5.14 TIM_PRD

Base address: 0x4000_1800
Address offset: 0x2C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**ARR[15:0]**: Auto-load value	R/W

16.5.15 TIM_REP

Base address: 0x4000_1800
Address offset: 0x30
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**REP[15:0]**: Repetition counter value	R/W

16.5.16 TIM_CHR1

Base address: 0x4000_1800

Address offset: 0x34
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR1[15:0]**: Capture/compare value for channel 1	R/W

16.5.17 TIM_CHR2

Base address: 0x4000_1800
Address offset: 0x38
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR2[15:0]**: Capture/compare value for channel 2	R/W

16.5.18 TIM_CHR3

Base address: 0x4000_1800
Address offset: 0x3C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR3[15:0]**: Capture/compare value for channel 3	R/W

16.5.19 TIM_CHR4

Base address: 0x4000_1800
Address offset: 0x40
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR4[15:0]**: Capture/compare value for channel 4	R/W

16.5.20 TIM_BKDT

Base address: 0x4000_1800
Address offset: 0x44
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15>	**MOE**: Main output enable 1'b0: OC and OCN output are disabled 1'b1: OC and OCN output are enabled	R/W
<14>	**AOE**: Automatic output enable 1'b0: Bit[16] can be set only by software 1'b1: Bit[16] can be ser by software or automatically at the next update event.	R/W
<13:12>	Reserved	R/W
<11>	**OSSR**: Off-state selection for run mode (Bit[16] =1) 1'b0: OC/OCN output are disabled when inactive 1'b1: OC/OCN output are enabled when inactive	R/W
<10>	**OSSI**: Off-state selection for idle mode (Bit[16] = 0) 1'b0: OC/OCN output are disabled when inactive 1'b1: OC/OCN output are enabled when inactive	R/W
<9:8>	**LOCK[1:0]**: Lock configuration 2'b00: lock off 2'b01: lock level 1, BKDT.bit[14], BKDT.bit[11], BKDT.bit[10], BKDT.bit[7:0],CTRL2.bit[18], CTRL2.bit[16], CTRL2.bit[14], CTRL2.bit[13], CTRL2.bit[12], CTRL2.bit[11], CTRL2.bit[10], CTRL2.bit[9], CTRL2.bit[8] are write protected 2'b10: lock level 2. Level 1 + CC polarity bits (CHEP) are write protected 2'b11: lock level 3. Level 2 + CC control bits (CHMDx) are write protected	R/W
<7:0>	**DT[7:0]**: Dead-time generator setup 8'b0xxxxxxx: bit[7:0] * t_{DTS} 8'b10xxxxxx: (64+bit[5:0])*2 * t_{DTS} 8'b110xxxxx: (32+bit[4:0])*8 * t_{DTS} 8'b111xxxxx: (32+bit[4:0])*16** t_{DTS}	R/W

16.5.21 TIM_AUX

Base address: 0x4000_1800
Address offset: 0x50
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:2>	Reserved.	R/W
<1:0>	**OCREF_CLR[1:0]**: OCREF clear input source selection 2'b00: COMP1 output 2'b01: COMP2 output 2'b10: COMP3 output 2'b11: Reserved	R/W

16.5.22 TIM_CHMD3

Base address: 0x4000_1800
Address offset: 0x54
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:25>	Reserved.	R/W
<23:17>	Reserved	R/W
<15>	**OC6CE**: Output compare 6 clear enable Refer to OC1CE	R/W
<24,14:12>	**OC6M[3:0]**: Output compare 6 mode Refer to OC1M[3:0]	R/W
<11>	**OC6PE**: Output compare 6 preload enable Refer to OC1PE	R/W
<10>	**OC6FE**: Output compare 6 fast enable Refer to OC1FE	R/W
<9:8>	Reserved	R/W
<7>	**OC5CE**: Output compare 5 clear enable Refer to OC1CE	R/W
<16,6:4>	**OC5M[3:0]**: Output compare 5 mode Refer to OC1M[3:0]	R/W
<3>	**OC5PE**: Output compare 5 preload enable Refer to OC1PE	R/W
<2>	**OC5FE**: Output compare 5 fast enable Refer to OC1FE	R/W
<1:0>	Reserved	R/W

16.5.23 TIM_CHR5

Base address: 0x4000_1800
Address offset: 0x58

Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31>	**GC5C3**: Group channel 5 and channel 3 1'b0: No effect of OC5REF on OC3REFC 1'b1: OC3REFC is the logical AND of OC3REFC and OC5REF	R/W
<30>	**GC5C2**: Group channel 5 and channel 2 1'b0: No effect of OC5REF on OC2REFC 1'b1: OC2REFC is the logical AND of OC2REFC and OC5REF	R/W
<29>	**GC5C1**: Group channel 5 and channel 1 1'b0: No effect of OC5REF on OC1REFC 1'b1: OC1REFC is the logical AND of OC1REFC and OC5REF	R/W
<28:16>	Reserved	R/W
<15:0>	**CCR5[15:0]**: Capture/Compare 5 value	R/W

16.5.24 TIM_CHR6

Base address: 0x4000_1800
Address offset: 0x5C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**CCR6[15:0]**: Capture/compare 6 value	R/W

16.5.25 TIM_BRK

Base address: 0x4000_1800
Address offset: 0x60
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:18>	Reserved.	R/W
<17:14>	**ETRSEL[3:0]**: ETR source selection 4'b0000: ETR legacy mode 4'b0001: COMP1 output 4'b0010: COMP2 output 4'b0011: 4'b0100: 4'b0101:	R/W

	4’b0110: COMP3 output	
<13:0>	Reserved	R/W

16.5.26 TIM_ISCLT

Base address: 0x4000_1800
Address offset: 0x68
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:28>	Reserved.	R/W
<27:24>	**TI4SEL[3:0]**: Select TI4 input 4’b0000: TIMG_CH4 input Others: Reserved	R/W
<23:20>	Reserved	R/W
<19:16>	**TI3SEL[3:0]**: Select TI3 input 4’b0000: TIMG_CH3 input 4’b0001: COMP3 output Others: Reserved	R/W
<15:12>	Reserved	R/W
<11:8>	**TI2SEL[3:0]**: Select TI2 input 4’b0000: TIMG_CH2 input 4’b0001: COMP2 output Others: Reserved	R/W
<7:4>	Reserved	R/W
<3:0>	**TI1SEL[3:0]**: Select TI1 input 4’b0000: TIMG_CH1 input 4’b0001: COMP1 output Others: Reserved	R/W

16.5.27 TIMG register map

The base address of the TIMB registers is 0x4000_1800.
The following table gives the TIMG register map and reset values.

Table 42 TIMG register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIM_CTRL1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIFREMAP	Res	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value	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
0x04	TIM_CTRL2	Res	Res	Res	Res	Res	Res	Res	Res	MMS2[3:0]				Res	OIS6	Res	OIS5	Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res	CCPC
	Reset value	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
0x08	TIM_SLVMD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TS[4:3]		Res	Res	Res	SMS[3]	ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
	Reset value	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
0x0C	TIM_REQEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value	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
0x10	TIM_STATUS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6IF	CC5IF	Res	Res	Res	CC4OF	CC3OF	CC2OF	CC1OF	Res	Res	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value	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
0x14	TIM_EVT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
	Reset value	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
0x18	TIM_CHMD1 (Output)	Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	Reset value	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
	TIM_CHMD1 (Input)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
	Reset value	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
0x1C	TIM_CHMD2 (Output)	Res	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
	Reset value	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
	TIM_CHMD2 (Input)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
	Reset value	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
0x20	TIME_CHEP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6P	CC6E	Res	Res	CC5P	CC5E	CC4NP	Res	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value	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
0x24	TIM_CNTR	UIFCP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	Reset value	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
0x28	TIM_PSCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	Reset value	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
0x2C	TIM_PRD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	TIM_REP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REP[15:0]															
	Reset value	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
0x34	TIM_CHR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR1[15:0]															
	Reset value	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
0x38	TIM_CHR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR2[15:0]															
	Reset value	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
0x3C	TIM_CHR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR3[15:0]															
	Reset value	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
0x40	TIM_CHR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR4[15:0]															
	Reset value	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
0x44	TIM_BKDT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MOE	AOE	Res	Res	OSSR	OSSI	LOCK[1:0]		DT[7:0]							
	Reset value	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
0x50	TIM_AUX	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OCREF_CLR[1:0]	
	Reset value	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
0x54	TIM_CHMD3	Res	Res	Res	Res	Res	Res	Res	OC6M[3]	Res	Res	Res	Res	Res	Res	Res	OC5M[3]	OC6CE	OC6M[2:0]			OC6PE	OC6FE	Res	Res	OC5CE	OC5M[2:0]			OC5PE	OC5FE	Res	Res
	Reset value	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
0x58	TIM_CHR5	GC5C3	GC5C2	GC5C1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR5[15:0]															
	Reset value	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
0x5C	TIM_CHR6	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR6[15:0]															
	Reset value	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
0x60	TIM_BRK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		ETRSEL[3:0]			Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value	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	1
0x68	TIM_ISLCT	Res	Res	Res	Res	TI4SEL[3:0]				Res	Res	Res	Res	TI3SEL[3:0]				Res	Res	Res	Res	TI2SEL[3:0]				Res	Res	Res	Res	TI1SEL[3:0]			
	Reset value	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

17 Basic Timers (TIMB)

17.1 Introduction

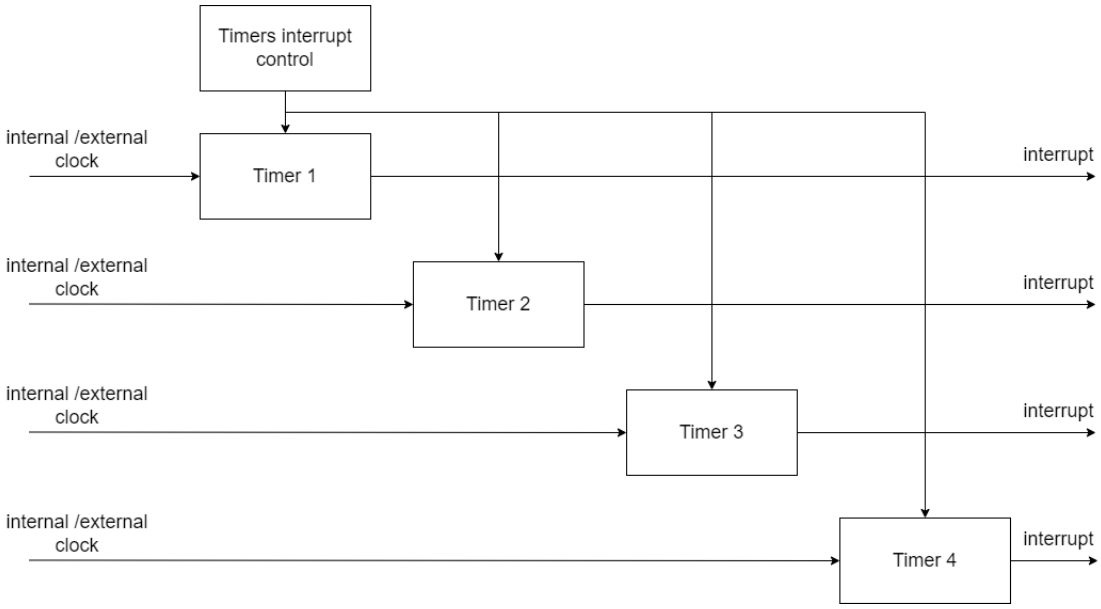


Figure 34 Diagram of basic timer

17.2 TIMB main features

- 32-bit down counter
- Free-running mode and user-defined count mode
  - Free-running mode: Timer loads the maximum value and counter wrapping to its maximum value.
  - User-defined count mode: Timer loads the current value of the TIMERx_LDR.
- Interrupt generation with mask
- Load initial value when Timer is enabled after reset or disabled or timer counts to 0.

17.3 TIMB signal descriptions

The basic timer signals are shown in the following table.

Table 43 TIMB signal descriptions

Signal	Description	I/O
BTIN	External clock input	I
BTOUT	Toggle output	O

17.4 TIMB functional description

There are 2 Basic Timer, TIMER1x and TIMER2x, in XT32Hxx series. Each Basic Timer supports up to 4 separately programmable Timers with 32-bit counter width. The Timers count down from a programmed value and generate an interrupt when the count reaches zero. The Basic Timer supports two modes of operation which can be selected by setting the MODE bit of TIMERx_CCR register. When the MODE bit is set to 1, the Timer operates in “User-defined count mode”. The Timer loads the current value of the TIMERx_LDR register when the Timer is enabled or if the count goes to 0. When the MODE bit is set to 0, the Timer operates in “Free-running mode”. The Timer sets the 32-bit counter to its maximum value with all its 32 bits loaded with 1. The timer counter wrapping to its maximum value allows time to reprogram or disable the timer before another interrupt.

The clock source for the Timer counter is selectable from either an internal clock source: HCLK, PCLK, or PLL, or from an external clock source. The internal clock source can be divided down or adjusted by setting the divider ratio or configuration in the ICG_HCLK_DIV, ICG_PCLK_DIV, or ICG_PLL_DIV register before applying to the Timer. The external clock source can be applied to the Timer by setting the relevant ICG_TIMER_EXCLK_ENABLE_1 register bits [7:4] and bits [3:0] to 1. Please refer to the Internal clock source (ICG) section 1.3 and 1.4 for more information on Basic Timer clock selection and configuration.

The Timer can be enabled by writing 1 to bit 0 of the TIMERx_CCR register. When a timer is enabled, it loads the initial count value and decrements the count value on each rising edge of the counter clock. When a timer is disabled, the timer would stop running and its associated registers in the timer clock domain would also be reset.

The Basic Timer generates an output that toggles whenever the timer counter reaches 0. The respective timer toggling outputs can be observed at the IO pins by configuring the Port Mux (PADI) to CFG2_AUX1 or CFG2_AUX2. Please refer to the Table 23 XT32H05x IO assignment in the Port mux Chapter, section 1.1 and section 1.4 for more information on Port Mux configuration.

17.4.1 Usage flow

1. Initialize timer
   a. Disable timer by writing 0 to TIMERx_CCR.EN[0].
   b. Set counter mode by programming TIMERx_CCR.MODE[1].
   c. Set appropriate values and interrupt mask TIMERX_CCR.MASK[2].
2. Load counter value to TIMERx_LDR[31:0]
3. Enable timer by writing 1 to TIMERx_CCR.EN[0].

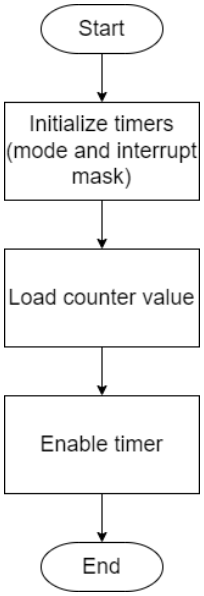


Figure 35 Basic timer use flow

17.5 TIMB registers

17.5.1 TIMERx_LDR

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset:

TIMERx1 0x00

TIMERx2 0x14

TIMERx3 0x28

TIMERx4 0x3C

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_LOAD[31:0]**: Load count Set the value to be loaded to the Timer counter.	R/W

17.5.2 TIMERx_CCVR

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset:

TIMERx1 0x04

TIMERx2 0x18

TIMERx3 0x2C

TIMERx4 0x40

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_VALUE[31:0]**: Current value This register shows the current Timer counter value.	R

17.5.3 TIMERx_CCR

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset:

TIMERx1 0x08

TIMERx2 0x1C

TIMERx3 0x30

TIMERx4 0x44

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved	R/W
<2>	**MASK**: Interrupt mask 1'b0: Not masked 1'b1: Masked	R/W
<1>	**MODE**: Timer mode 1'b0: Free-running mode 1'b1: User-defined count mode	R/W
<0>	**EN**: Timer enable 1'b0: Disable 1'b1: Enable	R/W

17.5.4 TIMERx_EOI

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset:

TIMERx1 0x0C

TIMERx2 0x20

TIMERx3 0x34

TIMERx4 0x48

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R
<0>	**EOI**: Read to clear interrupt Read this bit to clear the interrupt from the Timer.	R

17.5.5 TIMERx_STAT

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset:

TIMERx1 0x10

TIMERx2 0x24

TIMERx3 0x38

TIMERx4 0x4C

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R
<0>	**STAT**: Interrupt status This bit reflects the interrupt status for a specific Timer.	R

17.5.6 TIMERsINTSTAT

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset: 0xA0

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved	R
<3>	**TIMER4_STAT**: Timerx4 interrupt status This bit reflects the interrupt status for Timerx4.	R
<2>	**TIMER3_STAT**: Timerx3 interrupt status This bit reflects the interrupt status for Timerx3.	R
<1>	**TIMER2_STAT**: Timerx2 interrupt status This bit reflects the interrupt status for Timerx2.	R
<0>	**TIMER1_STAT**: Timerx1 interrupt status This bit reflects the interrupt status for Timerx1.	R

17.5.7 TIMERsEOI

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset: 0xA4

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved	R
<3>	**TIMER4_EOI**: Timerx4 End-of-Interrupt Read to clear Timerx4 interrupt	R
<2>	**TIMER3_EOI**: Timerx3 End-of-Interrupt Read to clear Timerx3 interrupt	R
<1>	**TIMER2_EOI**: Timerx2 End-of-Interrupt Read to clear Timerx2 interrupt	R
<0>	**TIMER1_EOI**: Timerx1 End-of-Interrupt Read to clear Timerx1 interrupt	R

17.5.8 TIMERsRAWINTSTAT

Base address:

TIMER1x 0x4000_1400

TIMER2x 0x4000_1500

Address offset: 0xA8

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved	R
<3>	**TIMER4_RSTAT**: Timerx4 interrupt raw status	R
<2>	**TIMER3_RSTAT**: Timerx3 interrupt raw status	R
<1>	**TIMER2_RSTAT**: Timerx2 interrupt raw status	R
<0>	**TIMER1_RSTAT**: Timerx1 interrupt raw status	R

17.5.9 TIMB register map

The base address of the TIM1x registers is 0x4000_1400.
The base address of the TIMB2x registers is 0x4000_1500.
The following table gives the TIMB register map and reset values.

Table 44 TIMB register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMER1_LDR	CNT_LOAD[31:0]																															
	Reset value	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
0x04	TIMER1_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x08	TIMER1_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MASK	MODE	EN
	Reset value	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
0x0C	TIMER1_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EOI
	Reset value	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
0x10	TIMER1_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STAT
	Reset value	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
0x14	TIMER2_LDR	CNT_LOAD[31:0]																															
	Reset value	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
0x18	TIMER2_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x1C	TIMER2_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MASK	MODE	EN
	Reset value	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
0x20	TIMER2_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EOI
	Reset value	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
0x24	TIMER2_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STAT
	Reset value	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
0x28	TIMER3_LDR	CNT_LOAD[31:0]																															
	Reset value	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
0x2C	TIMER3_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x30	TIMER3_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MASK	MODE	EN
	Reset value	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
0x34	TIMER3_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EOI
	Reset value	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
0x38	TIMER3_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STAT
	Reset value	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
0x3C	TIMER4_LDR	CNT_LOAD[31:0]																															

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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
0x40	TIMER4_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x44	TIMER4_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MASK	MODE	EN
	Reset value	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
0x48	TIMER4_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EOI
	Reset value	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
0x4C	TIMER4_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STAT
	Reset value	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
0xA0	INT_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIMER4_STAT	TIMER3_STAT	TIMER2_STAT	TIMER1_STAT
	Reset value	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
0xA4	INT_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIMER4_EOI	TIMER3_EOI	TIMER2_EOI	TIMER1_EOI
	Reset value	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
0xA8	INT_RSTAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIMER4_RSTAT	TIMER3_RSTAT	TIMER2_RSTAT	TIMER1_RSTAT
	Reset value	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

18 Independent watchdog (WDTI)

18.1 Introduction

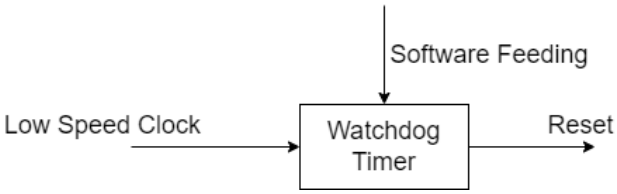


Figure 36 Independent watchdog timer

18.2 WDTI main features

- 32-bit down counter
- Wraps
- User restart
- Kicking the dog
- Interrupt generator

18.3 WDTI functional description

After user start WDTI, the WDTI count load the initial value and start to count. If the user did not kick the dog, system will be reset when WDTI counter count to 0. When the user kicks dog, the counter will be load the initial value and restarted.

WDTI share clock source with RTC. Set AON_CFG.CFG_32KCLKSRC[1:0] to select clock source, b’00 selected 32k RC clock and b’01 selected 32K external crystal clock.

18.4 WDTI registers

18.4.1 WDT_CR(WDT control register)

Base address: 0x4000_3400
Address offset: 0x00
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:5>	Reserved.	R/W
<4:2>	**RPL[2:0]**:Reset pulse length 3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles	R/W
<1>	**RMOD**: Response mode 1'b0: generate a system reset 1'b1: first generate an interrupt and if it is not cleared a second timeout ocurrs then generate a system reset	R/W
<0>	**EN**: WDT enable	R/W

18.4.2 WDT_TORR(WDT timeout period register)

Base address: 0x4000_3400
Address offset: 0x04
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:4>	**TOP_INIT[3:0]**: Timeout period for initialization $t = 2^{(16+i)}$	R/W
<3:0>	**TOP[3:0]**: Timeout period $t = 2^{(16+i)}$	R/W

18.4.3 WDT_CCVR(WDT current counter value register)

Base address: 0x4000_3400
Address offset: 0x08
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_VALUE[31:0]**: Current counter value	R/W

18.4.4 WDT_CRR(WDT restart register)

Base address: 0x4000_3400
Address offset: 0x0C
Type: Write
Reset value: 0x7FFF_FFFF

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CCR[7:0]**: Restart counter by 0x76 is written.	R/W

18.4.5 WDTI register map

The base address of the WDTI registers is 0x4000_3400.
The following table gives the WDTI register map and reset values.

Table 45 WDTI register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WDT_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RPL[2:0]			RMOD	EN
	Reset value	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
0x04	WDT_TORR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TOP[3:0]			
	Reset value	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
0x08	WDT_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x0C	WDT_CRR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR[7:0]							
	Reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

19 Windows watchdog (WDTW)

19.1 Introduction

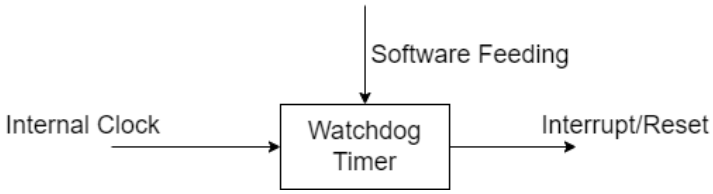


Figure 37 Windows watchdog timer

19.2 WDTW main features

- 32-bit down counter
- Interrupt generator

19.3 WDTW functional description

1. Calculate operation based on window size and counter range. Set WDTW operation clock source by programming ICG_CLK_CFG.CLK_CTL_WDTWSRC[3:2].

   b'00 means WDTW use APB clock as source, b'01 means WDTW use AHB clock as source, b'10 means WDTW use PLL output as source.

2. Initialize WDTW pulse width by programming WDT_CR.RPL[4:2] and counter range by programming WDT_TORR.TOP[3:0].
3. Set WDTW to interrupt mode. Write 1 to WDT_CR.RMOD[1].
4. Enable interrupt. Enable NVIC.IRQ_WDTW.
5. Start WDTW to write 1 to WDT_CR.EN[0].
6. If the user kick dog, WDTW will be restart by write 0x76 to WDT_CRR.
7. When WDTW counter counts to 0, the interrupt is generated. The user should handle the IRQ. If the user can not handle the IRQ, WDTW will reload counter and start to count. When WDTW counter counts to 0 again, system will be reset.

19.4 WDTW registers

19.4.1 WDT_CR(WDT control register)

Base address: 0x4000_0C00
Address offset: 0x00
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:5>	Reserved.	R/W
<4:2>	**RPL[2:0]**: Reset pulse length 3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles	R/W
<1>	**RMOD**: Response mode 1'b0: generate a system reset 1'b1: first generate an interrupt and if it is not cleared a second timeout ocurrs then generate a system reset	R/W
<0>	**EN**: WDT enable	R/W

19.4.2 WDT_TORR(WDT timeout period register)

Base address: 0x4000_0C00
Address offset: 0x04
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:4>	**TOP_INIT[3:0]**: Timeout period for initialization $t = 2^{(16+i)}$	R/W
<3:0>	**TOP[3:0]**: Timeout period $t = 2^{(16+i)}$	R/W

19.4.3 WDT_CCVR(WDT current counter value register)

Base address: 0x4000_3400
Address offset: 0x08
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_VALUE[31:0]**: Current counter value	R/W

19.4.4 WDT_CRR(WDT restart counter register)

Base address: 0x4000_0C00
Address offset: 0x0C
Type: Write
Reset value: 0x07FFF_FFFF

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**CCR[7:0]**: Restart counter by 0x76 is written.	R/W

19.4.5 WDT_STAT(WDT status register)

Base address: 0x4000_0C00
Address offset: 0x10
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**STAT**: Interrupt status. 1'b0: interrupt is inactive 1'b1: interrupt is active	R/W

19.4.6 WDT_EOI(WDT clear interrupt register)

Base address: 0x4000_0C00
Address offset: 0x14
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**EOI**: Read to clear Interrupt.	R/W

19.4.7 WDTW register map

The base address of the WDTW registers is 0x4000_0C00.
The following table gives the WDTW register map and reset values.

Table 46 WDTW register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WDT_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RPL[2:0]			RMOD	EN
	Reset value	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
0x04	WDT_TORR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TOP[3:0]			
	Reset value	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
0x08	WDT_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x0C	WDT_CRR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR[7:0]							
	Reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x10	WDT_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STAT
	Reset value	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
0x14	WDT_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EOI
	Reset value	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

20 Real-timer clock (RTC)

20.1 Introduction

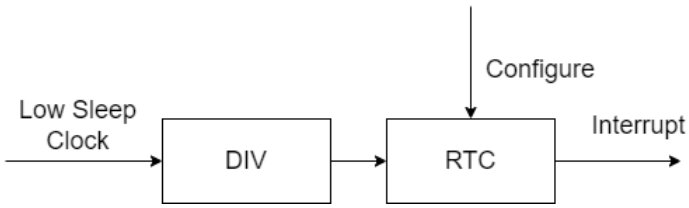


Figure 38 RTC

20.2 RTC main features

RTC implement count-up counter. The counter may load a start value by programming RTC_CLR. When counter reach the maximum value, it wraps to 0 and then continues incrementing.
RTC generates an interrupt when counter reached the match value by programming RTC_CMR. User can polling the interrupt status and clear the interrupt status.
RTC can run in a counter wrap mode by programming RTC_CCR.WEN[3]. When a match occurs, the counter wrap to 0.

20.3 RTC functional description

20.3.1 Configure RTC

1. Set RTC source and reference clock divider by AON

a. Set AON_CFG.CFG_32KCLKSRC[1:0] to select RTC clock source, b’00 selected 32k RC clock and b’01 selected 32K external crystal clock.
b. Set AON_CFG.CFG_RTCRST[31] to select RTC mode.
   b’0 means when counter reach the maximum value, system will be reset.
   b’1 means when counter reach the maximum value, an interrupt will be occurred.
c. Set clock divider in AON_RTC_REFCLK_DIV[31:0].
d. Power source clock. Set 0 to AON_CLK_CTRL.CLKCTRL_32KXTAL_PD[1] or AON_CLK_CTRL.CLKCTRL_32KRC_PD[0] according to RTC clock source.

2. Enable RTC.

a. Set start value to RTC_CLR.CNT_LOAD[31:0].
b. Set 1 to RTC_CCR.EN[2] to start counter.

20.3.2 Set Alarm

1. Set RTC source and reference clock divider by AON, refer to above session.
2. Set counter match value to CMR.

Set match value to RTC_CMR.CBT_MATCH[31:0]. When counter reach to the match value, an interrupt will be occurred.

3. Enable interrupt and Enable RTC

a. Set start value to RTC_CLR.CNT_LOAD[31:0].
b. Set 1 to RTC_CCR.IEN[0] to enable interrupt, set 0 to RTC_CCR.MASK[1] to unmask interrupt, and set 1 to RTC_CCR.EN[2] to start counter.

4. Clear interrupt

Read RTC_EOI to clear RTC interrupt.

20.4 RTC register

20.4.1 RTC_CCVR(RTC current counter value register)

Base address: 0x4000_3500
Address offset: 0x00
Type: Read-only
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_VALUE[31:0]**: Count current value	R/W

20.4.2 RTC_CMR(RTC counter match register)

Base address: 0x4000_3500
Address offset: 0x04
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_MATCH[31:0]**: Interrupt match value	R/W

20.4.3 RTC_CLR

Base address: 0x4000_3500
Address offset: 0x08
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:0>	**CNT_LOAD[31:0]**: Count load value	R/W

20.4.4 RTC_CCR

Base address: 0x4000_3500
Address offset: 0x0C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved	R/W
<3>	**WEN**: Wrap enable 0 – Wrap disable 1 - Wrap enable	R/W

Bit	Description	Type
<2>	**EN**: Counter enable 0 – Counter disable 1 – Counter enable	R/W
<1>	**MASK**: Interrupt mask 0 – Interrupt unmasked 1 – Interrupt masked	R/W
<0>	**IEN**: Interrupt enable 0 - Interrupt disable 1 – Interrupt enable	R/W

20.4.5 RTC_STAT

Base address: 0x4000_3500
Address offset: 0x10
Type: Read-only
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**STAT**: Interrupt status 0 – interrupt is inactive 1 – Interrupt is active	R/W

20.4.6 RTC_RSTAT

Base address: 0x4000_3500
Address offset: 0x14
Type: Read-only
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**RSTAT**: Interrupt raw status 0 – interrupt is inactive 1 – Interrupt is active	R/W

20.4.7 RTC_EOI

Base address: 0x4000_3500
Address offset: 0x18
Type: Read-only
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved	R/W
<0>	**EOI**: The match interrupt is cleared by read this location.	R/W

20.4.8 RTC register map

The base address of the RTC registers is 0x4000_3500.
The following table gives the RTC register map and reset values.

Table 47 RTC register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RTC_CCVR	CNT_VALUE[31:0]																															
	Reset value	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
0x04	RTC_CMR	CNT_MATCH[31:0]																															
	Reset value	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
0x08	RTC_CLR	CNT_LOAD[31:0]																															
	Reset value	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
0x0C	RTC_CCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WEN	EN	MASK	IEN
	Reset value	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
0x10	RTC_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STAT
	Reset value	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
0x14	RTC_RSTAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RSTAT
	Reset value	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
0x18	RTC_EOI	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EOI
	Reset value	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

21 Inter-integrated circuit (I2C)

21.1 Introduction

XT32H05x serial provides two I^2C (inter-integrated circuit) bus interface which handles communications between microcontroller and the serial I^2C bus.

21.2 I2C main features

- Slave and master modes
- Standard mode (up to 100Kbps)
- Fast mode (up to 400Kbps)
- Fast mode plus (up to 1Mbps)
- High speed mode (up to 3.4Mbps)
- 7-bit and 10-bit addressing mode
- General call
- Programmable setup and hold times
- Programmable digital noise filter
- DMA capability
- 8-bytes buffer

21.3 I2C signal descriptions

The I2C signals are shown in the following table.

Table 48 I2C signal descriptions

Signal	Description	I/O
SCL	Bidirectional serial clock	I/O
SDA	Bidirectional serial data	I/O

21.4 I2C functional description

21.4.1 Operation Mode

The I2C module can operate in one of the four following modes:

- Slave transmit
- Slave receive
- Master transmit
- Master receive

By default, it operates in master mode after reset.

21.4.2 Communication flow

In Master mode, the I2C module initiates a data transfer and generates the clock signal. A serial data transfer begins with a START condition and ends with a STOP condition. The START condition is generated when data are written to the transmit FIFO. Transmission The STOP condition is generated automatically when the Tx FIFO is empty.

In Slave mode, the I2C module is capable of recognizing its own 7-bit or 10-bit slave address as well as the general call address. The general call address detection can be enabled or disabled through software.

Data and addresses are transferred as 8-bit bytes starting with the MSB bit. The first byte following the START condition contain the slave address to be addressed. A second address byte is issued if the I2C slave has a 10-bit slave address. The address is always transmitted in the Master mode. Upon a successful receive of address/data, the receiving device would issue an acknowledge bit to the transmitter. The following diagram shows a typical I2C transmission in 7-bit addressing mode:

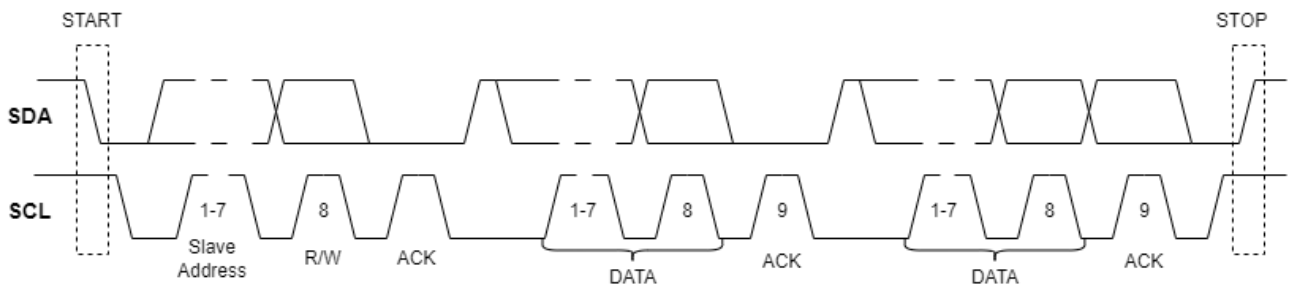


Figure 1. I2C transmission protocol (7-bit addressing)

The master can initiate data transmission and reception to/from the I2C bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit/receive data to/from the bus, acting as either a slave-transmitter or slave-receiver. It is possible for the master to transmit data consecutively to different I2C slave by first transmitting a repeated START condition and then address another I2C slave without first generating a STOP condition.

21.4.3 Register access

The following list of registers should only be written when I2C is disabled.

- I2C_CON, I2C_TAR, I2C_SAR, I2C_HS_MADDR
- I2C_SS_SCL_HCNT, I2C_SS_SCL_LCNT, I2C_FS_SCL_HCNT, I2C_FS_SCL_LCNT, I2C_HS_SCL_HCNT, I2C_HS_SCL_LCNT
- I2C_SDA_HOLD, I2C_SDA_SETUP
- I2C_FS_SPKLEN, I2C_HS_SPKLEN

21.4.4 Operation mode setting

I2C operates in master mode when bit[0] and bit[6] in I2C_CON are set to 1. And when bit[0] and bit[6] in I2C_CON are set to 0, I2C operates in slave mode. Other settings for bit[0] and bit[6] in I2C_CON is not a valid configuration.

21.4.5 Master data transfer

Perform the following steps to transfer data in I2C master mode:

1. Disable I2C by writing 0 to I2C_ENABLE.ENABLE[0].
2. Configure I2C (program I2C_CON)
   a. Enable RESTART by writing 1 to I2C_CON.RSTRT_EN[5]
   b. Set the master/slave address length (7-bit or 10-bit) by programming I2C_CON.10ADDR_MASTER[4]/I2C_CON.10ADDR_SLAVE[3].
   c. Set speed mode by programming I2C_CON.SPEED[2:1].
   d. Disable slave and enable master by writing 1 to I2C_CON.SLVDISABLE[6] and I2C_CON.MASTER_MODE[0].
3. Program the target address to I2C_TAR.TAR[9:0].
4. Program the SCL period (Set xS_HCNT and xS_LCNT), refer to the clock configuration section.
5. Unmask the interrupt by writing 1 to the relevant bit in I2C_INT_MASK.
6. Set RX/TX FIFO threshold level that triggers the RX_FULL/TX_EMPTY interrupt by programming I2C_RX_TL and I2C_TX_TL.
7. Enable I2C by writing 1 to I2C_ENABLE.ENABLE[0].
8. Write CMD and DATA to I2C_DATA_CMD[8:0].
9. For write command,
   a. Wait for the TX_EMPTY status or interrupt in I2C_INT_RAWSTAT or I2C_INT_STAT.
   b. Loop back to step 8 if there is more command/data to send. Otherwise, slip to step 11.
10. For read command,
    a. Wait for the RX_FULL status or interrupt in I2C_INT_RAWSTAT or I2C_INT_STAT.
    b. Read the data from DATA_CMD and loop back to step 9.a

11. Check to see if I2C is active.
12. If I2C is inactive and no additional data need to be transmitted, disable I2C.

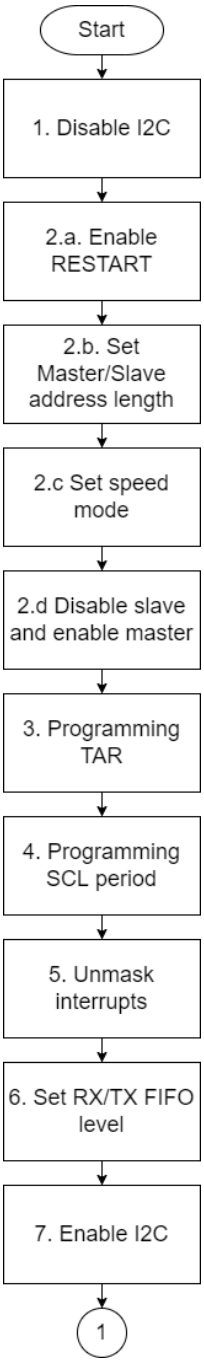


Figure 39 I2C master transfer data -1

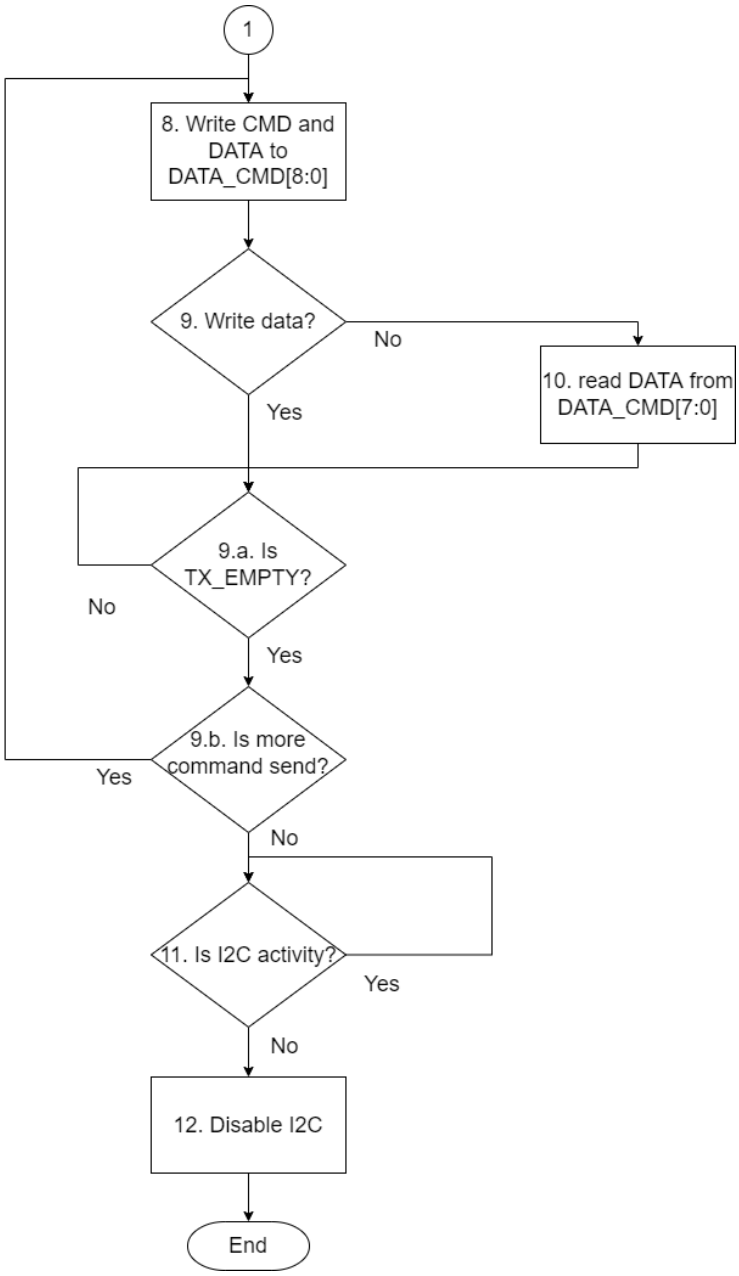


Figure 40 I2C master transfer data-2

21.4.6 Slave data transfer

Perform the following steps to transfer data in I2C slave mode:

1. Disable I2C by writing 0 to I2C_ENABLE.ENABLE[0].
2. Program the slave address to I2C_SAR.SAR[9:0].
3. Configure I2C (program the I2C_CON)
   a. Enable RESTART by writing 1 to I2C_CON.RSTRT_EN[5]
   b. Set the master/slave address length (7-bit or 10-bit) by programming I2C_CON.10ADDR_MASTER[4]/I2C_CON.10ADDR_SLAVE[3].
   c. Set speed mode by programming I2C_CON.SPEED[2:1].
   d. Enable slave mode and disable master mode by writing 0 to I2C_CON.SLVDISABLE[6] and I2C_CON. MASTER_MODE[0].
4. Unmask the interrupt by writing 1 to the relevant bit in I2C_INT_MASK.
5. Enable I2C by writing 1 to I2C_ENABLE.ENABLE[0].
6. Check and see if either RD_REQ or RX_FULL interrupt is set. If RD_REQ / RXFULL interrupt is set, then go to case a / b respectively. Otherwise go to step 7.
   a. If both RD_REQ and TX_ABRT are set, the slave transmit operation has been aborted. Clear the TX_ABRT source by reading the I2C_INT_CLR_TXA register. Then write the data to be transmitted to DATA_CMD[7:0] and write 0 to DATA_CMD[8] to indicate that this is a transmit operation. Clear RD_REQ interrupt and go to step 7.
   b. If RX_FULL is set, then read the received data from DATA_CMD[7:0] and go to step 7.
7. Check the I2C activity. If I2C is active, loop back to step 6. If I2C is not active and no additional data need to be transmitted, disable I2C.

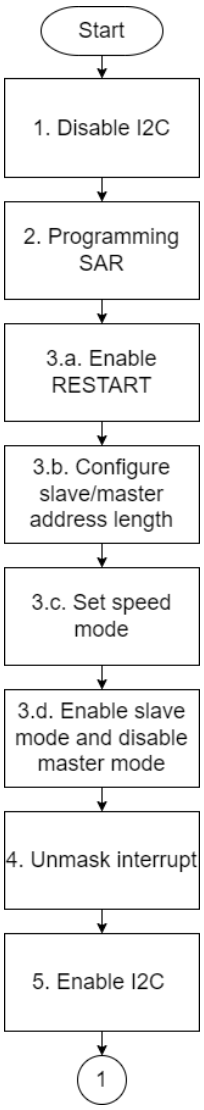


Figure 41 I2C salve transfer data -1

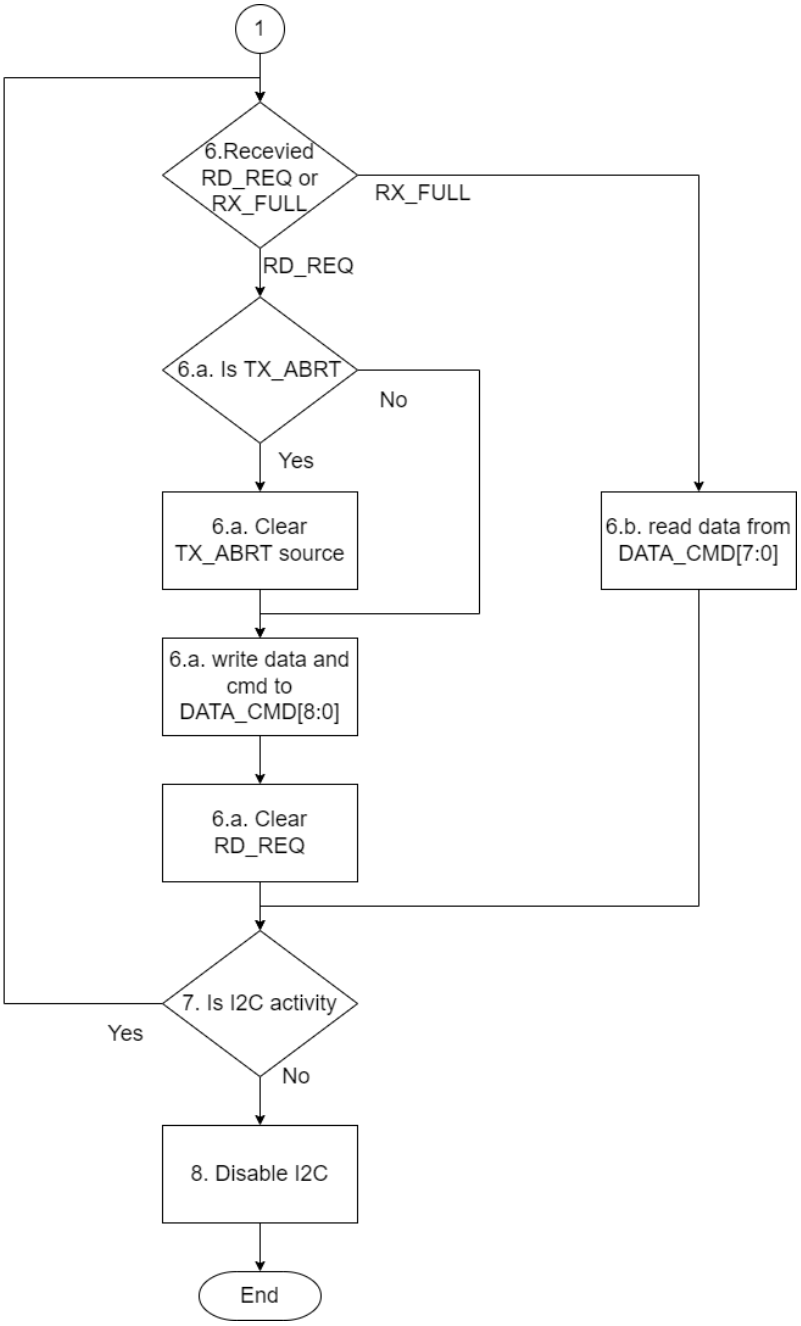


Figure 42 I2C slave transfer data -2

21.4.7 Clock configuration

The I2C module supports different speed mode including:

- Standard mode (up to 100Kbps)
- Fast mode (up to 400Kbps)
- Fast mode plus (up to 1Mbps)
- High speed mode (up to 3.4Mbps)

The minimum value of the spike suppression limit (I2C_FS_SPKLEN / I2C_HS_SPKLEN), the minimum data setup time, SCL low time and SCL high time under different speed mode are summarized in the table below.

Table 49 The minimum time of I2C

Speed mode	Minimum value of I2C_xx_SPKLEN	Minimum SCL Low time (t_{LOW})	Minimum SCL high time(t_{HIGH})	Minimum Data set-up time ($t_{SU;DAT}$)
SS	1	4700ns	4000ns	250ns
FS	1	1300ns	600ns	100ns
FS+	2	500ns	260ns	50ns
HS(400pf)	1	320ns	120ns	10ns
HS(100pf)	1	160ns	60ns	10ns

21.4.7.1 SCL clock (t_{LOW} and t_{HIGH})

This section describes how to determine the value for the I2C SCL low count register (I2C_xx_SCL_LCNT) and the I2C SCL high count register (I2C_xx_SCL_HCNT) at different speed mode.

Due to the hardware design requirements, the following conditions must be met:

1. I2C_xx_SCL_LCNT values must be larger than I2C_xx_SPKLEN+7.
2. I2C_xx_SCL_HCNT values must be larger than I2C_xx_SPKLEN+5.

In addition, please note the actual SCL low time and SCL high time is determined by:

1. The actual SCL low time is I2C_xx_SCL_LCNT + 1
2. The actual SCL high time is I2C_xx_SCL_HCNT + (I2C_xx_SPKLEN + 7)

Since the SCL period, minimum SCL low period, and minimum SCL high period are given by:

$$SCL\ period = \frac{IC\ clk\ @Hz}{BaudRate\ @bps}$$

$$minimun\ SCL\ low\ period = minimum\ SCL\ low\ time\ \times IC\ clk$$

$$minimum\ SCL\ high\ period = minimum\ SCl\ high\ time\ \times IC\ clk$$

As the SCL period must be longer than the sum of the minimum SCL low period and minimum SCL high period, we have:

$$SCL\ period\ \geq (minimum\ SCL\ low\ period\ + minimum\ SCL\ high\ period)$$

The timing difference between the actual SCL period and the minimum SCL period is given by:

$$delta = SCL\ period - (minimum\ SCL\ low\ period + minimum\ SCL\ high\ period)$$

Thus,

$$SCL\ low\ period = minimum\ SCL\ low\ period + delta/2$$

$$SCL\ high\ period = minimum\ SCL\ high\ period + delta/2$$

Since the actual SCL low time is I2C_xx_SCL_LCNT+1 and I2C_xx_SCL_LCNT must be larger than I2C_xx_SPKLEN+7, we have:

$$SCL\ low\ period\ > (I2C_xx_SPKLEN + 7) + 1$$

Similarly, the actual SCL high time is given by:

$$SCL\ high\ period > (I2C_xx_SPKLEN + 7) + (I2C_xx_SPKLEN + 5)$$

The relationship between I2C_xx_SCL_LCNT and I2C_xx_SCL_HCNT in terms of the SCL low period and SCL high period is given by:

$$I2C_xx_SCL_LCNT = SCl\ low\ period - 1$$

$$I2C_xx_SCL_HCNT = SCL\ high\ period - I2C_xx_SPKLEN - 7$$

Notes:

If any one of the above conditions cannot be met, please consider increasing the IC clk or decreasing the transmission baud rate.

21.4.7.2 Data hold time ($t_{HD;DAT}$)

The I2C protocol specification requires 300ns of hold time on the SDA signal in standard mode and fast mode, and a hold time long enough to bridge the logic 1 and 0 transition of the falling edge of SCL in high-speed mode and fast mode plus.

To facilitate the SDA hold time adjustment, the I2C module supports programmable SDA hold time through programming the I2C_SDA_HOLD register.

I2C_SDA_HOLD[15:0] defines the SDA hold time (SDA_TX_HOLD) when the module acts as a transmitter for both slave and master mode (after SCL goes from high to low).

I2C_SDA_HOLD[23:16] defines the SDA hold time (SDA_RX_HOLD) when the module acts as a receiver. This value defines how long the SDA transition would be extended whenever SCL is high in the receiver.

Table 50 and Table 51 summarizes the maximum values for SDA_RX_HOLD and the minimum values for the SDA_TX_HOLD for the I2C module.

Table 50 SDA_RX_HOLD maximum values

Speed mode	Maximum values for SDA_RX_HOLD
SS	I2C_SS_SCL_HCNT – I2C_FS_SPKLEN -3
FS	I2C_FS_SCL_HCNT – I2C_FS_SPKLEN -3
FS+	I2C_FS_SCL_HCNT – I2C_FS_SPKLEN -3
HS(400pf)	Min{I2C_FS_SCL_HCNT – I2C_FS_SPKLEN -3, (I2C_HS_SCL_LCNT/2} – I2C_HS_SPKLEN -3}
HS(100pf)	Min{I2C_FS_SCL_HCNT – I2C_FS_SPKLEN -3, I2C_HS_SCL_LCNT – I2C_HS_SPKLEN -3}

Table 51 SDA_TX_HOLD minimum values

Operation mode	Minimum values for SDA_TX_HOLD
Master	1
Slave	I2C_xx_SPKLEN +7

21.4.7.3 Data set-up time ($t_{SU;DAT}$)

$$SDA\ SETUP = IC\ clk \times minimum\ Data\ setup\ time$$

21.4.8 DMA transfer

XT32H0 I2C support data transfer through DMA control for improved performance. The following describes how to configure the I2C module for DMA transfer:

1. Configure the I2C for master/slave data transfer. Please refer to Section 1.4.5 Master data transfer or Section 1.4.6 Slave data transfer for more information.
2. Enable the I2C DMA and configure the DMA threshold level.
   a. Enable I2C DMA receive handshaking by writing 1 to the I2C_DMA_CR.RDMAE[0]. Or enable I2C DMA transmit handshaking by writing 1 to the I2C_DMA_CR.TDMAE[1].
   b. For DMA receive operation, set the I2C DMA receive FIFO threshold in the I2C_DMA_RDLR register. Whenever the receive FIFO is at or above the threshold, i.e. I2C_DMA_RDLR+1, the DMA receive request is triggered.
   c. For DMA transmit operation, set the I2C DMA transmit FIFO threshold in the I2C_DMA_TDLR register. Whenever the transmit FIFO is less than or equal to the threshold, i.e. I2C_DMA_TDLR, the DMA transmit request is triggered.
3. Configure the DMA hardware handshaking interface as described in section 7.3.4 of the System Configuration Controller Chapter.
4. Configure the DMA controller for DMA receive or DMA transmit operations as described in the Direct Memory Access Controller Chapter.

5. Associate the DMA controller with the I2C module (using HAL APIs).
6. Enable the DMA channel as described in section 1.3.6 Data Transfer programming flow of the DMA controller Chapter.

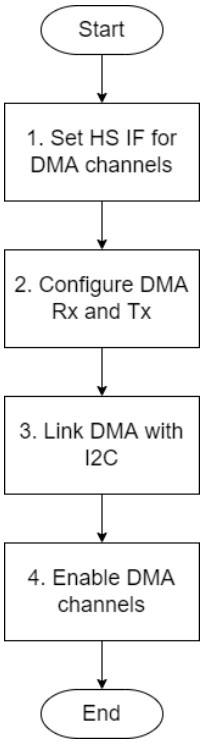


Figure 43 I2C DMA configuration flow

21.5 I2C registers

21.5.1 I2C_CON (I2C control register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x00

Type: Read/Write

Reset value: 0x0000007F

This register can only be written when the I2C is disabled. Writes at other times have no effect.

Bit	Description	Type
<31:10>	Reserved	R/W
<9>	**REFH_CTRL**: Specify whether to hold the bus when the Rx FIFO is full. This bit is read-only.	R/W
<8>	**TXE_CTRL**: Control to generate TX_EMPTY interrupt. 1'b0: Generate TX_EMPTY interrupt if the transmit buffer is at or below the threshold of TX FIFO 1'b1: Generate TX_EMPTY interrupt if the transmit buffer is at or below the threshold of TX FIFO and the internal transfer is completed Reset value: 0	R/W
<7>	**STPD**: This register bit is effective under slave mode only. 1'b0: Generate STOP_DET interrupt when a STOP condition has occurred. 1'b1: Generate STOP_DET interrupt when a STOP condition has occurred and the device has been addressed. When slave received general call, it does not generate STOP_DET. Reset value: 0	R/W
<6>	**SLVDISABLE**: Disable I2C slave. 1'b0: Slave is enabled 1'b1: Slave is disabled Reset value: 1	R/W
<5>	**RSTRT_EN**: Determines whether RESTART conditions may be sent when I2C is a master. 1'b0: Disable 1'b1: Enable Reset value: 1	R/W
<4>	**10ADDR_MASTER**: When I2C is acting as a master, this bit controls whether it starts its transfers in 7- or 10-bit addressing mode. 1'b0: 7-bit addressing 1'b1: 10-bit addressing Reset value: 1	R/W

Bit	Description	Type
<3>	**10ADDR_SLAVE**: When I2C is acting as a slave, this bit controls whether it responds to 7- or 10-bit addresses. 1'b0: 7-bit addressing 1'b1: 10-bit addressing Reset value: 1	R/W
<2:1>	**SPEED[1:0]**: Speed mode 2'b00: Reserved 2'b01: Standard mode (up to 100 Kbps) 2'b10: Fast mode (up to 400 Kbps) or fast mode plus (up to 1 Mbps) 2'b11: High speed mode (up to 3.4 Mbps) Reset value: 0x3	R/W
<0>	**MASTER_MODE**: Enable master mode. 1'b0: Disable 1'b1: Enable Reset value: 1	R/W

21.5.2 I2C_TAR (I2C target address register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x04

Type: Read/Write

Reset value: 0x00000055

This register can only be written when the I2C is disabled. It is not necessary to write to this register while acting as a slave.

Bit	Description	Type
<31:12>	Reserved	R/W
<11>	**SPECIAL**: This bit is set in conjunction with bit[10] to indicates whether the software performs a General Call or START BYTE command. 1'b0: Ignore bit[10] GC_OR_START and use bit [9:0] TAR normally 1'b1: Perform special I2C command as specified in GC_OR_START bit Reset value: 0	R/W
<10>	**GC_OR_STRT**: If bit[11] is set to 1, this bit indicates whether a General Call or START byte command is to be performed. 1'b0: General Call 1'b1: START BYTE Reset value: 0	R/W
<9:0>	**TAR[9:0]**: Target address for any master transaction. Reset value: 0x55	R/W

21.5.3 I2C_SAR (I2C slave address register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x08

Type: Read/Write

Reset value: 0x00000055

This register can only be written when the I2C is disabled. Writes at other times have no effect. It is not necessary to write to this register if the I2C peripheral act as I2C master only.

Bit	Description	Type
<31:10>	Reserved	R/W
<9:0>	**SAR[9:0]**: Slave address when the I2C peripheral is operating as a slave. Reset value: 0x55	R/W

21.5.4 I2C_HS_MADDR (I2C high speed master mode code address register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x0C

Type: Read/Write

Reset value: 0x00000001

This register can only be written when the I2C is disabled. Writes at other times have no effect. It is effective only when acting as a master and operating in high speed mode.

Bit	Description	Type
<31: 3>	Reserved	R/W
<2:0>	**HS_MSTC[2:0]**: HS mode master code. The valid value is in range 0 to7. Reset value: 0x0	R/W

21.5.5 I2C_DATA_CMD (I2C Rx/Tx data buffer and command register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x10

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type

<31:9>	Reserved.	R/W
<8>	**CMD**: Control read or write command. 1'b0: Write 1'b1: Read Note: 1. In master-transmit mode, program this bit to 0 with the data to be transmitted to DAT[7:0]. 2. In master-receive mode, program this bit to 1 will trigger the start condition. For every byte that has been received, program this bit to 1 to send the acknowledge bit to continue the I2C transmission. 3. In slave-transmit mode, program this bit to 0 with the data to be transmitted to DAT[7:0]. 4. In slave-receive mode, program this bit to 1 for every byte received to send the acknowledge bit to continue the I2C transmission. Reset value: 0x0	R/W
<7:0>	**DAT[7:0]**: This register contains the data to be transmitted or received on the I2C bus. Reset value: 0x00	R/W

21.5.6 I2C_SS_SCL_HCNT (Standard Speed Clock SCL High Count Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x14

Type: Read/Write

Reset value: 0x00000190

This register can only be written when the I2C is disabled. Writes at other times have no effect.

It is effective only when acting as a master.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**HCNT[15:0]**: SCL clock high period count for standard speed. The valid value is in between 6 to 65525.	R/W

21.5.7 I2C_SS_SCL_LCNT (Standard Speed Clock SCL Low Count Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x18

Type: Read/Write

Reset value: 0x000001D6

This register can only be written when the I2C is disabled. Writes at other times have no effect.

It is effective only when acting as a master.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**LCNT[15:0]**: SCL clock low period count for standard speed. The minimum value is in 8.	R/W

21.5.8 I2C_FS_SCL_HCNT (Fast or Fast Plus Mode Clock SCL High Count Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x1C

Type: Read/Write

Reset value: 0x0000003C

This register can only be written when the I2C is disabled. Writes at other times have no effect.

It is effective only when acting as a master.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**HCNT[15:0]**: SCL clock high period count for fast mode or fast mode plus. The minimum valid value is 6.	R/W

21.5.9 I2C_FS_SCL_LCNT (Fast or Fast Plus Mode Clock SCL Low Count Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x20

Type: Read/Write

Reset value: 0x00000082

This register can only be written when the I2C is disabled. Writes at other times have no effect.

It is effective only when acting as a master.

Bit	Description	Type
<31:16>	Reserved.	R/W

<15:0>	**LCNT[15:0]**: SCL clock low period count for fast mode or fast mode plus. The minimum valid value is 8.	R/W

21.5.10 I2C_HS_SCL_HCNT (High Speed Clock SCL High Count Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x24

Type: Read/Write

Reset value: 0x00000006

This register can only be written when the I2C is disabled. Writes at other times have no effect.

It is effective only when acting as a master.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**HCNT[15:0]**: SCL clock high period count for high speed. The minimum valid value is 6. Reset value: 0x0006	R/W

21.5.11 I2C_HS_SCL_LCNT (High Speed Clock SCL Low Count Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x28

Type: Read/Write

Reset value: 0x00000010

This register can only be written when the I2C is disabled. Writes at other times have no effect.

It is effective only when acting as a master.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**LCNT[15:0]**: SCL clock low period count for high speed. The minimum valid value is in 8. Reset value: 0x0010	R/W

21.5.12 I2C_INT_STAT (I2C Interrupt Status Register)

Base address:

I2C1 0x4000_2000
I2C2 0x4000_2100

Address offset: 0x2C

Type: Read

Reset value: 0x00000000

This register shows the interrupt status after the I2C interrupt mask (I2C_INT_MASK) has been applied.

Bit	Description	Type
<31:12>	Reserved.	R/W
<11>	**GEN_CALL**: General call This bit is set when a General Call address is received and acknowledged. It is cleared by disabling the I2C module or when bit 0 of the I2C_INT_CLR_GCALL register is read.	R
<10>	**START_DET**: START detected Indicates that a START or Repeated START condition has occurred on the I2C bus.	R
<9>	**STOP_DET**: STOP detected When STPD=0 in I2C_CON, it indicates whether a STOP condition has occurred regardless of whether it is in slave or master mode. When STPD=1 in I2C_CON, it indicates whether a STOP condition has occurred in master mode. When operating in slave mode, it indicates whether a STOP condition has occurred and the device has been addressed.	R
<8>	**ACTIVITY**: I2C activity This bit stays set and captures I2C module activity. It is cleared by either disabling the I2C module, or reading the I2C_INT_CLR_ACT register, or reading the I2C_INT_CLR register.	R
<7>	**RX_DONE**: RX done This bit is set to 1 if the master does not acknowledge a transmitted byte when operating in slave transmit mode.	R
<6>	**TX_ABRT**: TX abort This bit indicates “transmit abort” event and it is set if I2C module is unable to complete the I2C transmission when acting as either master or slave transmitter.	R
<5>	**RD_REQ**: Read data request This bit is set when the I2C module is acting as a slave and another I2C master is requesting to read data. The I2C module holds the I2C clock (SCL=0) until this interrupt is serviced.	R
<4>	**TX_EMPTY**: TX FIFO empty When TXE_CTRL=0 in I2C_CON, this bit is set to 1 when the transmit buffer is at or below the threshold of TX FIFO. When TXE_CTRL=1 in I2C_CON, this bit is set to 1 when the transmit buffer is at or below the threshold of TX FIFO and the internal transfer is completed.	R

Bit	Description	Type
<3>	**TX_OVER**: TX overflow This bit is set when the TX FIFO buffer is overflow.	R
<2>	**RX_FULL**: RX FIFO full This bit is asserted when the receiver buffer reaches or exceeds the RX FIFO threshold level value in the I2C_RX_TL register	R
<1>	**RX_OVER**: RX overflow This bit is set when the receiver buffer is completely filled and an additional byte is received from an external I2C device.	R
<0>	**RX_UNDER**: RX underflow This bit is set when the MCU attempts to read the receive buffer while it is empty.	R

21.5.13 I2C_INT_MASK (I2C Interrupt Mask Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x30

Type: Read

Reset value: 0x000008FF

Bit	Description	Type
<31:12>	Reserved.	R/W
<11>	**GEN_CALL**: General call mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<10>	**START_DET**: START detected mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<9>	**STOP_DET**: STOP detected mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<8>	**ACTIVITY**: I2C activity mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<7>	**RX_DONE**: RX done mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<6>	**TX_ABRT**: TX abort mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W

Bit	Description	Type
<5>	**RD_REQ**: Read data request mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<4>	**TX_EMPTY**: TX FIFO empty mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<3>	**TX_OVER**: TX overflow mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<2>	**RX_FULL**: RX FIFO full mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<1>	**RX_OVER**: RX overflow mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W
<0>	**RX_UNDER**: RX underflow mask 1'b0: Mask interrupt 1'b1: Unmask interrupt	R/W

21.5.14 I2C_INT_RAWSTAT (I2C raw interrupt status register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x34

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:12>	Reserved.	R/W
<11>	**GEN_CALL**: General call This bit is set when a General Call address is received and acknowledged. It is cleared by disabling the I2C module or when bit 0 of the I2C_INT_CLR_GCALL register is read.	R
<10>	**START_DET**: START detected Indicates that a START or Repeated START condition has occurred on the I2C bus.	R
<9>	**STOP_DET**: STOP detected When STPD=0 in I2C_CON, it indicates whether a STOP condition has occurred regardless of whether it is in slave or master mode. When STPD=1 in I2C_CON, it indicates whether a STOP condition has occurred	R

	in master mode. When operating in slave mode, it indicates whether a STOP condition has occurred and the device has been addressed.	
<8>	**ACTIVITY**: I2C activity This bit stays set and captures I2C module activity. It is cleared by either disabling the I2C module, or reading the I2C_INT_CLR_ACT register, or reading the I2C_INT_CLR register.	R
<7>	**RX_DONE**: RX done This bit is set to 1 if the master does not acknowledge a transmitted byte when operating in slave transmit mode.	R
<6>	**TX_ABRT**: TX abort This bit indicates “transmit abort” event and it is set if I2C module is unable to complete the I2C transmission when acting as either master or slave transmitter	R
<5>	**RD_REQ**: Read data request This bit is set when the I2C module is acting as a slave and another I2C master is requesting to read data. The I2C module holds the I2C clock (SCL=0) until this interrupt is serviced.	R
<4>	**TX_EMPTY**: TX FIFO empty When TXE_CTRL=0 in I2C_CON, this bit is set to 1 when the transmit buffer is at or below the threshold of TX FIFO. When TXE_CTRL=1 in I2C_CON, this bit is set to 1 when the transmit buffer is at or below the threshold of TX FIFO and the internal transfer is completed.	R
<3>	**TX_OVER**: TX overflow This bit is set when the TX FIFO buffer is overflow.	R
<2>	**RX_FULL**: RX FIFO full This bit is asserted when the receiver buffer reaches or exceeds the RX FIFO threshold level value in the I2C_RX_TL register	R
<1>	**RX_OVER**: RX overflow This bit is set when the receiver buffer is completely filled and an additional byte is received from an external I2C device.	R
<0>	**RX_UNDER**: RX underflow This bit is set when the MCU attempts to read the receive buffer while it is empty.	R

21.5.15 I2C_RX_TL (I2C receive FIFO threshold register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x38

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type

<31:8>	Reserved.	R/W
<7:0>	**RX_TL[7:0]**: Rx FIFO threshold level. The I2C Rx FIFO depth is 8 and the valid range is 0 to 7 for 1 entry to 8 entries. Any number larger than 7 will set the Rx FIFO threshold level to 7.	R/W

21.5.16 I2C_TX_TL (I2C transmit FIFO threshold register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x3C

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:8>	Reserved, read only is 0	R/W
<7:0>	**TX_TL[7:0]**: Tx FIFO threshold level. The I2C Tx FIFO depth is 8 and the valid range is 0 to 7 for 1 entry to 8 entries. Any number larger than 7 will set the Tx FIFO threshold level to 7.	R/W

21.5.17 I2C_INT_CLR (I2C clear interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x40

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear all interrupts	R

21.5.18 I2C_INT_CLR_RXU (I2C clear RX_UNDER interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x44

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear RX_UNDER interrupt	R

21.5.19 I2C_INT_CLR_RXO (I2C clear RX_OVER interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x48

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear RX_OVER interrupt	R

21.5.20 I2C_INT_CLR_TXO (I2C clear TX_OVER interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x4C

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear TX_OVER interrupt	R

21.5.21 I2C_INT_CLR_RDR (I2C clear RD_REQ interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x50

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W

Bit	Description	Type
<0>	**CLR**: Read to clear RD_REQ interrupt	R

21.5.22 I2C_INT_CLR_TXA (I2C clear TX_ABRT interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x54

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear TX_ABRT interrupt	R

21.5.23 I2C_INT_CLR_RXD (I2C clear RX_DONE interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x58

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear RX_DONE interrupt	R

21.5.24 I2C_INT_CLR_ACT (I2C clear ACTIVITY interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x5C

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear ACTIVITY interrupt	R

21.5.25 I2C_INT_CLR_STOP_DET (I2C clear STOP_DET interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x60

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear STOP_DET interrupt	R

21.5.26 I2C_INT_CLR_START_DET (I2C clear START_DET interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x64

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear START_DET interrupt	R

21.5.27 I2C_INT_CLR_GCALL (I2C clear GEN_CALL interrupt register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x68

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Read to clear GEN_CALL interrupt	R

21.5.28 I2C_ENABLE (I2C enable register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x6C

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:2>	Reserved.	R/W
<1>	**ABORT**: Abort transfer in master mode. It is used in conjunction with bit[0]. 1'b0: ABORT not initiated, or ABORT done 1'b1: ABORT operation in progress	R/W
<0>	**ENABLE**: I2C enable. 1'b0: Disable 1'b1: Enable	R/W

21.5.29 I2C_STATUS (I2C status register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x70

Type: Read

Reset value: 0x00000006

When I2C is disabled:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0
- Bits 5 and 6 are set to 0 (When the master or slave state machines goes to idle)

Bit	Description	Type
<31:7>	Reserved.	R/W
<6>	**SLV_ACTVITY**: When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 1'b0: The Slave component is not Active 1'b1: The Slave component is Active	R
<5>	**MST_ACTVITY**: When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 1'b0: The Master component is not Active 1'b1: The Master component is Active	R
<4>	**RFF**: Receive FIFO Completely Full. 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full	R
<3>	**RFNE**: Receive FIFO Not Empty. 1'b0: Receive FIFO is empty 1'b1: Receive FIFO is not empty	R
<2>	**TFE**: Transmit FIFO Completely Empty. 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty	R
<1>	**TFNF**: Transmit FIFO Not Full. 1'b0: Transmit FIFO is full 1'b1: Transmit FIFO is not full	R
<0>	**ACTIVITY**: I2C Activity Status 1'b0: I2C is inactive 1'b1: I2C is active	R

21.5.30 I2C_TX_FL (I2C transmit FIFO level register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x74

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved.	R/W
<3:0>	**TXFL[3:0]**: Contains the number of valid data entries in the transmit FIFO.	R

21.5.31 I2C_RX_FL (I2C receive FIFO level register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x78

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved.	R/W
<3:0>	**RXFL[3:0]**: Contains the number of valid data entries in the receive FIFO.	R

21.5.32 I2C_SDA_HOLD (I2C SDA hold time length register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x7C

Type: Read/Write

Reset value: 0x00000001

This register can only be written when the I2C is disabled. Writes at other times have no effect.

Bit	Description	Type
<31:24>	Reserved.	R/W
<23:16>	**SDA_RX_HLD[7:0]**: The SDA hold time in units of i2c_clk period, when I2C acts as a receiver.	R/W
<15:0>	**SDA_TX_HLD[15:0]**: The SDA hold time in units of i2c_clk period, when I2C acts as a transmitter.	R/W

21.5.33 I2C_TX_ABRT_SRC (I2C transmit abort source register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x80

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:24>	**TX_FLUSH_CNT[7:0]**: This field preserves the TXFLR value prior to the last TX_ABRT event. It is cleared whenever I2C is disabled. It is valid for master-transmitter or slave-transmitter.	R
<23:17>	Reserved	R/W
<16>	**USER_ABRT:** Master has detected the transfer abort (ENABLE[1]). It is valid for master-transmitter.	R
<15>	**SLVRD_INTX:** This bit is set to 1 when the processor side responds to a slave mode request for data to be transmitted to a remote master and the user writes a 1 in *CMD* (bit[8]) of IC_DATA_CMD register. It is valid for slave-transmitter.	R
<14>	**SLV_ARBLOST:** This bit is set to 1 when the slave lost the bus while transmitting data to a remote master. When this bit is set, bit [12] is set at the same time. It is valid for slave-transmitter	R
<13>	**SLVFLUSH_TXF:** This bit is set when the slave has received a read command and some data exists in the TX FIFO, so the slave issues a *TX_ABRT* interrupt to flush the old data in the TX FIFO. It is valid for slave-transmitter.	R
<12>	**LOST:** This bit is set when the master has lost arbitration, or if bit [14] is also set, then the slave transmitter has lost arbitration. It is valid for master-transmitter or slave-transmitter.	R
<11>	**MASTER_DIS:** This bit is set when the user tries to initiate a Master operation with the Master mode disabled. It is valid for master-transmitter or master-receiver.	R
<10>	**10B_RD_NORSTRT:** This bit is set when the restart is disabled (I2C_CON.Bit[5] = 0) and the master sends a read command in 10-bit addressing mode. It is valid for master-receiver.	R
<9>	**SBYTE_NORSTRT:** This bit is set when the restart is disabled (I2C_CON.Bit[5] = 0) and the user is trying to send a START Byte. It is valid for master.	R
<8>	**HS_NORSTRT:** This bit is set when the restart is disabled (I2C_CON.Bit[5] = 0) and the user is trying to use the master to transfer data in High Speed mode. It is valid for master-transmitter or master-receiver.	R
<7>	**SBYTE_ACTDET:** This bit is set when the master has sent a START Byte and the START Byte was acknowledged (wrong behavior). It is valid for master.	R
<6>	**HS_ACTDET:** This bit is set when the master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). It is valid for master.	R

Bit	Description	Type
<5>	**GCALL_READ:** This bit is set when the I2C is in master mode and sent a General Call but the user programmed the byte following the General Call to be a read from the bus (I2C_DATA_CMD.bit[9] is set to 1). It is valid for master-transmitter.	R
<4>	**GCALL_NOACK:** This bit is set when the I2C is in master mode and sent a General Call but no slave on the bus acknowledged the General Call. It is valid for master-transmitter.	R
<3>	**TXDATA_NOACK:** This bit is set when the master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). It is valid for master-transmitter.	R
<2>	**10ADDR2_NOACK:** This bit is set when the master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. It is valid for master-transmitter or master-receiver.	R
<1>	**10ADDR1_NOACK:** This bit is set when the master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. It is valid for master-transmitter or master-receiver.	R
<0>	**7ADDR_NOACK:** This bit is set when the master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. It is valid for master-transmitter or master-receiver.	R

21.5.34 I2C_DMA_CR (DMA control register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x88

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:2>	Reserved.	R/W
<1>	**TDMAE**: Enable the transmit FIFO DMA channel. 1'b0: transmit DMA disabled 1'b1: transmit DMA enabled	R/W
<0>	**RDMAE**: Enable the receive FIFO DMA channel. 1'b0: receive DMA disabled 1'b1: receive DMA enabled	R/W

21.5.35 I2C_DMA_TDLR (DMA transmit data level register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x8C

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved.	R/W
<2:0>	**DMATDL[3:0]**: Transmit data level. This value controls the transmit data level at which a DMA request is triggered.	R/W

21.5.36 I2C_DMA_RDLR (DMA receive data level register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x90

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved.	R/W
<2:0>	**DMARDL[3:0]**: Receive data level. This value controls the receive data level at which a DMA request is triggered.	R/W

21.5.37 I2C_SDA_SETUP (I2C SDA Setup Register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x94

Type: Read/Write

Reset value: 0x00000064

This register is only used by I2C when operated as a slave transmitter. It can only be written when the I2C is disabled. Writes at other times have no effect.

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:0>	**SDA_STP[7:0]**: SDA setup time. The minimum valid value is 2.	R/W

21.5.38 I2C_ACK_GENERAL_CALL (I2C ACK general call register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0x98
Type: Read/Write
Reset value: 0x00000001
This register is available only when the I2C is in the slave mode.

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**ACK_GCALL**: ACK general call. 1'b0: No general call acknowledged 1'b1: General call acknowledged	R/W

21.5.39 I2C_ENABLE_STATUS (I2C enable status register)

Base address:

I2C1 0x4000_2000
I2C2 0x4000_2100

Address offset: 0x9C
Type: Read
Reset value: 0x00000000
When I2C is enabled, bit[2] and bit[1] are force to 0, bit[0] is forced to 1.
When I2C is disabled, bit[2] and bit[1] are valid as soon as bit[0] is read as 0.

Bit	Description	Type
<31:3>	Reserved.	R/W
<2>	**SLV_FF**: Indicate if a slave-receiver operation has been aborted when I2C is disabled	R
<1>	**SLV_RX_ABRT**: Indicates if a slave operation has been aborted when I2C is disabled	R
<0>	**IC_EN**: I2C enabled status.	R

21.5.40 I2C_FS_SPKLEN (I2C SS and FS spike suppression limit register)

Base address:

I2C1 0x4000_2000
I2C2 0x4000_2100

Address offset: 0xA0
Type: Read/Write
Reset value: 0x00000005
This register can only be written when the I2C is disabled. Writes at other times have no effect.

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**SPKLEN[7:0]**: Set the spike suppression limit for standard mode, fast mode or fast mode plus. The minimum valid value is 1.	R/W

21.5.41 I2C_HS_SPKLEN (I2C HS spike suppression limit register)

Base address:

I2C1 0x4000_2000

I2C2 0x4000_2100

Address offset: 0xA4

Type: Read/Write

Reset value: 0x00000001

This register can only be written when the I2C is disabled. Writes at other times have no effect.

Bit	Description	Type
<31:8>	Reserved.	R/W
7:0	**SPKLEN[7:0]**: Set the spike suppression limit for high speed mode. The minimum valid value is 1.	R/W

21.5.42 I2C register map

The base address of the I2C registers is
I2C1 0x4000_2000;
I2C2 0x4000_2100.
The following table gives the I2C register map and reset values.

Table 52 I2C register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	I2C_CON	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RFFH_CTRL	TXE_CTRL	STPD	SLVDISABLE	RSTRT_EN	10ADDR_MASTER	10ADDR_SLAVE	SPEED[1:0]		MASTER_MODE
	Reset value	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
0x04	I2C_TAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SPECIAL	GC_OR_STRT	TAR[9:0]									
	Reset value	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
0x08	I2C_SAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SAR[9:0]									
	Reset value	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
0x0C	I2C_HS_MADDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		HS_MSTC[2:0]	
	Reset value	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
0x10	I2C_DATA_CMD	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CMD	DAT[7:0]							
	Reset value	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
0x14	I2C_SS_SCL_HCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	HCNT[15:0]															
	Reset value	0	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	1	0	0	0	0
0x18	I2C_SS_SCL_LCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LCNT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0
0x1C	I2C_FS_SCL_HCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	HCNT[15:0]															
	Reset value	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	1	1	1	0	0
0x20	I2C_FS_SCL_LCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LCNT[15:0]															
	Reset value	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	0	0	0	1	0
0x24	I2C_HS_SCL_HCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	HCNT[15:0]															
	Reset value	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	1	1	0
0x28	I2C_HS_SCL_LCNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LCNT[15:0]															

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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	1	0	0	0	0
0x2C	I2C_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GEN_CALL	START_DET	STOP_DET	ACTIVITY	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
	Reset value	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
0x30	I2C_MASK	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GEN_CALL	START_DET	STOP_DET	ACTIVITY	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
	Reset value	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
0x34	I2C_RAW_STAT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GEN_CALL	START_DET	STOP_DET	ACTIVITY	RX_DONE	TX_ABRT	RD_REQ	TX_EMPTY	TX_OVER	RX_FULL	RX_OVER	RX_UNDER
	Reset value	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
0x38	I2C_RX_TL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RX_TL[7:0]							
	Reset value	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
0x3C	I2C_TX_TL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TX_TL[7:0]							
	Reset value	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
0x40	I2C_CLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x44	I2C_CLR_RXU	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x48	I2C_CLR_RXO	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x4C	I2C_CLR_TXO	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x50	I2C_CLR_RD_REQ	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x54	I2C_CLR_TX_ABRT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x58	I2C_CLR_RX_DONE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x5C	I2C_CLR_ACTIVITY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x60	I2C_CLR_STOP_DET	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x64	I2C_CLR_START_DET	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x68	I2C_CLR_GEN_CALL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x6C	I2C_ENABLE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ABORT	ENABLE
	Reset value	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
0x70	I2C_STATUS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SLV_ACTIVITY	MST_ACTIVITY	RFF	RFNE	TFE	TFNF	ACTIVITY
	Reset value	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
0x74	I2C_TXFLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TXFLR[3:0]			
	Reset value	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
0x78	I2C_RXFLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RXFLR[3:0]			
	Reset value	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
0x7C	I2C_SDA_HLD	Res	Res	Res	Res	Res	Res	Res	Res	SDA_RX_HLD[7:0]								SDA_TX_HLD[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x80	I2C_TXA_SRC	TX_FLUSH_CNT[7:0]								Res	Res	Res	Res	Res	Res	Res	USRT_ABRT	SLVRD_INTX	SLV_ARBLOST	SLVFLUSH_TXF	LOST	MASTER_DIS	10B_RD_NORSTRT	SBYTE_NORSTRT	HS_NORSTRT	SBYTE_ACKDET	HS_ACKDET	GCALL_READ	GCALL_NOACK	TXDATA_NOACK	10ADDR2_NOACK	10ADDR1_NOACK	7ADDR_NOACK
	Reset value	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
0x88	I2C_DMA_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDMAE	RDMAE
	Reset value	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
0x8C	I2C_DMA_TDLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMATDL[3:0]			
	Reset value	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
0x90	I2C_DMA_RDLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMARDL[3:0]			
	Reset value	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
0x94	I2C_SDA_STP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SDA_STP[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
0x98	I2C_ACK_GCALL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ACK_GCALL
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x9C	I2C_ENABLE_STATUS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SLV_FF	SLV_RX_ABRT	IC_EN
	Reset value	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
0xA0	I2C_FS_SPKLEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SPKLEN[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0xA4	I2C_HS_SPKLEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SPKLEN[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

22 Universal asynchronous receiver/transmitter (UART)

22.1 Introduction

XT32G0 series provides four UART bus interfaces for asynchronous serial communication with external devices or sensors. UART1 and UART2 support transfer with flow control. UART3 and UART4 provide two-lines data transfer.

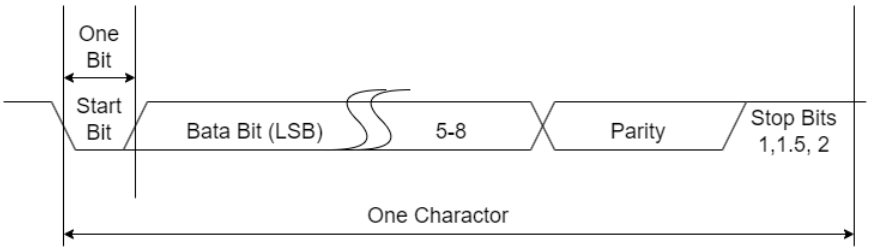


Figure 44 RS232 protocol

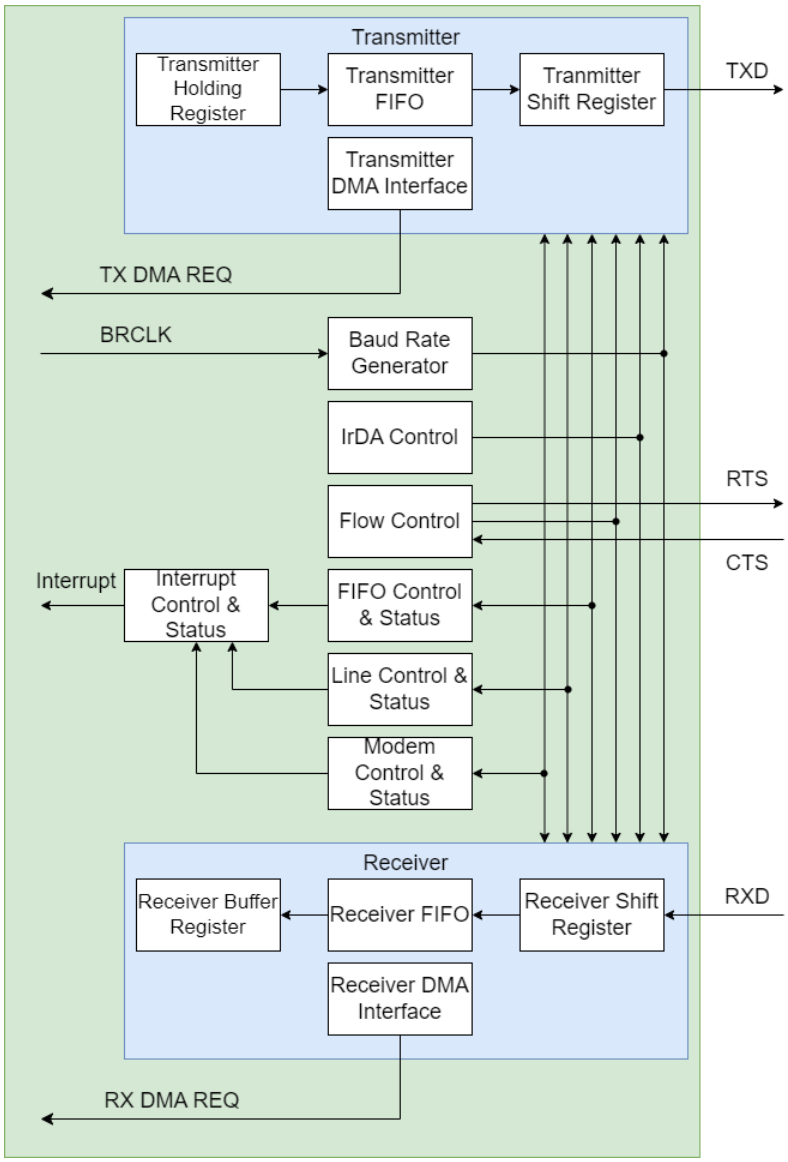


Figure 45 UART block diagram

22.2 UART main features

- 16 bytes FIFO depths
- Support DMA transfer
- Generate interrupts
- IrDA SIR mode with up to 115.2 Kbaud data rate
- Programmable serial data baud rate
- Programmable character of 5-8 data bits
- Parity check
- Optional stop bit with 1/1.5/2 options
- Auto Flow Control mode

22.3 UART signal descriptions

The UART signals are shown in the following table.

Table 53 UART signal descriptions

Signal	Description	I/O
RXD	Receive data	I
TXD	Transmit data	O
CTS	Clear to send (only for UART1 and UART2)	I
RTS	Request to send (only for UART1 and UART2)	O

22.4 UART functional descriptions

22.4.1 Register access

The registers LCR, MCR can be accessed to write when UART is idle state.

22.4.2 Calculate baudrate

$$baudrate = \frac{sclk}{16 \times divisor}$$

Divisor: Its low byte and high bytes are saved in DLL register and DLH register, respectively (see section21.4.3 and section 21.4.5):

$$divisor = (DLH<<8) \mid DLL$$

In XT32H0, sclk = baudrate source clock / divisor (refer to session 4 ICG).

Note: The baudrate clock generator is shared by SPI. We suggest the clock source is the same as system clock (HCLK) and divider is fixed in default value (1).

22.4.3 UART Transmit operation in Polling mode

When UART performs to transmit, the programming sequence show as bellowing steps:

1. Assign IOs to UART port, refer to session 9.3.1.
2. Configure flow control: Set MCR.

   If enable auto flow control, set MCR[5] to 1, else disable auto flow control, set MCR[5] to 0.

   MCR[1] can force RTS to active.

   MCR[0] force DTR signal.
3. Set transfer baud rate.

   Enable Divisor Latch switch to enter baud rate setting mode: Set LCR[7] to 1.

   Set up the Divisor Latch register for required baud rate: write divisor to DLL and DLH, calculate divisor refer to session 22.4.2.

Disable Divisor Latch switch to enter normal mode: clear LCR[7] to 0.

4. Set the transfer parameters, such as data size, parity bit, stop bits: Set LCR.
   Example：
   If configurate 8-bits data length, 1 stop bit and no parity, set LCR[1:0] to 3, LCR[2] to 0, LCR[3] to 0.
5. If use TX FIFO, set TX FIFO threshold (FCR[5:4]) and enable FIFO mode(FCR[0]).
6. Check whether transmission is completed or TX FIFO is empty.
   In non FIFO mode, read THRE status from LSR (LSR[5]), wait to set THRE until timeout.
   Or in FIFO mode, read TEMT status from LSR (LSR[6]), wait to TEMT until timeout.
7. Transmit data: write data to THR register (see 21.4.2).
8. Check whether complete to transmit all data. If more data will be transmitted, back to the step 6 to continue.

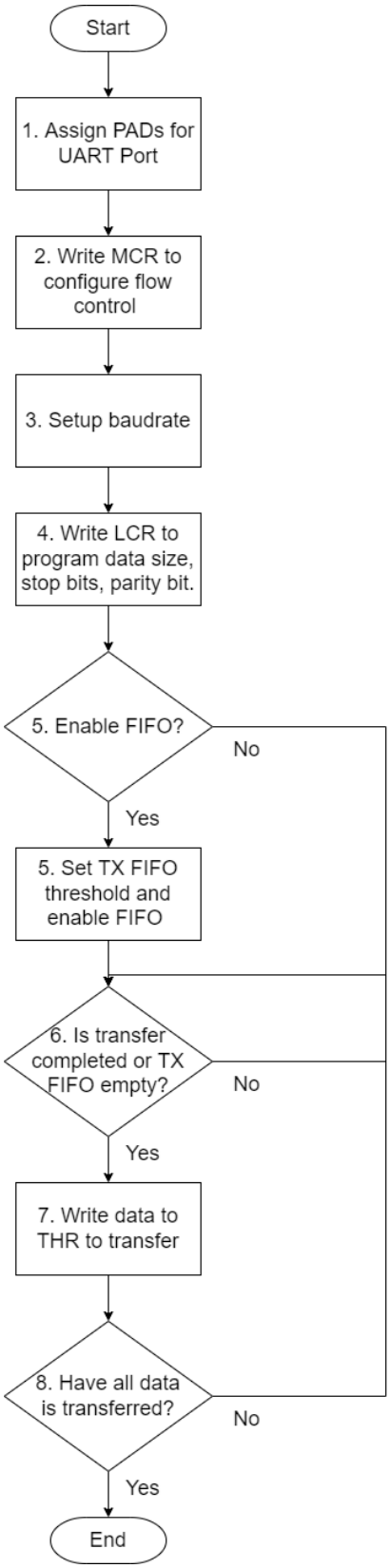


Figure 46 UART transmit in polling mode

22.4.4 UART Transmit operation in Interrupt mode

When UART performs to transmit in Interrupt mode, the programming sequence show as bellowing steps:

1. Assign IOs to UART port.
2. Configure flow control: Set MCR.

Example：

If enable auto flow control, set MCR[5] to 1, else disable auto flow control, set MCR[5] to 0.

MCR[1] can force RTS to active.

MCR[0] force DTR signal.

3. Set transfer baudrate.

Enable Divisor Latch switch to enter baud rate setting mode: Set LCR[7] to1.

Set up the Divisor Latch register for required baud rate: write divisor to DLL and DLH .

Disable Divisor Latch switch to enter normal mode: clear LCR[7] to 0.

4. Set the transfer parameters, such as data size, parity bit, stop bits: Set LCR.

Example：

If configurate 8-bits data length, 1 stop bit and no parity, set LCR[1:0] to 3, LCR[2] to 0, LCR[3] to 0.

5. If use TX FIFO, set TX FIFO threshold (FCR[5:4]) and enable FIFO mode(FCR[0]).
6. Enable enquired interrupt:

If use TX FIFO, Set PTIME (IER[7]) and ETBEI (IER[1]) to 1 (see 21.4.4).

7. When core received interrupt request, check pending interrupt in IIR and enter IRQ handler:

For TX, received THR empty (IIR[3:0]=2) interrupt (see 21.4.6).

8. In IRQ handler, disable interrupt by writing data to THR register (see 21.4.2).
9. Check whether all of data is transmitted. If more data need to transfer, back to step 6 to wait the next THRE interrupt. If no data need to transfer, go to the next step.
10. Disable THRE interrupt, end to transmit.

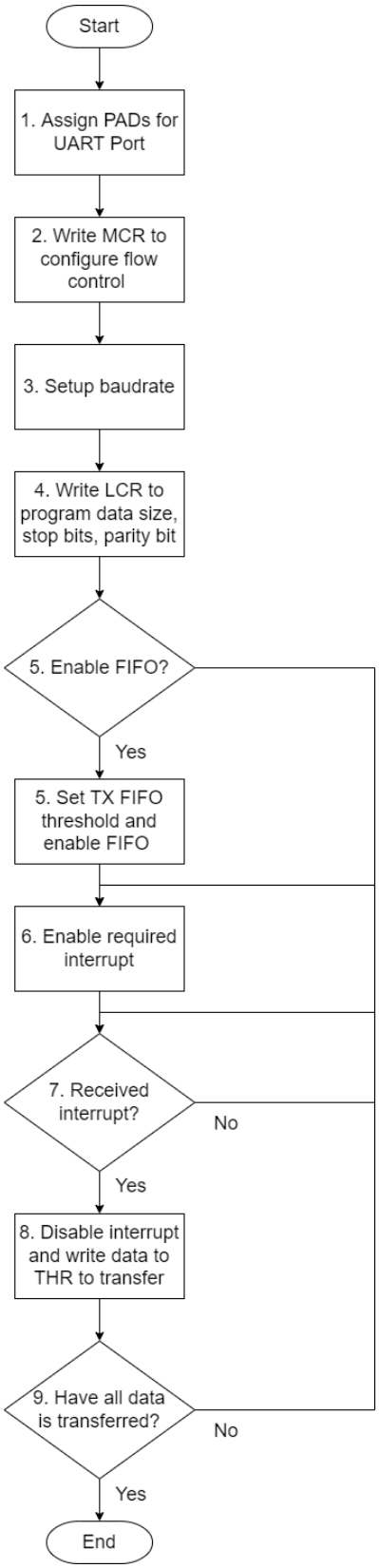


Figure 47 UART transmit in Interrupt mode

22.4.5 UART Transmit operation in DMA mode

When UART performs to transmit in DAM mode, the programming sequence show as bellowing steps:

1. Assign IOs to UART port.
2. Configure flow control: Set MCR.

Example：

If enable auto flow control, set MCR[5] to 1, else disable auto flow control, set MCR[5] to 0.

MCR[1] can force RTS to active.

MCR[0] force DTR signal.

3. Set transfer baud rate.

Enable Divisor Latch switch to enter baud rate setting mode: Set LCR[7] to1.

Set up the Divisor Latch register for required baud rate: write divisor to DLL and DLH .

Disable Divisor Latch switch to enter normal mode: clear LCR[7] to 0.

4. Set the transfer parameters, such as data size, parity bit, stop bits: Set LCR.

Example：

If configurate 8-bits data length, 1 stop bit and no parity, set LCR[1:0] to 3, LCR[2] to 0, LCR[3] to 0.

5. Must use TX FIFO. Set TX FIFO threshold (FCR[5:4]) and enable FIFO mode(FCR[0]).
6. Configurate DMA transfer parameters, includes source address, destination address (THR), channel data length, refer to session 10.
7. Enable required DMA channel interrupt.
8. Link DMA with UART, refer to session 8.
9. Write transmit data by DMA.
10. Waiting until transmission is completed.

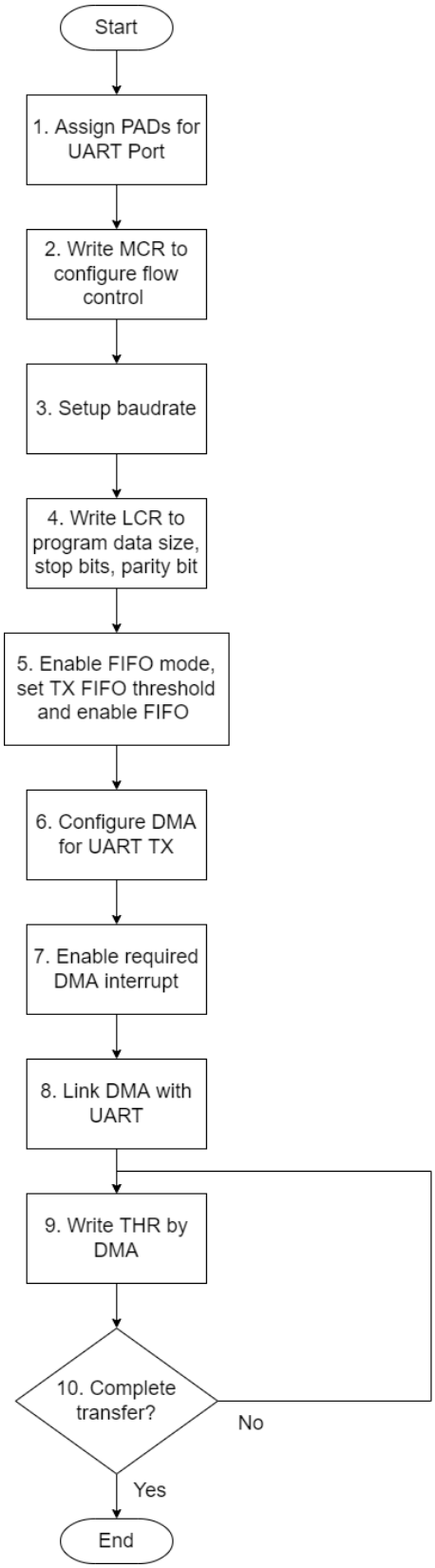


Figure 48 UART transmit in DMA mode

22.4.6 UART Receive operation in Polling mode

When UART performs to receive, the programming sequence show as bellowing steps:

1. Assign IOs to UART port.
2. Configure flow control: Set MCR.

Example：

If enable auto flow control, set MCR[5] to 1, else disable auto flow control, set MCR[5] to 0.

MCR[1] can force RTS to active.

MCR[0] force DTR signal.

3. Set transfer baud rate.

Enable Divisor Latch switch to enter baud rate setting mode: Set LCR[7] to1.

Set up the Divisor Latch register for required baud rate: write divisor to DLL and DLH .

Disable Divisor Latch switch to enter normal mode: clear LCR[7] to 0.

4. Set the transfer parameters, such as data size, parity bit, stop bits: Set LCR.

Example：

If configurate 8-bits data length, 1 stop bit and no parity, set LCR[1:0] to 3, LCR[2] to 0, LCR[3] to 0.

5. If use RX FIFO. Set RX FIFO threshold (FCR[7:6]) and enable FIFO mode(FCR[0]).
6. Check whether data is received or RX FIFO is not empty.

In non FIFO mode or FIFO mode, read DR status from LSR (LSR[0]).

7. Read data from RBR register (see 21.4.2).
8. Check whether complete to receive all data. If more data will be received, back to the step 6 to continue.

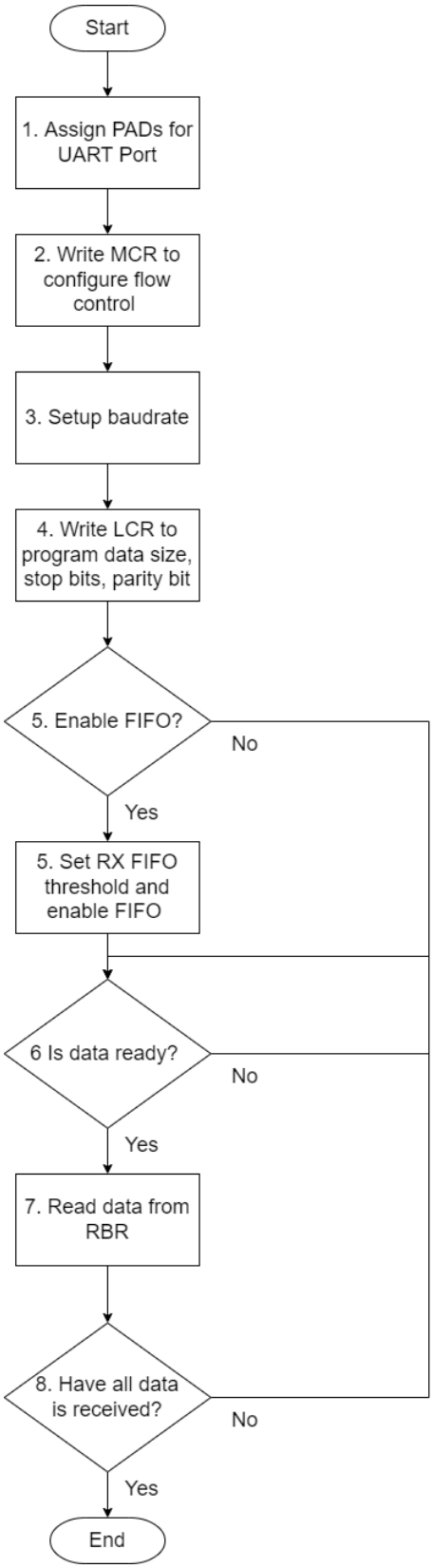


Figure 49 UART Receive operation in Polling mode

22.4.7 UART Receive operation in Interrupt mode

When UART performs to receive in Interrupt mode, the programming sequence show as bellowing steps:

1. Assign IOs to UART port.
2. Configure flow control: Set MCR.

Example：

If enable auto flow control, set MCR[5] to 1, else disable auto flow control, set MCR[5] to 0.

MCR[1] can force RTS to active.

MCR[0] force DTR signal.

3. Set transfer baud rate.

Enable Divisor Latch switch to enter baud rate setting mode: Set LCR[7] to1.

Set up the Divisor Latch register for required baud rate: write divisor to DLL and DLH .

Disable Divisor Latch switch to enter normal mode: clear LCR[7] to 0.

4. Set the transfer parameters, such as data size, parity bit, stop bits: Set LCR.

Example：

If configurate 8-bits data length, 1 stop bit and no parity, set LCR[1:0] to 3, LCR[2] to 0, LCR[3] to 0.

5. If use RX FIFO. Set RX FIFO threshold (FCR[7:6]) and enable FIFO mode(FCR[0]).
6. Enable the required interrupt: for TX, ELSI (IER[2]) and ERBFI (IER[0]) are used (see 21.4.4).
7. When core received interrupt request, check pending interrupt in IIR and enter IRQ handler:

For RX, received data available (IIR[3:0]=4) interrupt (see 21.4.6).

8. In IRQ handler, disable interrupt and read data from RBR register (see 21.4.2).
9. Check whether all of data is received. If more data need to be received, back to step 6 to wait the next required interrupt. If no data need to be received, go to the next step.
10. Disable required interrupt, end to receive.

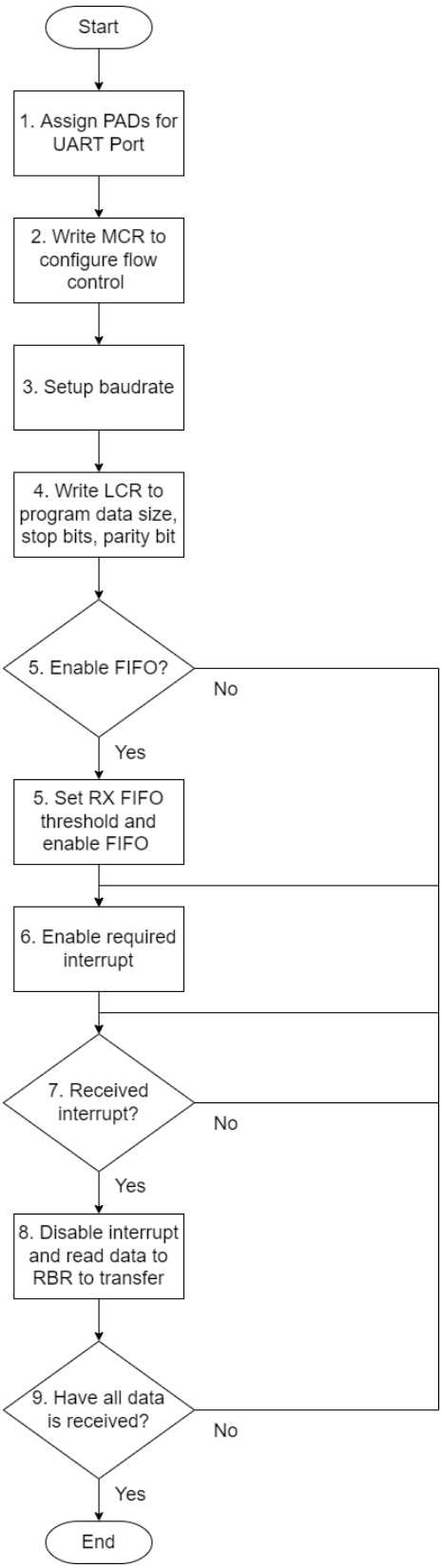


Figure 50 UART Receive operation in Interrupt mode

22.4.8 UART Receive operation in DMA mode

When UART performs to receive in DMA mode, the programming sequence show as bellowing steps:

1. Assign IOs to UART port.
2. Configure flow control: Set MCR.

Example：

If enable auto flow control, set MCR[5] to 1, else disable auto flow control, set MCR[5] to 0.

MCR[1] can force RTS to active.

MCR[0] force DTR signal.

3. Set transfer baud rate.

Enable Divisor Latch switch to enter baud rate setting mode: Set LCR[7] to1.

Set up the Divisor Latch register for required baud rate: write divisor to DLL and DLH .

Disable Divisor Latch switch to enter normal mode: clear LCR[7] to 0.

4. Set the transfer parameters, such as data size, parity bit, stop bits: Set LCR.

Example：

If configurate 8-bits data length, 1 stop bit and no parity, set LCR[1:0] to 3, LCR[2] to 0, LCR[3] to 0.

5. Must use RX FIFO. Set RX FIFO threshold (FCR[7:6]) and enable FIFO mode(FCR[0]).
6. Configurate DMA transfer parameters, includes source address (RBR), destination address, channel data length, refer to session 10.
7. Enable DMA channel interrupt.
8. Link DMA with UART, refer to session 8.
9. Read data from RBR by DMA.
10. Waiting until all data are received.

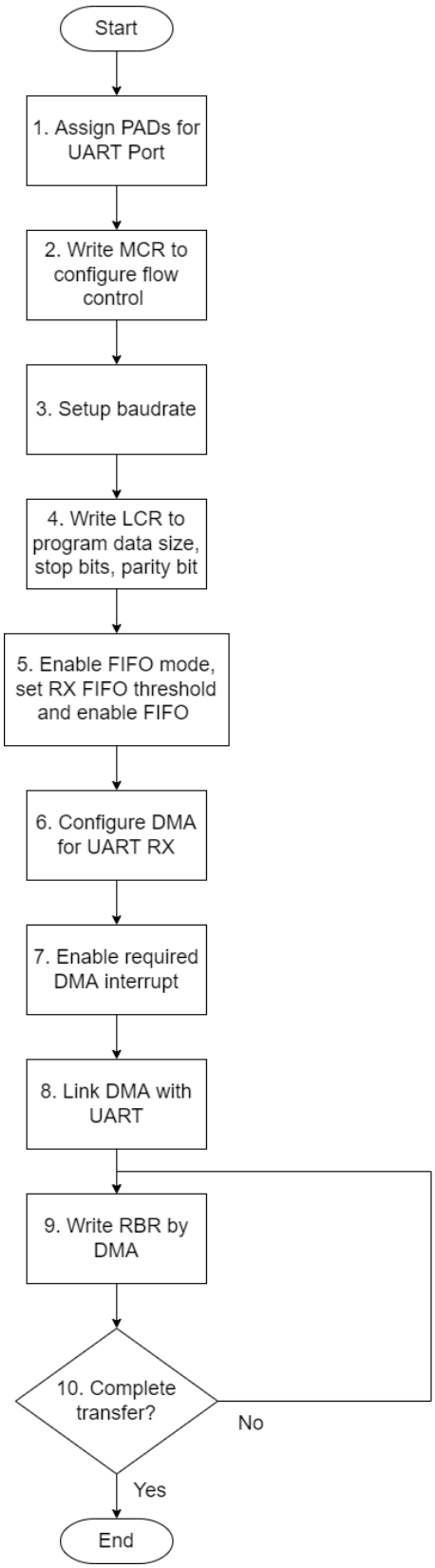


Figure 51 UART Receive operation in DMA mode

22.5 UART registers

22.5.1 UART_RBR (Receive Buffer Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x00
Type: Read
Reset value: 0x00000000
The register can be accessed when LCR[7] is cleared and is valid if the received data is ready (LSR[0] = 1) .

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:0>	**RCV_BUF[7:0]**: Received data buffer	R/W

22.5.2 UART_THR (Transmit Holding Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x00
Type: Write
Reset value: 0x00000000
The register can be accessed when LCR[7] is cleared. When the THR is empty (LSR[5] = 1), the transmitted data should only ne written to the THR.

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:0>	**TRH_BUF[7:0]**: Transmitted data buffer.	R/W

22.5.3 UART_DLL (Divisor Latch Low Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x00

Type: Read/Write

Reset value: 0x00000000

The register can be accessed when LCR[7] =1 and UART is not busy (USR[0] =0).

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:0>	**DIVL[7:0]**: Lower 8 bits of the 16-bit baud rate divisor for UART.	R/W

22.5.4 UART_IER (Interrupt Enable Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x04

Type: Read/Write

Reset value: 0x00000000

This register can be accessed when LCR[7] =1 .

Bit	Description	Type
<31:8>	Reserved.	R/W
<7>	**PTIME**: Enable/Disable THRE Interrupt mode. 1'b0: Disabled 1'b1: Enabled THRE interrupt mode	R/W
<6:4>	Reserved.	R/W
<3>	**EDSSI**: Enable/Disable Modern Status Interrupt. This is the fourth highest priority interrupt. 1'b0: Disabled the modem status register interrupt 1'b1: Enabled the modem status register interrupt	R/W
<2>	**ELSI**: Enable/Disable Receiver Line Status Interrupt. This is the highest priority interrupt. 1'b0: Disabled the receiver line status interrupt 1'b1: Enabled the receiver line status interrupt	R/W
<1>	**ETBEI**: Enable/Disable Transmitter Holding Register Empty interrupt. The transmitter holding register empty is generated at/below threshold (in programmable THRE interrupt mode) This is the third highest priority interrupt. 1'b0: Disabled 1'b1: Enabled	R/W
<0>	**ETBFI**: Enable/Disable Received data Available Interrupt and the Character Timeout Interrupt (if FIFOs enabled).	R/W

Bit	Description	Type
	This is the second highest priority interrupt. 1'b0: Disabled the receiver ready interrupt 1'b1: Enabled the receiver ready interrupt	

22.5.5 UART_DLH (Divisor Latch High Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x04
Type: Read/Write
Reset value: 0x00000000
The register can be accessed when LCR[7] =1 and UART is not busy (USR[0] =0).

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:0>	**DIVH[7:0]**: Upper 8 bits of the 16-bit baud rate divisor for UART.	R/W

22.5.6 UART_IIR (Interrupt Identity Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x08
Type: Read
Reset value: 0x00000000
It can be accessed to read only.

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:6>	**FIFOEN[1:0]**: Indicates whether the FIFOs are enabled or disabled. 2'b00: Disabled 2'b11: Enabled	R/W
<5:4>	Reserved.	R/W
<3:0>	**IID[3:0]**: Indicates the highest priority pending interrupt. 4'b0000: Modem status (Fourth priority level interrupt), clear interrupt by reading MSR 4'b0001: No interrupt pending	R/W

Bit	Description	Type
	4’b0010: THR empty (Third priority level interrupt), clear interrupt by reading IIR or writing THR or TX FIFO is above the threshold. 4’b0100: Receiver data available (Second priority level interrupt), clear interrupt by reading RBR or RX FIFO is below the trigger level. 4’b0110: Receiver line status (Highest priority level interrupt), clear interrupt by reading LSR 4’b0111: Busy detect (Fifth priority level interrupt), clear interrupt by reading USR 4’b1100: Character timeout (Second priority level interrupt), clear interrupt by reading RBR Others: Reserved	

22.5.7 UART_FCR (FIFO Control Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x08
Type: Write
Reset value: 0x00000001

Bit	Description	Type
<31:8>	Reserved.	R/W
<7:6>	**RCVR[1:0]**: Receiver Trigger, select the trigger level in the receiver FIFO at the received Data Available Interrupt (ERBEI) is generated. 2’b00: 1 character data in the RX FIFO 2’b01: 1/4 full in the RX FIFO 2’b10: 1/2 full in the RX FIFO 2’b11: 1 characters less than full in the RX FIFO	R/W
<5:4>	**TET[1:0]**: TX Empty Trigger, select the empty threshold level at the THRE Interrupts are generated. 2’b00: FIFO empty 2’b01: 2 character in the TX FIFO 2’b10: 1/4 full in the TX FIFO 2’b11: 1/2 full in the TX FIFO	R/W
<3>	**DMAM**: DMA mode. 1’b0: Mode 0 (single transfer) 1’b1: Mode 1 (burst transfer)	R/W
<2>	**TFIFOR**: Transmit FIFO reset, it resets the control portion of the transmit FIFO and treats the FIFO as empty. This bit is self-clearing.	R/W

Bit	Description	Type
	1'b0: None 1'b1: Reset	
<1>	**RFIFOR**: Receiver FIFO reset, it resets the control portion of the receive FIFO and treats the FIFO as empty. This bit is self-clearing. 1'b0: None 1'b1: Reset	R/W
<0>	**FIFOE**: Enable/disable the transmit (TX) and receive (RX) FIFOs. 1'b0: Disabled FIFOs 1'b1: Enabled FIFOs	R/W

22.5.8 UART_LCR (Line Control Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x0C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:8>	Reserved.	R/W
<7>	**DLAB**: Enable/Disable the Divisor Latch Register (DLL and DLH. It is writable only when UART is idle (USR[0 =0], and must be cleared after initial baud rate setup. 1'b0: Disable, normal operation 1'b1: Enable to read and write DLL and DLH	R/W
<6>	**BC**: Set a break condition to be transmitted to the receiving device. 1'b0: No break 1'b1: the serial output is forced to logic 0 state. if UART is worked in SIR mode, SIR output line is continuously pulsed	R/W
<5>	**SP**: Enable Stick Parity. This bit is used to force parity value. When PEN, EPS, and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled. 1'b0: Disable Stick Parity 1'b1: Stick Parity	R/W
<4>	**EPS**: Set Even or Odd Parity. This is used to select between even and odd parity, when parity is enabled (PEN set to 1). If set to 1, an even number of logic 1s is transmitted or checked. If set to 0, an odd number of logic 1s is transmitted or	R/W

Bit	Description	Type
	checked. 1'b0: Odd Parity 1'b1: Even Parity	
<3>	**PEN**: Select the parity generation and detection in transmitted and received character respectively. 1'b0: Parity disabled 1'b1: Parity enabled	R/W
<2>	**STOP**: Set the number of stop bits per character that the peripheral transmits and receives. 1'b0: 1 stop bit 1'b1: 1.5 stop bits if dat alength is 5 bits, otherwise 2 stop bits	R/W
<1:0>	**DLS[1:0]**: Set the number of data bits per character that the peripheral transmits and receives. 2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits	R/W

22.5.9 UART_MCR (Modem Control Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x10
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:7>	Reserved.	R/W
<6>	**SIRE**: Enable/Disable the IrDA SIR Mode features. 1'b0: IrDA SIR mode disabled 1'b1: IrDA SIR mode enabled.	R/W
<5>	**AFCE**: Enable/Disable Auto Flow Control. 1'b0: Auto Flow Control mode disabled 1'b1: Auto Flow Control mode enabled	R/W
<4>	**LB**: Put the UART into a diagnostic mode for test purposes. 1'b0: normal operation 1'b1: Enable loopback mode	R/W
<3:2>	Reserved	R/W

<1>	**RTS**: Control the Request to send (RTS) output to inform the modem or data set that the UART is ready to exchange data. 1'b0: RTS de-asserted, RTS output high 1'b1: RTS asserted, RTS output low	R/W
<0>	**DTR**: Control the Data Terminal Ready (DTR) output to inform the modem or data set that the UART is ready to establish a communication. 1'b0: DTR de-asserted, DTR output high 1'b1: DTR asserted, DTR output low	R/W

22.5.10 UART_LSR (Line Status Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x14
Type: Read
Reset value: 0x00000060
LSR, Line status register provide the status of the transmit hold register.

Bit	Description	Type
<31:8>	Reserved.	R/W
<7>	**RFE**: Indicates when there are any errors in the receiver FIFO, such as at least one parity error, framing error, or break detection. This bit is valid if FCR[0]=1 (FIFOs are enabled). 1'b0: No error in RX FIFO 1'b1: Error detected in RX FIFO Reset value: 0x0	R/W
<6>	**TEMT**: Indicates transmitter is empty. If in FIFO mode, it is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode, it is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty. 1'b0: transmitter not Empty 1'b1: transmitter Empty Reset value: 0x1	R/W
<5>	**THRE**: Indicates the THR or TX FIFOs empty if THRE mode disable(IER[7] =0). Or indicates that the TX FIFOs is FULL related with FCR[5:4] if both IER[7] =1 and FCR[0] =1. 1'b0: THR not empty 1'b1: THR empty(THRE) Reset value: 0x1	R/W

Bit	Description	Type
<4>	**BI**: Indicates the detection of a break in the receiver. 1'b0: No Break 1'b1: Break detected Reset value: 0x0	R/W
<3>	**FE**: Indicate the occurrence of a framing error in the receiver. 1'b0: No framing error 1'b1: Framing error Reset value: 0x0	R/W
<2>	**PE**: Indicates the occurrence of a parity error in the receiver. 1'b0: No parity error 1'b1: Parity error Reset value: 0x0	R/W
<1>	**OE**: Indicates the occurrence of an overrun error. 1'b00: No overrun error 1'b01: overrun error Reset value: 0x0	R/W
<0>	**DR**: Indicates that the receiver contains at least one character in the RBR or the receiver FIFO. 1'b00: No data ready 1'b01: Data ready Reset value: 0x0	R/W

22.5.11 UART_MSR (Modem Status Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00
UART4 0x4000_1F00

Address offset: 0x18
Type: Read
Reset value: 0x00000040

Bit	Description	Type
<31:8>	Reserved.	R/W
<7>	**DCD**: Indicates the current DCD(Data Carrier Detect) state that the data carrier has been detected by the modem or data set. 1'b0: No data carrier 1'b1: Carrier has been detected Reset value: 0x0	R/W
<6>	**RI**: Indicates the current RI (Ring Indicator) state that a telephone ringing signal	R/W

Bit	Description	Type
	has been received by modem or data set. 1'b0: No ring signal 1'b1: A telephone ringing signal has been received Reset value: 0x0	
<5>	**DSR**: Indicates the current DSR (Data Set Ready) state that the modem or data set is ready to establish the communications link with the UART. 1'b0: Not ready 1'b1: Ready to communicate with UART Reset value: 0x0	R/W
<4>	**CTS**: Indicates the current CTS (Clear to Send) state that the modem or data set is ready to exchange data with UART. 1'b0: Not ready 1'b1: Ready to exchange data with UART Reset value: 0x0	R/W
<3>	**DDCR**: Indicate that the DCD state has been changed since the last time the MSR was read. 1'b0: No change on DCD state 1'b1: DCD state has been changed Reset value: 0x0	R/W
<2>	**TERI**: Indicates that RI state has been changed since the last time the MSR was read. 1'b0: No change on RI state 1'b1: RI changed state Reset value: 0x0	R/W
<1>	**DDSR**: Indicates that DSR state has been changed since the last time the MSR was read. 1'b0: No change on DSR state 1'b1: DSR changed state Reset value: 0x0	R/W
<0>	**DCTS**: Indicates that CTS state has been changed since the last time the MSR was read. 1'b0: No change on CTS state 1'b1: CTS changed state Reset value: 0x0	R/W

22.5.12 UART_USR (UART Status Register)

Base address:

UART1 0x4000_1C00
UART2 0x4000_1D00
UART3 0x4000_1E00

UART4 0x4000_1F00

Address offset: 0x7C

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**BUSY**: Indicates that a serial transfer is in progress. 1'b0: UART is idle or inactive 1'b1: UART is busy Reset value: 0x0	R/W

22.5.13 UART register map

The base address of the UART registers is

UART1 0x4000_1C00;

UART2 0x4000_1D00;

UART3 0x4000_1E00;

UART4 0x4000_1F00.

The following table gives the UART register map and reset values.

Table 54 UART register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	UART_RBR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RCV_BUF[7:0]							
	Reset value	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
0x00	UART_THR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TRH_BUF[7:0]							
	Reset value	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
0x00	UART_DLL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DIVL[7:0]							
	Reset value	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
0x04	UART_DLH	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DIVH[7:0]							
	Reset value	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
0x04	UART_IER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PTIME	Res	Res	Res	EDSSI	ELSI	ETBEI	ERBFI
	Reset value	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
0x08	UART_IIR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FIFOSE[1:0]		Res	Res	IID[3:0]			
	Reset value	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	1
0x08	UART_FCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RCVR[1:0]		TET[1:0]		DMAM	TFIFOR	RFIFOR	FIFOE
	Reset value	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
0x0C	UART_LCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DLAB	BC	SP	EPS	PEN	STOP	DLS[1:0]	
	Reset value	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
0x10	UART_MCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SIRE	AFCE	LB	Res	Res	RTS	DTR
	Reset value	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
0x14	UART_LSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RFE	TEMT	THRE	BI	FE	PE	OE	DR
	Reset value	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	1	0	0	0	0	0
0x18	UART_MSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
	Reset value	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	0	0	0	0
0x7C	UART_USR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BUSY
	Reset value	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	1	1	0

23 Serial peripheral interface (SPI)

23.1 Introduction

XT32H05x provides two SPI bus interfaces for full duplex serial communication with external devices. SPI mode is selectable by software. Both SPI1 and SPI2 supports Master mode or Slave mode.

The SPI provides synchronous, serial communication between the MCU and external devices. Application software can manage the communication by polling the status flag or using dedicated SPI interrupt. A typical SPI transmission signal is shown below:

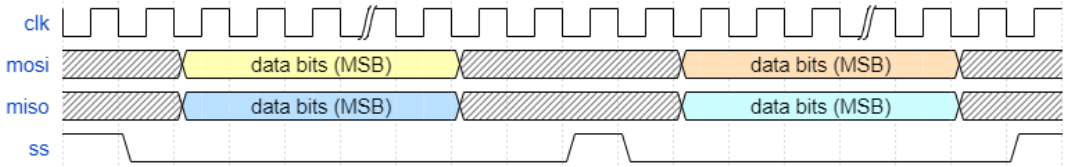


Figure 52 SPI signals

23.2 SPI main features

- Support 4 to 16 bits serial data transfer
- 16 bits width and 8 entries depth of transmit/receive FIFO buffer
- Support DMA transfer
- Generate interrupts
- Independent masking of interrupts
- Multi-master contention detection
- Programmable serial clock polarity and phase
- Choice of Motorola SPI and Texas Instrument Synchronous Serial Protocol

23.3 SPI signal descriptions

The SPI signals are shown in the following table.

Table 55 SPI signal descriptions

Signal	Description	I/O
SCK	Bidirectional serial clock	I/O
RXD	Receive data (MISO for master, MOSI for slave)	I
TXD	Transmit data (MOSI for master, MISO for slave)	O
SS	Bidirectional slave select	I/O

23.4 SPI functional description

23.4.1 Register access

Users are advised to observe the following rules regarding SPI register access.

1. Register SPI_CTRLR1, SPI_BAUDR, SPI_SER are valid when SPI is configured as master.
2. Register SPI_CTRLR0, SPI_CTRLR1, SPI_MWCR, SPI_BAUDR, and SPI_SER becomes writable when SPI is disabled.
3. Register SPI_DR becomes writable when SPI is enabled.

23.4.2 Clock Ratios

When SPI1 is configured as master, the device will generate the SCLK-serial clock. The frequency of SCLK can be calculated from the following equation:

$$f_{sclk} = \frac{f_{brclk}}{bauddivisor * sckdivisor}$$

, where $sckdivisor$ is the value of the BAUD register (refer to section 22.4.6)

$bauddivisor$ is the value of BAUDRATE_CLK_DIV register of ICG module (refer to section 4.4.5)

the recommended frequency ratio restriction is:

$$Master: \frac{f_{brclk}}{bauddivisor} \geq 2 \times (maximum\ f_{sclk})$$

$$Salve: \frac{f_{brclk}}{bauddivisor} \geq 10 \times (maximum\ f_{sclk})$$

23.4.3 Transmit and Receive FIFO buffers

XT32H05x has a built-in 8-entries (16-bits width) depth of transmit (TX) FIFO buffers and a 8-entries (16-bits width) depth of receive (RX) FIFO buffers.

The TX FIFO can be loaded by writing the DR register.

- TX FIFO empty interrupt (TXE):
  If the number of entries in the TX FIFO is less than or equal to the TX FIFO threshold value (see 22.4.7 TXFIFO_TLR register), the TX FIFO will generate a TX FIFO empty interrupt.
  When the software configures the DMA transfer mode, the transmit data level (see 22.4.21 DMA_TDLR) can be used to trigger the TX FIFO empty interrupt.
- TX FIFO overflow interrupt (TXO):
  If the number of entries in the TX FIFO is already full (=8 entries) and the user attempt to write more data, the TX FIFO will generate a TX FIFO overflow interrupt.
  The RX FIFO is loaded from the SPIx RXD line by the internal receive shift control logic, the received data are loaded from RX FIFO by reading the DR register.
- RX FIFO full interrupt (RXF):
  If the number of entries in the RX FIFO is greater than or equal to the RX FIFO threshold value plus 1 (see 22.4.8 RXFIFO_TLR register), the RX FIFO will generate a RX FIFO full interrupt.

When the software configures the DMA transfer mode, the receive data level (see 22.4.22 DMA_RDLR) can be used to trigger the RX FIFO full interrupt.

- RX FIFO overrun interrupt (RXO):
  If the RX FIFO is already full (=8 entries) and the data continues to fill the receive buffer, the RX FIFO will generate a RX FIFO overrun interrupt and the newly received data will be lost.
- RX FIFO underflow interrupt (RXU):
  If the user attempt to read from an empty RX FIFO, the RX FIFO will generate a RX FIFO underflow interrupt request to notify that the data read are invalid.

23.4.4 Transfer Modes

SPI can be configured for three different transfer modes by setting the register bits, SPI_CTRLR0[9:8]. The SPI transfer mode setting is shown as below:

- 2’b00: Transmit & Receive
- 2’b01: Transmit only
- 2’b10: Receive only

23.4.5 Master transfer programming in polling mode

The steps below show the programming sequence for SPI master transfer using polling.

1. Assign the IO as SPI ports:
   MOSI, MISO, SS_N and SCLK.
   Set SPI module to master mode in system config refer to session.
   Set SPI clock source and clock divider refer to session.
2. Initialize the SPI master:
   a. Disable the SPI: Set SPI_SSIEN.EN[0]=0.
   b. Configure the control transfer parameters: Write SPI_CTRLR0.
   Example:
   Write SPI_CTRLR0[9:8]=2`b00 to select transmit and receive mode
   Write SPI_CTRLR0[7] to set the SCLK polarity, SPI_CTRLR0[6] to set the SCLK phase
   Write SPI_CTRLR0[5:4]=2`b00 to select the SPI protocol frame format
   Write SPI_CTRLR0[3:0]=4`b0111 to set the 8-bit data frame length
   c. If it is a received only transfer, set the number of data frames expected for the transfer:
   Write SPI_CTRLR1 with the expected number of data frames minus 1
   Example:
   If SPI_CTRLR0[9:8] is set to 2`b10 and the user expects to receive 16 data frames continuously, then the user should write SPI_CTRLR1=0x0F (15).
   d. Set up the transfer baud rate:
   Write SPI_BAUDR (refer to 22.3.2).
   e. Set the TX/RX FIFO threshold levels: write SPI_TXFTLR, SPI_RXFTLR.

Example:

Write SPI_TXFTLR=0x02 to set the transmit FIFO threshold to 2 data entries. This means that the TXE interrupt would be asserted when there are less than or equal to 2 data entries remaining in the TX FIFO.

Write SPI_RXFTLR=0x03 to set the receive FIFO threshold to 3 data entries. This means that the RXF interrupt would be asserted when there are greater than or equal to 3 data entries in the RX FIFO.

f. Set interrupt masks: write SPI_IMR.

Example:

Write SPI_IMR= 0x3F to unmask all interrupt.

g. Enable the slave selection: write SPI_SER[0] =1.

h. Enable the SPI master: write SPI_SSIEN.EN[0] =1.

3. Check to ensure that the transmit FIFO is not full: read SPI_SR[1] should return 1.
4. If the transmit FIFO is not full, write the data to be transmitted to the data register: write SPI_DR.
5. Check to see if the receive FIFO is empty or not:
   Read SPI_SR[3] should return 1 if the receive FIFO is not empty and return 0 if the receive FIFO is empty.
6. If the receive FIFO is not empty, read the data received from the target slave device:
   Read SPI_DR.
7. Check and see if there is any remaining data to be transferred. If yes, loop back to the step 3 to continue.
8. Keep polling the status bit to wait for the completion of the transfer:
   Read SPI_SR[0]=0, or SPI_SR[2]=1, or SPI_SR[3]=1
9. Transfer completed. Write SPI_SSIEN.EN[0]=0 to disable the SPI master.

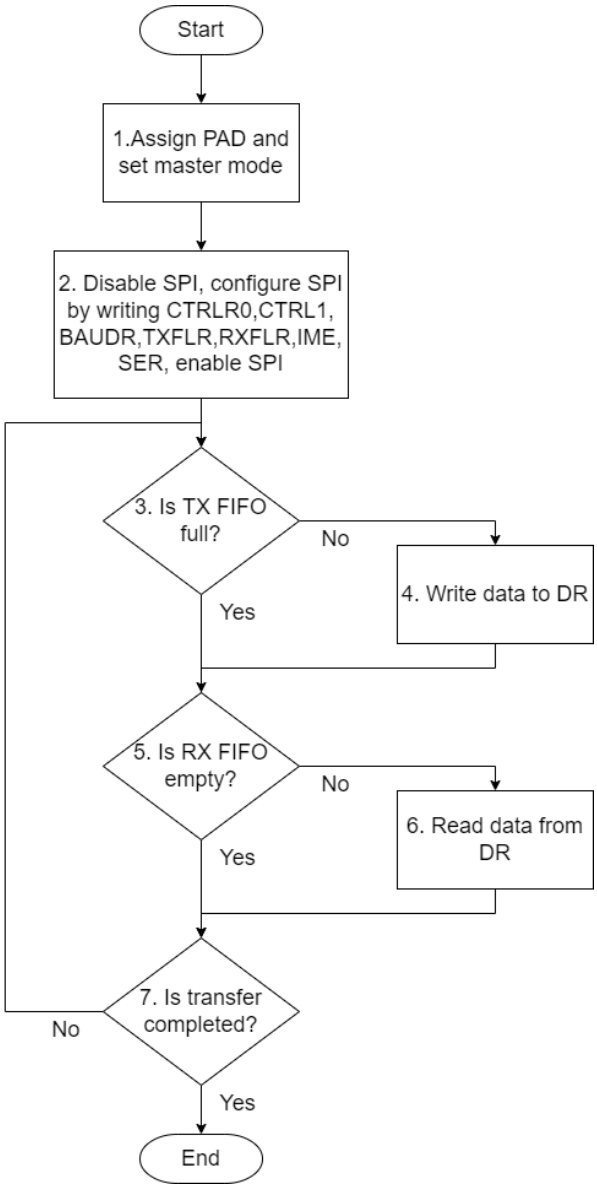


Figure 53 SPI Master transmit & receive in polling mode

23.4.6 Master transfer programming in Interrupt mode

The steps below show the programming sequence to perform SPI transmit and receive using interrupt.

1. Assign IO as SPI ports:
   MOSI, MISO, SS_N and SCLK.
   Set SPI module to master mode in system config refer to session.
   Set SPI clock source and clock divider refer to session.
2. Initialize the SPI master:
   a. Disable the SPI: write SPI_SSIEN.EN[0]=0.
   b. Configure the control transfer parameters: Write SPI_CTRLR0.
   c. If the transfer is received only, set the number of data frames in the transfer:

Write SPI_CTRLR1 with the expected number of data frames minus 1.

d. Set up the transfer baud rate:
   Write SPI_BAUDR (refer to 22.3.2)
e. Set the TX/RX FIFO threshold levels: write SPI_TXFTLR and SPI_RXFTLR
f. Unmask interrupt: write SPI_IMR.
g. Enable the target slave selection: write SPI_SER[0] =1.
h. Enable the SPI master: write SPI_SSIEN.EN[0] =1.

3. Set priority of SPI IRQ and enable SPI IRQ: see ICTL in section 11.4
4. Wait for the interrupt trigger and check the FIFO status.
   a. If TXE(SPI_RISR[0]) is triggered and the transmit FIFO is not full (SPI_SR[1]=1), then write data to TX FIFO: write SPI_DR
   b. If RXF(SPI_RISR[4]) is triggered and the receive FIFO is not empty (SPI_SR[3]=1), then read data from Rx FIFO: read SPI_DR.
5. Check and see if all the data have been transferred. If not, loop back to the step 4 to continue.
   a. During SPI transmit, check and see whether all the data have been transmitted.
   b. During SPI receive, check and see whether all the data have been received.
6. Transfer completed. Check the SPI busy status.
   Read SPI_SR[0]=0, or SPI_SR[2]=1, or SPI_SR[3]=1
7. Disable the interrupt. Write SPI_SSIEN.EN[0]=0 to disable the SPI master.

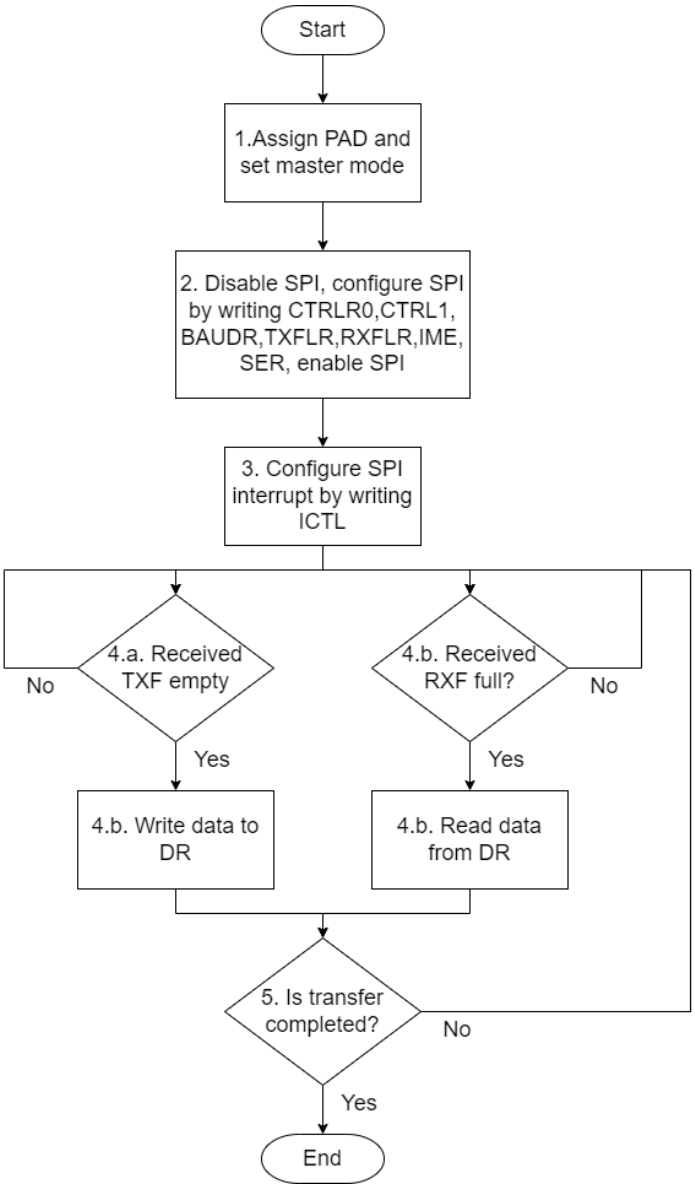


Figure 54 SPI Master transmit & receive in interrupt mode

23.4.7 Master transfer programming in DMA mode

The steps below show the programming sequence for SPI master transfer with DMA.

i. Configure IO as SPI ports:
   MOSI, MISO, SS_N and SCLK.
   Set SPI module to master mode in system config refer to session.
   Set SPI clock source and clock divider refer to session.

ii. Initialize the SPI master:
   a. Disable the SPI: write SPI_SSIEN.EN[0]=0.
   b. Configure the control transfer parameters: Write SPI_CTRLR0.
   c. If the transfer is received only, set the number of data frames for the transfer:

Write SPI_CTRLR1 with the desired number of data frames minus 1.

d. Set up the transfer baud rate: Write SPI_BAUDR (refer to 22.3.2)

e. Set TX/RX FIFO threshold levels:
Write SPI_TXFTLR and SPI_RXFTLR

f. Unmask interrupt: write SPI_IMR.

g. Enable the target slave selection: write SPI_SER =1.

h. Enable the SPI master: write SPI_SSIEN.EN[0] =1.

iii. Configure the DMA RX/TX channels

a) Configure the appropriate DMA handshaking interface by writing DMA_CFG and DMA_CFG_2 in SYSCFG

b) For SPI RX, configure the receive channel parameters including source address, destination address, channel data length and handshaking interface. (Refer to Chapter 9 DMA controller for details)

- Source address: SPI_DR register
- Destination address: Receive data buffer

c) For SPI TX, configure the transmit channel parameters including source address, destination address, channel data length and handshaking interface.

- Source address: Transmit data buffer
- Destination address: SPI_DR register

d) Enable DMA receive channel interrupt.

e) Enable DMA transmit channel interrupt.

iv. Associate the DMA channel to SPI.

v. Perform data transfer by DMA.

vi. Check whether data transfer is completed. If more data need to be received and transmitted, then wait for the DMA transfer.

vii. SPI is in idle state. Disable interrupt, write SPI_SSIEN.EN[0]=0 to disable the SPI master.

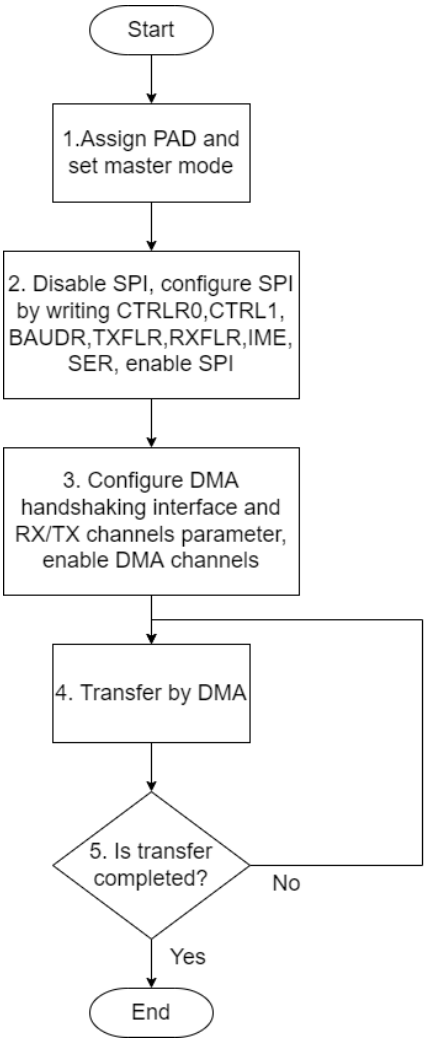


Figure 55 SPI Master transmit & receive in DMA mode

23.4.8 Slave transfer programming in polling mode

The steps below show the programming sequence for data transmit / receive operations for SPI slave using polling.

1. Assign IO as SPI ports:
   MOSI, MISO, SS_N and SCLK.
   Set SPI module to slave mode in system config refer to session.
   Set SPI clock source and clock divider refer to session.
2. Initialize the SPI slave:
   a. Disable the SPI: write SPI_SSIEN.EN[0]=0.
   b. Configure the control transfer parameters: Write SPI_CTRLR0.

Example:
Write SPI_CTRLR0[9:8]=2`b00 to select transmit and receive mode,
Write SPI_CTRLR0[7] to set SCLK polarity, write SPI_CTRLR0[6] to set SCLK phase,
Write SPI_CTRLR0[5:4]=2`b00 to set SPI protocol frame format,
Write SPI_CTRLR0[3:0]=4`b0111 to set 8-bit data frame length.

c. If it is a received only transfer, set the number of data frames expected for the transfer:
Write SPI_CTRLR1 with the expected number of data frames minus 1.
Example:
If SPI_CTRLR0[9:8] is set to 2`b10 and the user expects to receive 16 data frames continuously, then the user should write SPI_CTRLR1=0x0F (15).

d. Set the TX/RX FIFO threshold levels: write SPI_TXFTLR, SPI_RXFTLR.
Example:
Write SPI_TXFTLR=0x02 to set the transmit FIFO threshold to 2 data entries. This means that the TXE interrupt would be asserted when there are less than or equal to 2 data entries remaining in the TX FIFO.
Write SPI_RXFTLR=0x03 to set the receive FIFO threshold to 3 data entries. This means that the RXF interrupt would be asserted when there are greater than or equal to 3 data entries in the RX FIFO.

e. Set interrupt masks: write SPI_IMR.
Example:
Write SPI_IMR= 0x3F to unmask all interrupt

f. Enable the SPI slave: write SPI_SSIEN.EN[0]=1.

3. Write data to SPI TX FIFO.
4. Check to ensure that the transmit FIFO is not full: read SPI_SR[1] should return 1.
5. If the transmit FIFO is not full, write the data to be transmitted to the data register: write SPI_DR.
6. Check to see if the receive FIFO is empty or not:
   Read SPI_SR[3] should return 1 if the receive FIFO is not empty and return 0 if the receive FIFO is empty.
7. If the receive FIFO is not empty, read the data received from the target slave device:
   Read SPI_DR.
8. Check and see if more data need to be transmitted. If yes, loop back to step 4 to continue.
9. Keep polling the status bit to wait for completion of the transfer:
   Read SPI_SR[0]=0, or SPI_SR[2]=1, or SPI_SR[3]=1.
10. Transfer completed. Write SPI_SSIEN.EN[0]=0 to disable the SPI slave.

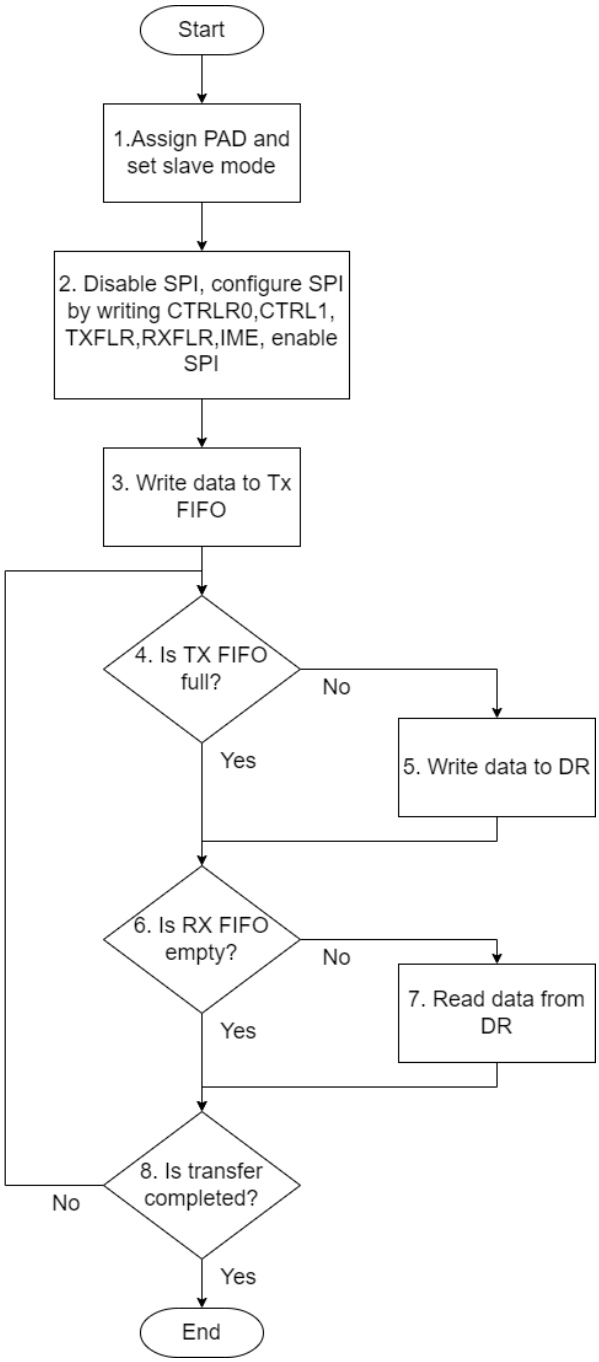


Figure 56 SPI Slave transmit & receive in polling mode

23.4.9 Slave transfer programming in Interrupt mode

The steps below show the programming sequence to perform SPI slave transmit and receive using interrupt.

1. Assign IO as SPI ports:
   MOSI, MISO, SS_N and SCLK.
   Set SPI module to slave mode in system config refer to session.
   Set SPI clock source and clock divider refer to session.
2. Initialize the SPI slave:
   a. Disable the SPI: write SPI_SSIEN.EN[0]=0.
   b. Configure the control transfer parameters: Write SPI_CTRLR0.
   c. If the transfer is received only, set the number of data frames in the transfer: Write SPI_CTRLR1 with the expected number of data frames minus 1.
   d. Set the TX/RX FIFO threshold levels: write SPI_TXFTLR and SPI_RXFTLR
   e. Unmask interrupt: write SPI_IMR.
   f. Enable the SPI slave: write SPI_SSIEN.EN[0]=1.
3. Set priority of SPI IRQ and enable SPI IRQ: see ICTL in section 11.4
4. Write data to TX FIFO: write SPI_DR
5. Wait for the interrupt triggered and check the FIFO status.
   a. If TXE(SPI_RISR[0]) is triggered and the transmit FIFO is not full (SPI_SR[1=1]), then write data to TX FIFO: write SPI_DR
   b. If RXF(SPI_RISR[4]) is triggering and receive FIFO is not empty (SPI_SR[3]=1), then read data from Rx FIFO: read SPI_DR.
6. Check and see if all data have been transferred. If not, loop back to step 4 to continue.
   a. During SPI transmit, check and see whether all the data have been transmitted.
   b. During SPI receive, check and see whether all the data have been received.
7. Transfer completed. Check the SPI busy status.
   Read SPI_SR[0]=0, or SPI_SR[2]=1, or SPI_SR[3]=1
8. Disable the interrupt. Write SPI_SSIEN.EN[0]=0 to disable the SPI slave.

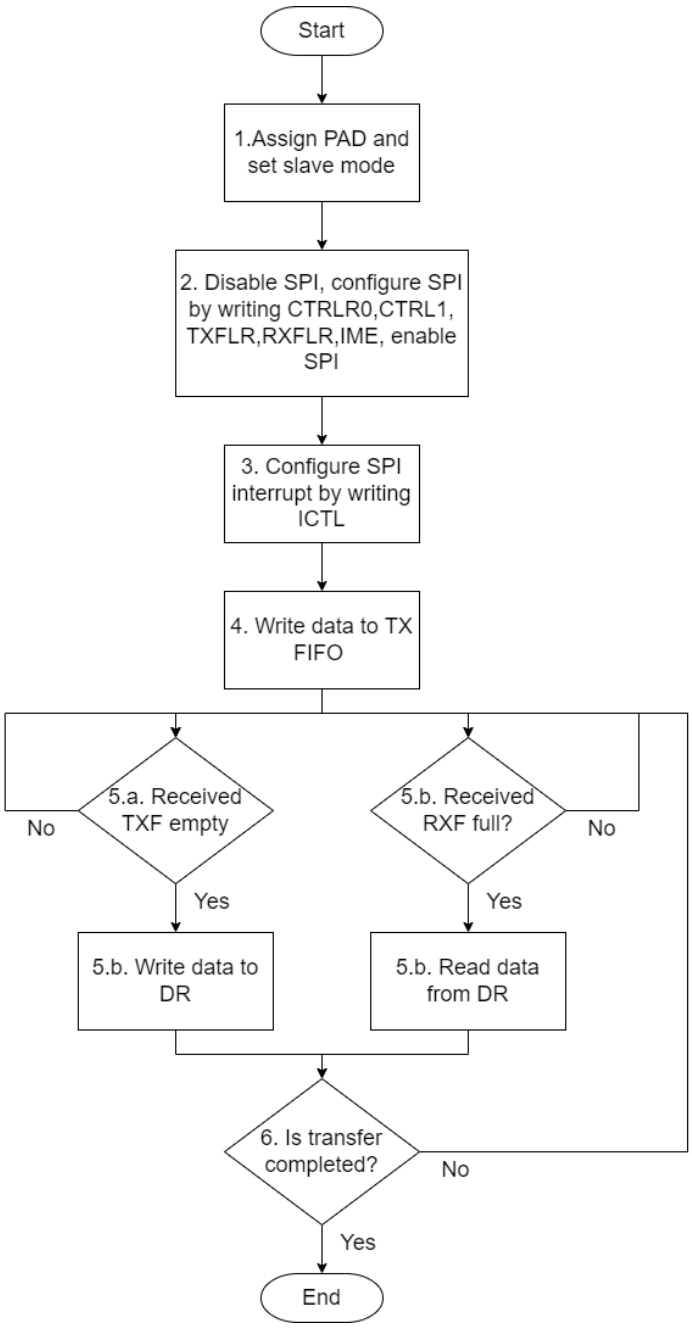


Figure 57 SPI Slave transmit & receive in interrupt mode

23.4.10 Slave transfer programming in DMA mode

The steps below show the programming sequence for SPI slave transfer with DMA.

1. Configure IO as SPI ports:
   MOSI, MISO, SS_N and SCLK.
   Set SPI module to slave mode in system config refer to session.
   Set SPI clock source and clock divider refer to session.

2. Initialize the SPI slave:
   a) Disable the SPI: write SPI_SSIEN.EN[0]=0.
   b) Configure the control transfer parameters: Write SPI_CTRLR0.
   c) If the transfer is received only, set the number of data frames for the transfer: Write SPI_CTRLR1 with the desired number of data frames minus 1.
   d) Set TX/RX FIFO threshold levels: Write SPI_TXFTLR and SPI_RXFTLR
   e) Unmask interrupt: write SPI_IMR.
   f) Enable the SPI Slave: write SPI_SSIEN.EN[0]=1.
3. Configure the DMA RX/TX channels
   a. Configure the appropriate DMA handshaking interface by writing DMA_CFG and DMA_CFG_2 in SYSCFG.
   b. For SPI RX, configure the receive channel parameters including source address, destination address, channel data length and handshaking interface. (Refer to Chapter 9 DMA controller for details)
   - Source address: SPI_DR register
   - Destination address: receive data buffer
   c. For SPI TX, configure the transmit channel parameters including source address, destination address, channel data length and handshaking interface.
   - Source address: Transmit data buffer
   - Destination address: SPI_DR register
   d. Enable DMA receive channel interrupt.
   e. Enable DMA transmit channel interrupt.
4. Associate the DMA channel to SPI.
5. Write data to TX FIFO
6. Perform data transfer by DMA.
7. Check whether data transfer is completed. If more data need to be received and transmitted, then wait for the DMA transfer.
8. SPI is in idle state. Disable interrupt, write SPI_SSIEN.EN[0]=0 to disable the SPI slave.

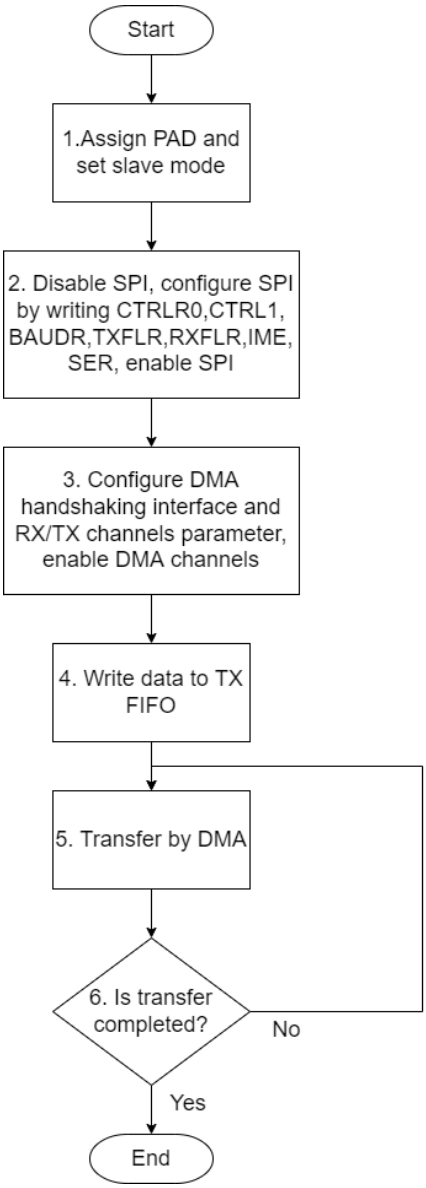


Figure 58 SPI Slave transmit & receive in polling mode

23.5 SPI registers

23.5.1 SPI_CTRLR0 (Control Register 0)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x00
Type: Read/Write
Reset value: 0x00000007

This register controls the serial data transfer. It can only be written when SPI is disabled. The register becomes read only when SPI is enabled.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:12>	**CFS[3:0]**: Sets the control frame size, that is the length of the control word for the Microwire frame format. 4'b0000: 1-bit control word 4'b0001: 2-bit control word 4'b0010: 3-bit control word 4'b0011: 4-bit control word 4'b0100: 5-bit control word 4'b0101: 6-bit control word 4'b0110: 7-bit control word 4'b0111: 8-bit control word 4'b1000: 9-bit control word 4'b1001: 10-bit control word 4'b1010: 11-bit control word 4'b1011: 12-bit control word 4'b1100: 13-bit control word 4'b1101: 14-bit control word 4'b1110: 15-bit control word 4'b1111: 16-bit control word Reset value: 0x0	R/W
<11>	**SRL**: Connects the transmit shift register output to receive shift register input for shift register loop. It used for testing purposes only. 1'b0: Normal Mode Operation 1'b1: Test Mode Operation Reset value: 0x0	R/W

Bit	Description	Type
<10>	**SLV_OE**: Enables/Disables the Slave output (SO). This bit is only relevant for Slave mode 1'b0: Slave transmit (SO) is enabled 1'b1: Slave transmit (SO) is disabled Reset value: 0x0	R/W
<9:8>	**TMOD[1:0]**: Set the mode of transfer. 2'b00: Transmit & Receive 2'b01: Transmit only 2'b10: Receive only 2'b11: Reserved Reset value: 0x0	R/W
<7>	**SCPOL**: Set the polarity of inactive serial clock. This bit is valid when the frame format is SPI. 1'b0: Inactive state of SCLK is low 1'b1: Inactive state of SCLK is high Reset value: 0x0	R/W
<6>	**SCPH**: Set the relationship of the serial clock with the slave select signal during data captured. This bit is valid when the frame format is SPI. 1'b0: Starts data capture on the 1st edge of SCLK 1'b1: Starts data capture on the 2nd edge of SCLK Reset value: 0x0	R/W
<5:4>	**FRF[1:0]**: Sets the frame format of serial protocol. 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: Reserved 2'b11: Reserved Reset value: 0x0	R/W
<3:0>	**DFS[3:0]**: Selects the data frame length. 4'b0000: Reserved-undefined operation 4'b0001: Reserved-undefined operation 4'b0010: Reserved-undefined operation 4'b0011: 4-bit serial data transfer 4'b0100: 5-bit serial data transfer 4'b0101: 6-bit serial data transfer 4'b0110: 7-bit serial data transfer 4'b0111: 8-bit serial data transfer 4'b1000: 9-bit serial data transfer 4'b1001: 10-bit serial data transfer 4'b1010: 11-bit serial data transfer 4'b1011: 12-bit serial data transfer	R/W

Bit	Description	Type
	4'b1100: 13-bit serial data transfer 4'b1101: 14-bit serial data transfer 4'b1110: 15-bit serial data transfer 4'b1111: 16-bit serial data transfer Reset value: 0x7	

23.5.2 SPI_CTRLR1 (Control Register 1)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x04
Type: Read/Write
Reset value: 0x00000000

This register is effective only when SPI is configured as master. When it is configured as slave, writing to this register has no effect. The register can only be written when SPI is disabled and becomes read only when SPI is enabled.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**NDF[15:0]**: Sets the number of data frames which will be continuously received. 0x0000: the data frame is 1 . . . . 0xFFFF: the data frame is 65536 Reset value: 0x0	R/W

23.5.3 SPI_SSIEN (SPI Enable Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x08
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**EN**: Enables/Disables all SPI operations. 1'b0: enable SPI 1'b1: Disable SPI	R/W

23.5.4 SPI_MWCR (Microwire Control Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x0C
Type: Read/Write
Reset value: 0x00000000

This register is valid if the frame format is Microwire protocol (CTRLR0[5:4]=2), and it is writable when the SPI is disabled (SPI_SSIEN[0]=0).

Bit	Description	Type
<31:3>	Reserved.	R/W
<2>	**MHS**: Enables/Disables the "busy/ready" handshaking interface for the Microwire protocol 1'b0: Disable the handshaking interface 1'b1: Enable the handshaking interface	R/W
<1>	**MDD**: Defines the direction of the data word for Microwire protocol 1'b0: The external device transmits data word to the MCU 1'b1: The MCU transmits data word to the external device	R/W
<0>	**MWMOD**: Sets transfer mode of Microwire protocol. 1'b0: Non-sequential transfer 1'b1: Sequential transfer	R/W

23.5.5 SPI_SER (Slave Enable Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x10
Type: Read/Write
Reset value: 0x00000000

This register is effective only when SPI is configured as master. It can only be written when SPI is idle or disabled.

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**SER**: Select and enable the slave line #0. 1'b0: Not selected 1'b1: Selected	R/W

23.5.6 SPI_BAUDR (Baud Rate Select)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x14
Type: Read/Write
Reset value: 0x00000000
This register is effective only when SPI is configured as master. It can only be written when the SPI is disabled (SPI_SSIEN[0]=0)

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**BR[15:0]**: SPI SCLK divider. Set the SPI SCLK divider to any even value between 2 to 65534. If the value is 0, the SCLK is disabled.	R/W

23.5.7 SPI_TXFTLR (Transmit FIFO Threshold Level Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x18
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved.	R/W

Bit	Description	Type
<2:0>	**TFT[2:0]**: Sets the transmit FIFO threshold level value. This field specifies the level of FIFO data entries at which an interrupt is generated. 3'b000: 0 entry (TXE interrupt is asserted when 0 entries in TX FIFO) 3'b001: 1 entry (TXE interrupt is asserted when 1 or less entries are in TX FIFO) 3'b010: 2 entries (TXE interrupt is asserted when 2 or less entries are in TX FIFO) 3'b011: 3 entries (TXE interrupt is asserted when 3 or less entries are in TX FIFO) 3'b100: 4 entries (TXE interrupt is asserted when 4 or less entries are in TX FIFO) 3'b101: 5 entries (TXE interrupt is asserted when 5 or less entries are in TX FIFO) 3'b110: 6 entries (TXE interrupt is asserted when 6 or less entries are in TX FIFO) 3'b111: 7 entries (TXE interrupt is asserted when 7 or less entries are in TX FIFO)	R/W

23.5.8 SPI_RXFTLR (Receive FIFO Threshold Level Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x1C
Type: Read/Write
Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved.	R/W
<2:0>	**RFT[2:0]**: Sets the receive FIFO threshold level value. This field specifies the level of FIFIO data entries at which an interrupt is generated. 3'b000: 1 entry (RXF interrupt is asserted when 1 or more entries are in RX FIFO) 3'b001: 2 entries (RXF interrupt is asserted when 2 or more entries are in RX FIFO) 3'b010: 3 entries (RXF interrupt is asserted when 3 or more entries are in RX FIFO) 3'b011: 4 entries (RXF interrupt is asserted when 4 or more entries are in RX FIFO) 3'b100: 5 entries (RXF interrupt is asserted when 5 or more entries are in RX FIFO) 3'b101: 6 entries (RXF interrupt is asserted when 6 or more entries are in RX FIFO) 3'b110: 7 entries (RXF interrupt is asserted when 7 or more entries are in RX FIFO) 3'b111: 8 entries (RXF interrupt is asserted when 8 or more entries are in RX FIFO)	R/W

23.5.9 SPI_TXFLR (Transmit FIFO Level Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600

SPI2 slave 0x4000_2700

Address offset: 0x20

Type: Read-only

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved.	R/W
<3:0>	**TFL[3:0]**: Contains the number of valid entries in the transmit FIFO	R/W

23.5.10 SPI_RXFLR (Receive FIFO Level Register)

Base address:

SPI1 master 0x4000_2400

SPI1 slave 0x4000_2500

SPI2 master 0x4000_2600

SPI2 slave 0x4000_2700

Address offset: 0x24

Type: Read-only

Reset value: 0x00000000

Bit	Description	Type
<31:4>	Reserved.	R/W
<3:0>	**RFL[3:0]**: Contains the number of valid entries in the receive FIFO	R/W

23.5.11 SPI_SR (Status Register)

Base address:

SPI1 master 0x4000_2400

SPI1 slave 0x4000_2500

SPI2 master 0x4000_2600

SPI2 slave 0x4000_2700

Address offset: 0x28

Type: Read-only

Reset value: 0x00000006

Bit	Description	Type
<31:7>	Reserved.	R/W
<6>	**DCOL**: Indicates the data collision error for SPI. It is only relevant when the SPI is configured as a master. The data collision error occurred if another master selects this device as a slave while data is transmitting. This bit is cleared when read. 1'b0: No error 1'b1: Data collision error	R/W
<5>	**TXE**: Indicates the transmission error for SPI. This bit is relevant only when SPI	R/W

Bit	Description	Type
	is configured as a slave. The transmission error occurred if the transmit FIFO is empty when a transfer is started. This bit is cleared when read. 1'b0: No error 1'b1: transmission error	
<4>	**RFF**: Indicates the receive FIFO full. 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full	R/W
<3>	**RFNE**: Indicates the receive FIFO not empty. 1'b0: Receive FIFO is empty 1'b1: Receive FIFO is not empty	R/W
<2>	**TFE**: Indicates the transmit FIFO empty. 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty	R/W
<1>	**TFNF**: Indicates the transmit FIFO not full. 1'b0: Transmit FIFO is full 1'b1: Transmit FIFO is not full	R/W
<0>	**BUSY**: Indicates the state of SPI busy. 1'b0: SPI is idle or disabled 1'b1: SPI is busy	R/W

23.5.12 SPI_IMR (Interrupt Mask Register)

Base address:

SPI1 master 0x4000_2400 (0x0000003F)
SPI1 slave 0x4000_2500 (0x0000001F)
SPI2 master 0x4000_2600 (0x0000003F)
SPI2 slave 0x4000_2700 (0x0000001F)

Address offset: 0x2C

Type: Read/Write

Reset value: 0x0000003F

Bit	Description	Type
<31:6>	Reserved.	R/W
<5>	**MST**: Sets multi-Master contention (MST) interrupt mask. The bit is only for SPI. It is available when SPI is configured as a slave. 1'b0: Interrupt masked 1'b1: Interrupt not masked Reset value: 0x1	R/W
<4>	**RXF**: Sets receive FIFO full (RXF) interrupt mask. 1'b0: Interrupt masked 1'b1: Interrupt not masked	R/W

	Reset value: 0x1	
<3>	**RXO**: Sets receive FIFO overflow (RXO) interrupt mask. 1'b0: Interrupt masked 1'b1: Interrupt not masked Reset value: 0x1	R/W
<2>	**RXU**: Sets receive FIFO underflow (RXU) interrupt mask. 1'b0: Interrupt masked 1'b1: Interrupt not masked Reset value: 0x1	R/W
<1>	**TXO**: Sets transmit FIFO overflow (TXO) interrupt mask. 1'b0: Interrupt masked 1'b1: Interrupt not masked Reset value: 0x1	R/W
<0>	**TXE**: Sets transmit FIFO empty (TXE) interrupt mask. 1'b0: Interrupt masked 1'b1: Interrupt not masked Reset value: 0x1	R/W

23.5.13 SPI_ISR (Interrupt Status Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x30

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:6>	Reserved.	R/W
<5>	**MST**: Indicates the active status of multi-Master contention (MST) interrupt after masking. It is available when SPI is configured as a slave. 1'b0: Interrupt not active after masking 1'b1: Interrupt active after masking Reset value: 0x0	R/W
<4>	**RXF**: Indicates the active status of receive FIFO full (RXF) interrupt after masking. 1'b0: Interrupt not active after masking 1'b1: Interrupt active after masking Reset value: 0x0	R/W
<3>	**RXO**: Indicates the active status of receive FIFO overflow (RXO) interrupt after masking.	R/W

Bit	Description	Type
	1'b0: Interrupt not active after masking 1'b1: Interrupt active after masking Reset value: 0x0	
<2>	**RXU**: Indicates the active status of receive FIFO underflow (RXU) interrupt after masking. 1'b0: Interrupt not active after masking 1'b1: Interrupt active after masking Reset value: 0x0	R/W
<1>	**TXO**: Indicates the active status of transmit FIFO overflow (TXO) interrupt after masking. 1'b0: Interrupt not active after masking 1'b1: Interrupt active after masking Reset value: 0x0	R/W
<0>	**TXE**: Indicates the active status of transmit FIFO empty (TXE) interrupt after masking. 1'b0: Interrupt not active after masking 1'b1: Interrupt active after masking Reset value: 0x0	R/W

23.5.14 SPI_RISR (Raw Interrupt Status Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x34

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:6>	Reserved.	R/W
<5>	**MST**: Indicates the active status of multi-Master contention (MST) raw interrupt before masking. It is available when SPI is configured as a slave. 1'b0: Interrupt not active before masking 1'b1: Interrupt active before masking Reset value: 0x0	R/W
<4>	**RXF**: Indicates the active status of receive FIFO full (RXF) raw interrupt before masking. 1'b0: Interrupt not active before masking 1'b1: Interrupt active before masking Reset value: 0x0	R/W

Bit	Description	Type
<3>	**RXO**: Indicates the active status of receive FIFO overflow (RXO) raw interrupt before masking. 1'b0: Interrupt not active before masking 1'b1: Interrupt active before masking Reset value: 0x0	R/W
<2>	**RXO**: Indicates the active status of receive FIFO underflow (RXU) raw interrupt before masking. 1'b0: Interrupt not active before masking 1'b1: Interrupt active before masking Reset value: 0x0	R/W
<1>	**TXO**: Indicates the active status of transmit FIFO overflow (TXO) raw interrupt before masking. 1'b0: Interrupt not active before masking 1'b1: Interrupt active before masking Reset value: 0x0	R/W
<0>	**TXE**: Indicates the active status of transmit FIFO empty (TXE) raw interrupt before masking. 1'b0: Interrupt not active before masking 1'b1: Interrupt active before masking Reset value: 0x0	R/W

23.5.15 SPI_TXO_ICR (Transmit FIFO Overflow Interrupt Clear Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x38
Type: Read
Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Clears the transmit FIFO overflow (TXO) interrupt upon reading this bit.	R/W

23.5.16 SPI_RXO_ICR (Receive FIFO Overflow Interrupt Clear Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x3C

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Clears the receive FIFO overflow (RXO) interrupt upon reading this bit.	R/W

23.5.17 SPI_RXU_ICR (Receive FIFO Overflow Interrupt Clear Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x40

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Clears the receive FIFO underflow (RXU) interrupt upon reading this bit.	R/W

23.5.18 SPI_MST_ICR (Multi-Master Interrupt Clear Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x44

Type: ReadReset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Clears the multi-Master contention (MST) interrupt upon reading this bit.	R/W

23.5.19 SPI_ICR (Interrupt Clear Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x48

Type: Read

Reset value: 0x00000000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**CLR**: Clear TXO/RXO/RXU/MST interrupts. A read clears all interrupts. 1'b0: No interrupt pending. 1'b1: One or more interrupt in TXO/RXO/RXU/MST are pending Reset value: 0x0	R/W

23.5.20 SPI_DMA_CR (DMA Control Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x4C

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:2>	Reserved.	R/W
<1>	**TDMAE**: Enables/Disables the transmit FIFO DMA channel. 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled Reset value: 0x0	R/W
<0>	**RDMAE**: Enables/Disables the receive FIFO DMA channel. 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled Reset value: 0x0	R/W

23.5.21 SPI_DMA_TDLR (DMA Transmit Data Level Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x50

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved.	R/W
<2:0>	**TDL[2:0]**: Sets the DMA transmit data level to control DMA TX request generation when the valid data entries in TX FIFO is less than or equal to the value set. 3'b000: 0 entry (DMA TX request is asserted if 0 entries are in TX FIFO) 3'b001: 1 entry (DMA TX request is asserted if 1 or less entries are in TX FIFO) 3'b010: 2 entries (DMA TX request is asserted if 2 or less entries are in TX FIFO) 3'b011: 3 entries (DMATX request is asserted if 3 or less entries are in TX FIFO) 3'b100: 4 entries (DMATX request is asserted if 4 or less entries are in TX FIFO) 3'b101: 5 entries (DMATX request is asserted if 5 or less entries are in TX FIFO) 3'b110: 6 entries (DMATX request is asserted if 6 or less entries are in TX FIFO) 3'b111: 7 entries (DMATX request is asserted if 7 or less entries are in TX FIFO) Reset value: 0x0	R/W

23.5.22 SPI_DMA_RDLR (DMA Receive Data Level Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x54

Type: Read/Write

Reset value: 0x00000000

Bit	Description	Type
<31:3>	Reserved.	R/W
<2:0>	**RDL[2:0]**: Sets the DMA receive data level to control DMA RX request generation when the valid data entries in RX FIFO is greater than or equal to the value set. 3'b000: 1 entry (DMA RX request is asserted if 1 or more entries are in RX FIFO) 3'b001: 2 entries (DMA RX request is asserted if 2 or more entries are in RX FIFO) 3'b010: 3 entries (DMA RX request is asserted if 3 or more entries are in RX FIFO)	R/W

Bit	Description	Type
	3’b011: 4 entries (DMA RX request is asserted if 4 or more entries are in RX FIFO) 3’b100: 5 entries (DMA RX request is asserted if 5 or more entries are in RX FIFO) 3’b101: 6 entries (DMA RX request is asserted if 6 or more entries are in RX FIFO) 3’b110: 7 entries (DMA RX request is asserted if 7 or more entries are in RX FIFO) 3’b111: 8 entries (DMA RX request is asserted if 8 or more entries are in RX FIFO)	

23.5.23 SPI_DR (Data Register)

Base address:

SPI1 master 0x4000_2400
SPI1 slave 0x4000_2500
SPI2 master 0x4000_2600
SPI2 slave 0x4000_2700

Address offset: 0x60Type: Read/Write

Reset value: 0x00000000

The register is a 16-bit data buffer for the transmit/receive FIFOs. It becomes writable when SPI is enabled (SPI_SSIEN[0]=1).

When this register is read, data in the receive FIFO buffer is accessed.

When this register is written, data are moved into the transmit FIFO buffer.

Bit	Description	Type
<31:16>	Reserved.	R/W
<15:0>	**DR[15:0]**: Transfer data buffer	R/W

23.5.24 SPI register map

The base address of the SPI registers is
SPI1 master 0x4000_2400;
SPI1 slave 0x4000_2500;
SPI2 master 0x4000_2600;
SPI2 slave 0x4000_2700.
The following table gives the SPI register map and reset values.

Table 56 SPI register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPI_CTRL0	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CFS[3:0]				SRL	SLV_OE	TMOD[1:0]		SCPOL	SCPH	FRF[1:0]		DFS[3:0]			
	Reset value	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	1	1	1
0x04	SPI_CTRL1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDF[15:0]															
	Reset value	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
0x08	SPI_SSIEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	EN
	Reset value	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
0x0C	SPI_MWCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MHS	MDD	MWMOD
	Reset value	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
0x10	SPI_SER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SER
	Reset value	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
0x14	SPI_BAUDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BR[15:0]															
	Reset value	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
0x18	SPI_TXFTLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TFT[2:0]		
	Reset value	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
0x1C	SPI_RXFTLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RFT[2:0]		
	Reset value	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
0x20	SPI_TXFLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TFL[3:0]			
	Reset value	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
0x24	SPI_RXFLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RFL[3:0]			
	Reset value	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
0x28	SPI_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DCOL	TXE	RFF	RFNE	TFE	TFNF	BUSY
	Reset value	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	1	1	0
0x2C	SPI_IMR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MST	RXF	RXO	RXU	TXO	TXE
	Reset value	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	1	1	1	1	1
0x30	SPI_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MST	RXF	RXO	RXU	TXO	TXE
	Reset value	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
0x34	SPI_RISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MST	RXF	RXO	RXU	TXO	TXE
	Reset value	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
0x38	SPI_TXOICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x3C	SPI_RXOICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	SPI_RXUICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x44	SPI_MSTICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x48	SPI_ICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CLR
	Reset value	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
0x4C	SPI_DMACR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDMAE	RMDAE
	Reset value	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
0x50	SPI_DMATDLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDL[2:0]		
	Reset value	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
0x54	SPI_DMARDLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RDL[2:0]		
	Reset value	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
0x60	SPI_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DR[15:0]															
	Reset value	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

24 LED driver (LED)

24.1 Introduction

XTX LED driver is an APB (Advanced Peripheral Bus) component and is driver designed for LED panel.

A compact common-anode LED display driver is implemented in XT32H series, with 9 ports LED0~LED8 for external LED driving.

All the ports LED0~LED8 can be configured as segments, each of which sinks a constant DC current through external LED. Individual 32 DC current levels can be set in each segment, with the range of 0~19.6mA. Furthermore, for each segment, an 8-bit PWM control signal tunes the current duty cycle, as well as the LED brightness.

Ports LED0~LED7 can be configured as common cells, with maximum sourcing current of 160mA.

The LED driver can directly drive up to 64 discrete LEDs, or 8 8-segment digits. Figure 55 shows the LED 8x8 matrix connection to the LED driver. Table 48 shows a suggested driving scheme for 8 8-segment digits connections.

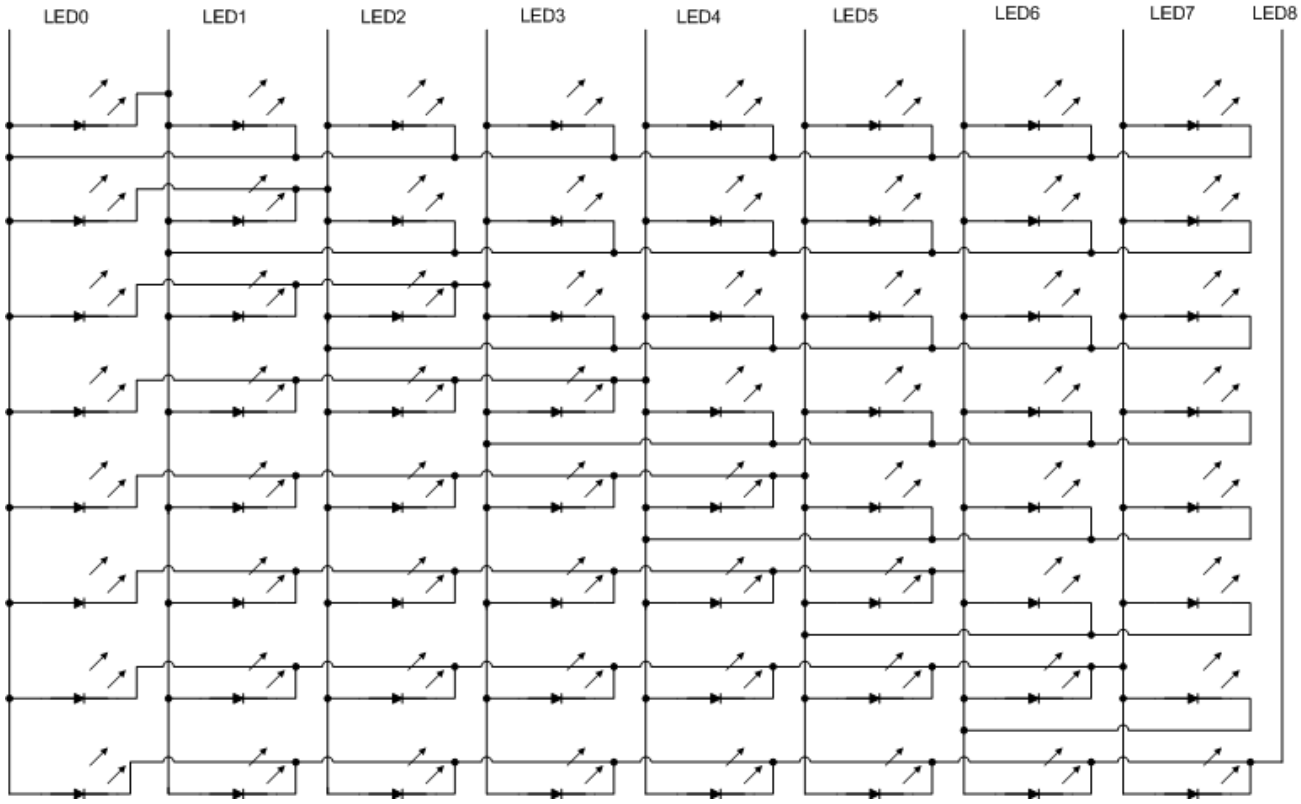


Figure 59 LED 8x8 Matrix

Table 57 A suggested driving scheme of 8 8-segment digits

	LED0	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8
DIGIT0	COM0	SEG a0	SEG b0	SEG c0	SEG d0	SEG e0	SEG f0	SEG g0	SEG dp0
DIGIT1	SEG a1	COM1	SEG b1	SEG c1	SEG d1	SEG e1	SEG f1	SEG g1	SEG dp1
DIGIT2	SEG a2	SEG b2	COM2	SEG c2	SEG d2	SEG e2	SEG f2	SEG g2	SEG dp2
DIGIT3	SEG a3	SEG b3	SEG c3	COM3	SEG d3	SEG e3	SEG f3	SEG g3	SEG dp3
DIGIT4	SEG a4	SEG b4	SEG c4	SEG d4	COM4	SEG e4	SEG f4	SEG g4	SEG dp4
DIGIT5	SEG a5	SEG b5	SEG c5	SEG d5	SEG e5	COM5	SEG f5	SEG g5	SEG dp5
DIGIT6	SEG a6	SEG b6	SEG c6	SEG d6	SEG e6	SEG f6	COM6	SEG g6	SEG dp6
DIGIT7	SEG a7	SEG b7	SEG c7	SEG d7	SEG e7	SEG f7	SEG g7	COM7	SEG dp7

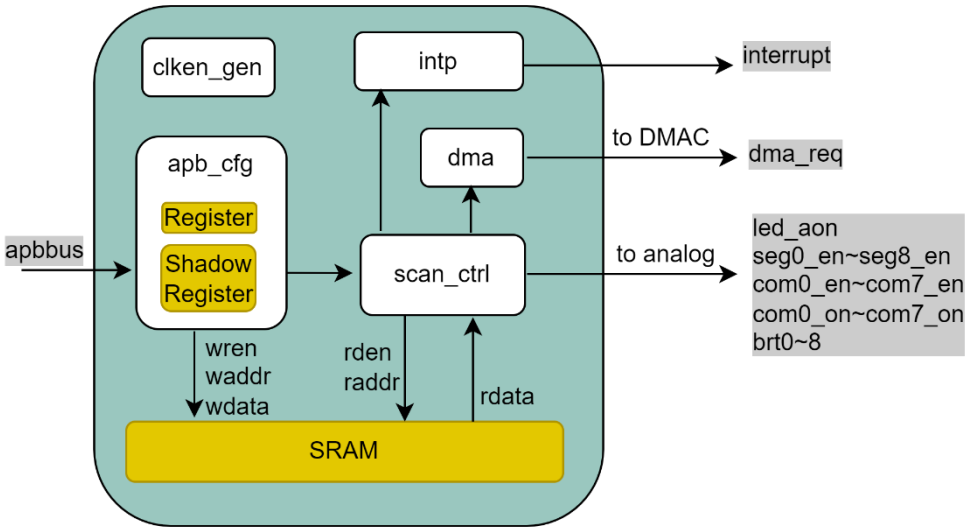


Figure 60 LED driver diagram

24.2 LED driver main features

- LED display driver
- LED COM and SEG control
- Two run modes (on/off and bright)
- Display frequency control
- Frame buffer control
- Animotion support
- Brightness control
- DMA support
- Interrupt support
- Error report

24.3 LED driver signal descriptions

The LED signals are shown in the following table.

Table 58 LED signal descriptions

Signal	Description	I/O
LED0	LED COM_0 and SEG_0 line	O
LED1	LED COM_1 and SEG_1 line	O
LED2	LED COM_2 and SEG_2 line	O
LED3	LED COM_3 and SEG_3 line	O
LED4	LED COM_4 and SEG_4 line	O
LED5	LED COM_5 and SEG_5 line	O
LED6	LED COM_6 and SEG_6 line	O
LED7	LED COM_7 and SEG_7 line	O
LED8	LED SEG_8 line	O

24.4 LED driver functional description

24.4.1 Run mode

LED provide two running modes. They are:

1. On-off mode : LED turn on/off is controlled by 1 bit in segment.
2. Brightness mode: LED lighten by SEG pulse and bright by width of SEG pulse.

24.4.1.1 On-off mode

1. Write 0 to LED_MODE.RMD[17] to set on/off mode.
2. Update display buffer information.
   a. Calculate buffer frame size and frame number

$$Size_{frame} = \text{Around}\left(\frac{MAX(Index_{COM})}{2}\right) \quad \text{(Equation 24-1)}$$

$$Number_{frame} = \frac{128}{Size_{frame}} \quad \text{(Equation 24-2)}$$

   b. Write frame size to LED_MODE.DN[27:25], write (frame number – 1) to LED_MODE.BFN[24:18] and write (frame number * frame size -1) to LEDMODE.ALAST[6:0]

24.4.1.2 Brightness mode

1. Write 1 to LED_MODE.RMD[17] to set brightness mode.
2. Update display buffer information.
   a. Calculate buffer frame size and frame number

$$Size_{frame} = Number_{COM} \times 5 \quad \text{(Equation 24-3)}$$

$$Number_{frame} = \frac{128}{Size_{frame}} \quad \text{(Equation 24-4)}$$

   b. Write (frame number – 1) to LED_MODE.BFN[24:18] and write (frame number * frame size -1) to LEDMODE.ALAST[6:0]

24.4.2 Frame format

When enabled COM number is less than 8, the data of COM will assign to the maximum index COM for on-off mode. For brightness mode, the data of COM will assign one by one. For example, when COM1 and COM8 are enabled, data of COM1 is the first data and is followed by data of COM8 in brightness mode. If LED works in on-off mode, one frame occupied 4 words, COM0, COM2 to COM7 would be filled with 0.

Table 59 Display buffer data format in on-off mode

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

0x00	Frame	Res	Res	Res	Res	Res	Res	Res	COM1[8:0]	Res	Res	Res	Res	Res	Res	Res	COM0[8:0]
0x04		Res	Res	Res	Res	Res	Res	Res	COM3[8:0]	Res	Res	Res	Res	Res	Res	Res	COM2[8:0]
0x08		Res	Res	Res	Res	Res	Res	Res	COM5[8:0]	Res	Res	Res	Res	Res	Res	Res	COM4[8:0]
0x0C		Res	Res	Res	Res	Res	Res	Res	COM7[8:0]	Res	Res	Res	Res	Res	Res	Res	COM6[8:0]

Table 60 Display buffer data format in brightness mode

Byte	Description	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
0x00	Frame COM0	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	0	0
0x04		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x08		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x0C		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x10		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x14	Frame COM1	0	0	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x18		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x1C		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x20		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x24		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x28	Frame COM2	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x2C		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	0	0
0x30		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x34		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x38		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x3C	Frame COM3	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x40		0	0	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x44		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x48		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x4C		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x50	Frame COM4	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x54		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x58		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	0	0
0x5C		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x60		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x64	Frame COM5	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x68		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x6C		0	0	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x70		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x74		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x78	Frame COM6	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x7C		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x80		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x84		SEG7_STRTPOS[7:0]	SEG7_STPPOS[7:0]	0	0
0x88		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]
0x8C	Frame COM7	SEG1_STRTPOS[7:0]	SEG1_STPPOS[7:0]	SEG0_STRTPOS[7:0]	SEG0_STPPOS[7:0]
0x90		SEG3_STRTPOS[7:0]	SEG3_STPPOS[7:0]	SEG2_STRTPOS[7:0]	SEG2_STPPOS[7:0]
0x94		SEG5_STRTPOS[7:0]	SEG5_STPPOS[7:0]	SEG4_STRTPOS[7:0]	SEG4_STPPOS[7:0]
0x98		0	0	SEG6_STRTPOS[7:0]	SEG6_STPPOS[7:0]
0x9C		Res	Res	SEG8_STRTPOS[7:0]	SEG8_STPPOS[7:0]

24.4.3 LED initial and turn on/off

24.4.3.1 Initialization

1. LED driver analog on. Write 1 to LED_CTRL.AON[29].
2. Set display frequency and dead time.

$$PSC = \frac{Freq_{APB}}{\left(Num_{COM} + (256 + Time_{Dead})\right) \times Freq_{SCAN}} \quad \text{(Equation 24-1)}$$

Write PSC to LED_CFG.PSC[15:0] and write dead time to LED_CFG.DTW[23:16]

3. Enable COM and SEG, write COM mapping to LED_CTRL.CxE[7:0] and write SEG mapping to LED_CTRL.SxE[16:8]
4. Set current bias. Write current bias to LED_CTRL.ITRIM[28:25]
5. Configure run mode, display mode and brightness step. Write run mode to LED_MODE.RMD[17], write display mode to LED_MODE.DMD[16] and write brightness step to LED_MODE.BRTS[15:11]
6. Configure display buffer, according to run mode to configure display buffer, refer to session 14.4.1.
7. Configure SEG brightness maximum value, write SEG lines brightness maximum value to LED_BRGx.BRTx
8. Enable frame interrupt. Write 1 to LED_CTRL.FIE[22]

24.4.3.2 Turn on display

1. Write 1 to LED_EG.UG[0]
2. Write 1 to LED_CTRL.LEDEN[17]

24.4.3.3 Turn off display

Write 0 to LED_CTRL_LEDEN[17]

24.4.4 Display mode

24.4.4.1 Dimming

1. Write 1 to LED_MODE.DMD[16] to set dimming display mode.
2. Write brightness step to LED_MODE.DIMS[10:8]
3. Write dimming interpolator to LED_CFG.IFN[31:24]

24.4.4.2 Blink

1. Write 1 to LED_MODE.DMD[16] to set dimming display mode.
2. Write 4 to LED_MODE.DIMS[10:8]
3. Write 2 to LED_CFG.IFN[31:24]

24.4.4.3 Normal display

1. Write 0 to LED_MODE.DMD[16] to set normal display mode.
2. Write 0 to LED_MODE.DIMS[10:8]
3. Write 0 to LED_CFG.IFN[31:24]

24.4.5 Example for work flow

This example described LED displays by frames and transfers data with DMA. It used two levels buffer they are screen buffer in application software and display buffer in LED driver (hardware). Application updates screen buffer to display screen design. DMA transfers frame data to display buffer.

Two levels buffer implements auto correction, shown in Figure 61 and Figure 62.

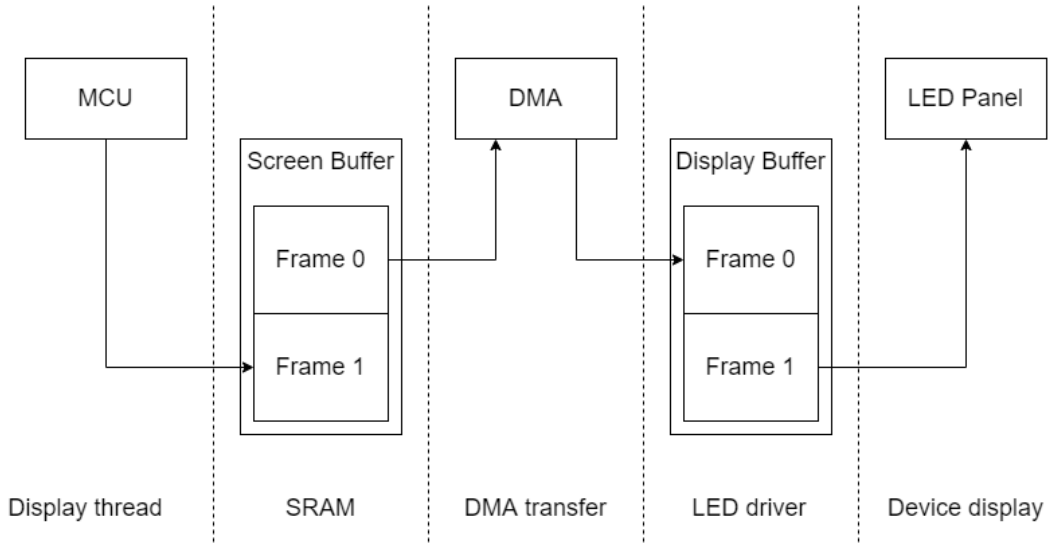


Figure 61 Display data flow

Note:

1. In initial, MCU fill the first display screen to all frames in two buffers.
2. When MCU set LED_ON to LED driver, LED driver read data from display buffer frame by frame continuously.
3. MCU recorded update frame in screen buffer. When MCU received frame completed interrupt from LED driver and updated screen buffer, MCU read display frame status from LED driver to decide DMA destination. MCU configure DMA source, destination and block size and software trigger DMA transfer. If MCU no more updated frame, MCU will trigger DMA to refresh all frames in the display buffer.
4. We assumed DMA transfer can be completed in during of display one frame. When DMA transfer did not complete in one display frame, the system display on error frame and correct data in the following frames (shown in Figure 5).
5. In all display modes, LED driver read data from display buffer frame by frame continuously.
6. Screen buffer frame format is the same as display buffer frame format.
7. When MCU write LED_OFF, LED driver read pointer of display buffer will be set to the beginning of frame 0.

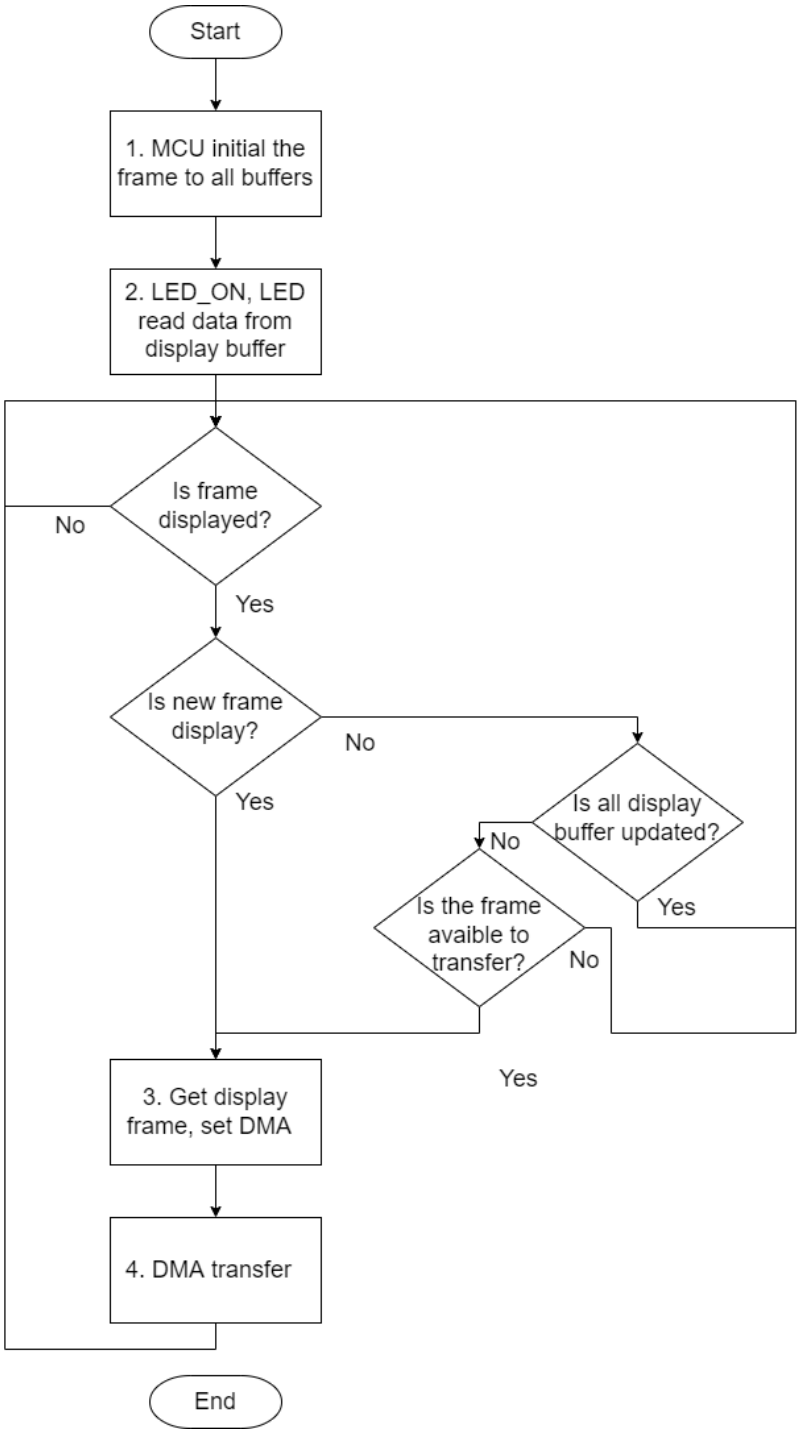


Figure 62 Display process

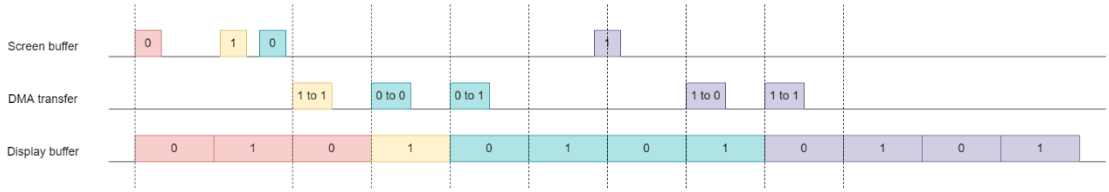


Figure 63 Example for display data flow

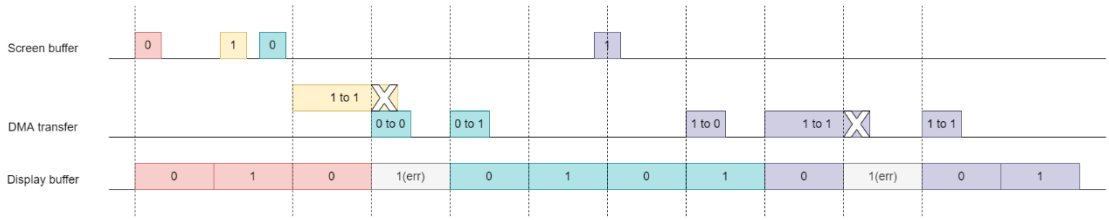


Figure 64 Example for display data when DMA error occurs

24.5 LED register

24.5.1 LED_CTRL

Base address: 0x4000_3000
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:30>	Reserved	R/W
<29>	**AON**: LED analog module ON 1'b0: LED analog module off 1'b1: LED analog module on	R/W
<28:25>	**ITRIM[3:0]**: current bias for SEG cell 4'b0000:	R/W
<24>	**PIE**: Pre-read interrupt enable 1'b0: disable 1'b1: enable	R/W
<23>	**CIE**: COM interrupt enable 1'b0: disable 1'b1: enable	R/W
<22>	**FIE**: Frame interrupt enable 1'b0: disable 1'b1: enable	R/W
<21>	**UIE**: Updare interrupt enable 1'b0: disable 1'b1: enable	R/W
<20>	**CDE**: COM DMA request enable 1'b0: disable 1'b1: enable	R/W
<19>	**FDE**: Frame DMA request enable 1'b0: disable 1'b1: enable	R/W
<18>	**UDE**: Update DMA request enable 1'b0: disable 1'b1: enable	R/W
<17>	**LEDEN**: LED enable 1'b0: disable 1'b1: enable	R/W
<16:8>	**SxE[8:0]**: SEG x (x=0 to 8) enable	R/W

Bit	Description	Type
	1'b0: disable 1'b1: enable	
<7:0>	**CxE[7:0]**: COM x (x=0 to 7) enable 1'b0: disable 1'b1: enable	R/W

24.5.2 LED_CFG

Base address: 0x4000_3000
Address offset: 0x04
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**IFN[7:0]**: interpolator frame number of dimming or blinking The default value is 0xFF.	R/W
<23:16>	**DTW[7:0]**: COM deadtime width. The deadtime is n+1.	R/W
<15:0>	**PSC[15:0]**: prescalar for LED clock. 0 is no divider. Other means divider is (n+1).	R/W

24.5.3 LED_MODE

Base address: 0x4000_3000
Address offset: 0x08
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type	
<31:28>	Reserved.	R/W	
<27:25>	**DN[2:0]**: Data number of one frame in on-off mode.	R/W	
<24:18>	**BFN[6:0]**: Frame number in buffer. The value is n+1.	R/W	
<17>	**RMD**: run mode. 1'b0: on-off mode 1'b1: duty mode	R/W	
<16>	**DMD**: display mode 1'b0: static mode 2'b1: dimming mode	R/W	
<15:11>	**BRTS[4:0]**: Bright step. 0 is the maximum value, The other value is n.	R/W	
<10:8>	**DIMS[2:0]**: the dimming step in dimming mode 3'b000: 1/16 3'b001: 1/8 3'b010: ¼ 3'b011: ½	R/W	

Bit	Description	Type	
	3’b100: 1		
<7>	Reserved.	R/W	
<6:0>	**ALAST[6:0]**: SRAM address of the last valid data.	R/W	

24.5.4 LED_BRG0

Base address: 0x4000_3000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:29>	Reserved.	R/W
<28:24>	**BRT3[4:0]**: The maximum brightness of SEG3.	R/W
<23:21>	Reserved.	R/W
<20:16>	**BRT2[4:0]**: The maximum brightness of SEG2.	R/W
<15:13>	Reserved.	R/W
<12:8>	**BRT1[4:0]**: The maximum brightness of SEG1.	R/W
<7:5>	Reserved.	R/W
<4:0>	**BRT0[4:0]**: The maximum brightness of SEG0.	R/W

24.5.5 LED_BRG1

Base address: 0x4000_3000
Address offset: 0x10
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:29>	Reserved.	R/W
<28:24>	**BRT7[4:0]**: The maximum brightness of SEG7.	R/W
<23:21>	Reserved.	R/W
<20:16>	**BRT6[4:0]**: The maximum brightness of SEG6.	R/W
<15:13>	Reserved.	R/W
<12:8>	**BRT5[4:0]**: The maximum brightness of SEG5.	R/W
<7:5>	Reserved.	R/W
<4:0>	**BRT4[4:0]**: The maximum brightness of SEG4.	R/W

24.5.6 LED_BRG2

Base address: 0x4000_3000
Address offset: 0x14

Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:5>	Reserved.	R/W
<4:0>	**BRT8[4:0]**: The maximum brightness of SEG8.	R/W

24.5.7 LED_EG

Base address: 0x4000_3000

Address offset: 0x18

Type: Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:1>	Reserved.	R/W
<0>	**UG**: update event. The bit is cleared by hardware. 1'b0: no action; 1'b1: generate update event.	W

24.5.8 LED_STATUS

Base address: 0x4000_3000

Address offset: 0x1C

Type: Read

Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved.	R/W
<3>	**PIF**: pre-read interrupt status 1'b0: no interrupt 1'b1: interrupt	R
<2>	**CIF**: COM interrupt status 1'b0: no interrupt 1'b1: interrupt	R
<1>	**FIF**: Frame interrupt status 1'b0: no interrupt 1'b1: interrupt	R
<0>	**UIF**: Update interrupt status. 1'b0: no interrupt; 1'b1: interrupt.	R

24.5.9 LED_DEBUG

Base address: 0x4000_3000
Address offset: 0x20
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:21>	Reserved.	R/W
<20>	**ADONERR**: analog switch error 1'b0: software writes 0 to clear the flag 1'b1: AON I s0 when DON is 1.	R/W0
<19>	**BUFERR**: Buffer config error 1'b0: software writes 0 to clear the flag 1'b1: config buffer when LED is enabled.	R/W0
<18:16>	**FSM[2:0]**: FSM status	R
<15:7>	Reserved.	R/W
<6:0>	**FRMN[6:0]**: Current display frame index in buffer.	R

24.5.10 LED_BUFFER

Base address: 0x4000_3000
Address offset: 0x40
Size : 128 Words
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:21>	**FDW[31:0]**: Frame data word.	W

24.5.11 LED register map

The base address of the LED registers is 0x4000_3000.
The following table gives the LED register map and reset values.

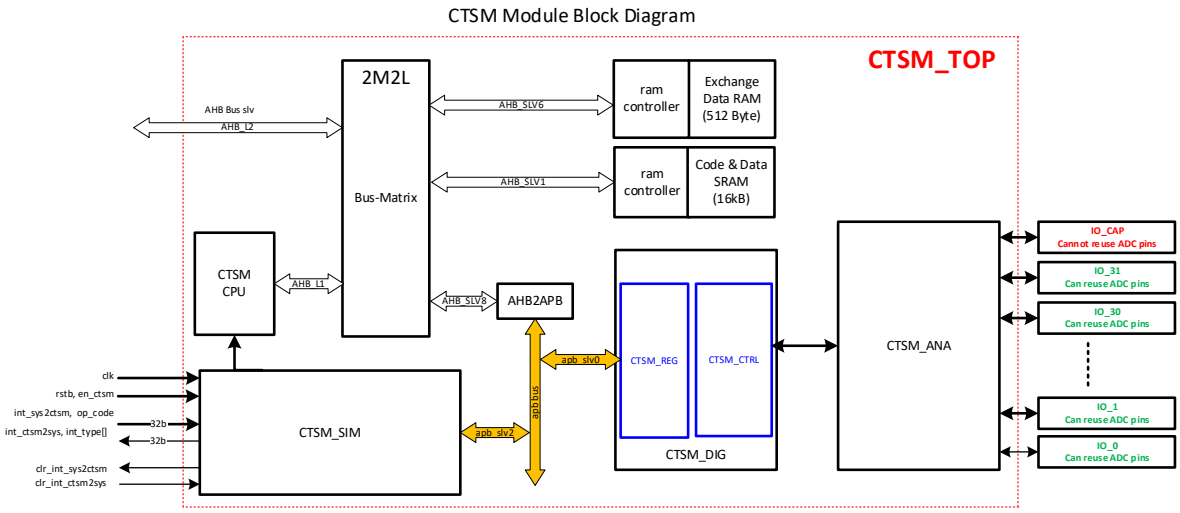
Table 61 LED register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CTRL	Res	Res	ANAON	[TRIM[3:0]				PIE	CIE	FIE	UIE	CDE	FDE	UDE	LEDEN	S8E	S7E	S6E	S5E	S4E	S3E	S2E	S2E	S0E	C7E	C6E	C5E	C4E	C3E	C2E	C1E	C0E
	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
0x04	CFG	IFN[7:0]								DTW[7:0]								PCS[15:0]															
	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
0x08	MODE	Res	Res	Res	Res	DM[2:0]				BFN[6:0]						RMD	DMD	BRTS[4:0]					DIMS[2:0]			Res	ALAST[6:0]						
	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
0x0C	BRG0	Res	Res	Res	BRT3[4:0]					Res	Res	Res	BRT2[4:0]					Res	Res	Res	BRT1[4:0]					Res	Res	Res	BRT0[4:0]				
	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
0x10	BRG1	Res	Res	Res	BRT7[4:0]					Res	Res	Res	BRT6[4:0]					Res	Res	Res	BRT5[4:0]					Res	Res	Res	BRT4[4:0]				
	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
0x14	BRG2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BRT8[4:0]				
	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
0x18	EG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UG
	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
0x1C	STATUS	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PIF	CIF	FIF	UIF
	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
0x20	DEBUG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ADONERR	BUFERR	FSM[2:0]			Res	Res	Res	Res	Res	Res	Res	Res	Res	FRMN[6:0]						
	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
0x40-0x23C	BUFF[0:127]																																
	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

25 Capacitance touch sensor module (CTSM)

25.1 Introduction

The capacitance touch sensor module provides a simple interface for adding capacitive sensing functionality to applications. Capacitive sensing technology can detect the presence of finger near an electrode separated by a dielectric such as glass or plastic. The XT32H0xx devices supports 32 channels capacitive touch sensing interface. The device enables the development of highly optimized touch-based applications with low latency and high accuracy.



* Notes:

- op_code[N:0] will be wire connected from main system to CTSM sub-system. CTSM CPU will read a read-only register to get the op_code[N:0].

- **ahb_slv1: 0x8002_00000 ~ x8002_7FFF**
- ahb_slv2: 0x8002_80000 ~ x8002_8FFF
- ahb_slv3: 0x8002_90000 ~ x8002_9FFF
- ahb_slv4: 0x8002_A0000 ~ x8002_AFFF
- ahb_slv5: 0x8002_B0000 ~ x8002_BFFF
- **ahb_slv6: 0x8002_C0000 ~ x8002_CFFF**
- ahb_slv7: 0x8002_D0000 ~ x8002_DFFF
- **ahb_slv8:**
  - ✓ **apb_slv0: 0x8002_E0000 ~ x8002_E7FF**
  - ✓ apb_slv1: 0x8002_E8000 ~ x8002_EFFF
  - ✓ **apb_slv2: 0x8002_F0000 ~ x8002_F7FF**
  - ✓ apb_slv3: 0x8002_F8000 ~ x8002_FFFF

Figure 65 CTSM block diagram

Table 62 Shared memory mapping

Memory	Start Address	Stop Address	Description
Code & data	0x8002_0000	0x8002_7FFF	Code and data area, 32k Bytes
Mailbox	0x8002_C000	0x8002_CFFF	Internal communication with main core, 4k Bytes

25.2 CTSM main features

- Capacitive Touch Sensor
  - self-capacitance measurement mode
  - mutual capacitance measurement mode
  - DC current measurement mode
- up to 32 Channels (all for self-capacitance method or up to 16 Rx channels and 16 Tx channels for mutual capacitance method)
  - Shadow channel
  - Combine channels
- Auto-correction
- Auto judgment
- Characteristic correction function to current control oscillator (Calibration mode)
- Offset current adjustment
- Sensor drive pulse frequency specification
- Measurement time specification
- Multi-frequency measurement
- Sensor drive pulse phase shift and edge diffusion
- Software trigger & external trigger
- Auto data transfer
  - Measurement channels configure
  - Measurement results
- Peripheral diagnosis

25.3 CTSM signal descriptions

The CTSM signals are shown in the following table.

Table 63 CTSM signal descriptions

Signal	Description	I/O
TSCAP	External capacitor connecting pin	I/O
TSN_0	Electrostatic capacitance measurement pin	I/O
TSN_1	Electrostatic capacitance measurement pin	I/O
TSN_2	Electrostatic capacitance measurement pin	I/O
TSN_3	Electrostatic capacitance measurement pin	I/O
TSN_4	Electrostatic capacitance measurement pin	I/O
TSN_5	Electrostatic capacitance measurement pin	I/O
TSN_6	Electrostatic capacitance measurement pin	I/O
TSN_7	Electrostatic capacitance measurement pin	I/O
TSN_8	Electrostatic capacitance measurement pin	I/O
TSN_9	Electrostatic capacitance measurement pin	I/O
TSN_10	Electrostatic capacitance measurement pin	I/O
TSN_11	Electrostatic capacitance measurement pin	I/O
TSN_12	Electrostatic capacitance measurement pin	I/O
TSN_13	Electrostatic capacitance measurement pin	I/O
TSN_14	Electrostatic capacitance measurement pin	I/O
TSN_15	Electrostatic capacitance measurement pin	I/O
TSN_16	Electrostatic capacitance measurement pin	I/O
TSN_17	Electrostatic capacitance measurement pin	I/O
TSN_18	Electrostatic capacitance measurement pin	I/O
TSN_19	Electrostatic capacitance measurement pin	I/O
TSN_20	Electrostatic capacitance measurement pin	I/O
TSN_21	Electrostatic capacitance measurement pin	I/O
TSN_22	Electrostatic capacitance measurement pin	I/O
TSN_23	Electrostatic capacitance measurement pin	I/O
TSN_24	Electrostatic capacitance measurement pin	I/O
TSN_25	Electrostatic capacitance measurement pin	I/O
TSN_26	Electrostatic capacitance measurement pin	I/O
TSN_27	Electrostatic capacitance measurement pin	I/O
TSN_28	Electrostatic capacitance measurement pin	I/O
TSN_29	Electrostatic capacitance measurement pin	I/O
TSN_30	Electrostatic capacitance measurement pin	I/O
TSN_31	Electrostatic capacitance measurement pin	I/O

To utilize the CTSM module for capacitive sensing, the relevant IO pads must first be configured to CFG_0 to act as the CTSM channel pins. Please refer to the Table XT32H0xxBxx IO assignment in Chapter 1 Port Mux (PADI) for the mapping between the IO pins and the CTSM channel pins.

25.4 CTSM functional description

XT32H0xx supports up to 32 sensing channels under self-capacitance method, or up to 16 RX channels and 16 TX channels under mutual-capacitance method.

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure the individual capacitance when a finger is nearby. Either single scan or multi-scan operations can be performed to detect the presence of a finger. In the mutual capacitance method, the capacitance between two complementary electrodes (transmit and receive pins) is measured.

The CTSM consists of various analog and digital blocks as shown in Figure 64. The CTSM measures the current flows through the switched capacitor by repeatedly charging and discharging CTSM channels. If a finger is in proximity, the capacitance as well as the current that flows through the corresponding CTSM channels changes. A clock is generated by supply the control current, which is proportional to the amount of current flowing through the switched capacitor, from the LDO circuit that generates TSCAP power supply to the current controlled oscillator. The counter in the CTSM measures the clock frequency that various depending on whether a finger is in proximity. The software subsequently read the counter value to determine whether a finger made contact or not.

Individual CTSM channel can be enabled or disabled by setting the relevant bits in the CHEN register. Each bit in the CHCTRL register corresponds to the CTSM channel that can be configured either as RX or TX channel.

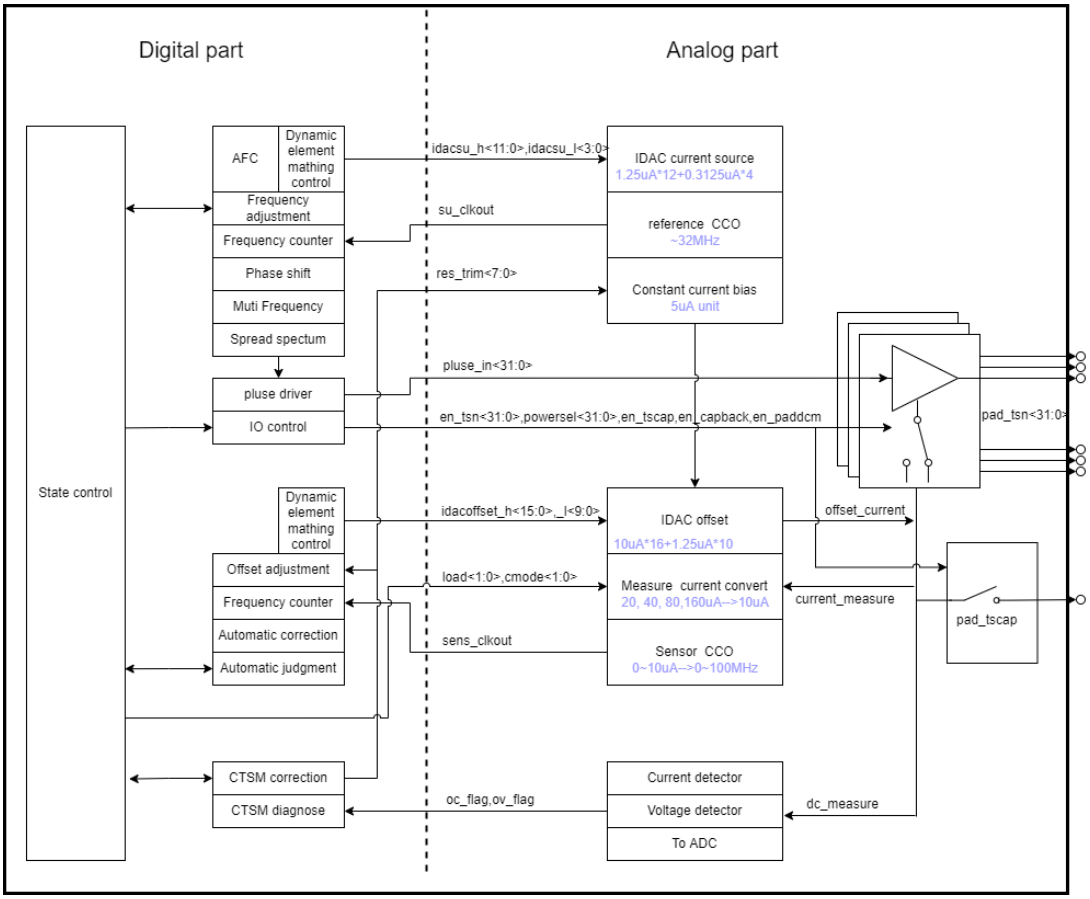


Figure 66 CTSM analog and digital interface diagram

25.4.1 Scanning

CTSM measurement flow is shown in the following diagram.

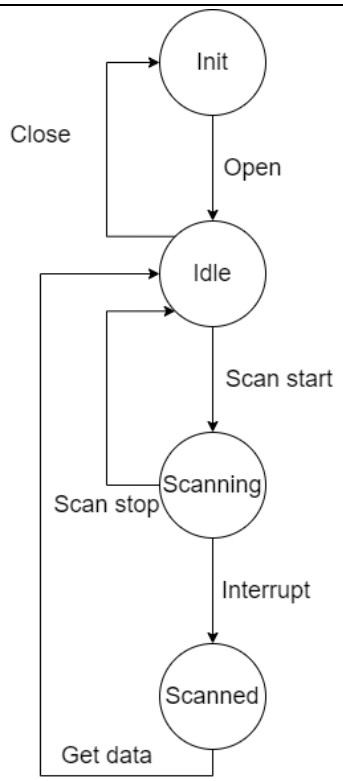


Figure 67 Scanning status

CTSM measurement begins by first starting up the CTSM regulator to setup the CTSM VDD voltage and reference bias. There are two different CTSM measurement mode: Capacitance switch measurement mode, and DC measurement mode. In the capacitance switch measurement mode, the drive pulse generates by CTSM switches the touch sensor IO and sense the current through the external capacitive charges. For DC measurement, no pulse is generated while the sensed current is sourced directly through the external or internal load.

When CTSM performs to measure touch sensor, the programming sequence show as bellowing steps:

1. Disable CTSM, write 0 to SYSCFG.CTSM_CTRL.CTSM_CTRL_EN[0];
2. Write DWA divider to CTSM_PSC.DWA_DIV[31:16] and write CTS_DIG divider to CTSM_PSC.DIG_DIV[15:0];
3. Enable CTSM, write 1 to SYSCFG.CTSM_CTRL.CTSM_CTRL_EN[0];
4. Config CTSM CAP IO, measurement channel IOs, and shield channel IO;
5. Calibrate sensor offset;
    a. Initialize CTSM;
        i. Set high resolution sensor pulse mode by writing 1 to CTSM_DCTRL. SDPSEL[10];
        ii. Set current range 20 uA by writing b’10 to CTSM_ACTRL.CMD[13:12]
        iii. Disable pseudorandom number mode by writing 0 to CTSM_DCTRL.PROFF[19]
        iv. Disable spread spectrum mode by writing 0 to CTSM_DCTRL.SOFF[18]
        v. Disable SUCLK SS mode to 0 by writing 0 to AD.SSCTRL[26:24]

vi. Disable tuning by writing 0 to OPT.TUNEN[2]

b. Enable interrupt by writing 1 to ICTL.IRQINTENH[31];

c. Config scan;

i. Write scan number to SCANNUM.SCANNUM[15:0];

ii. Set measurement period to 384us by writing counts to CTSM_TS.MSNUM[31:16];

iii. Set clock divider by writing divider to CTSM_DIV.BDIV[11:4];

iv. Enable scan channel by writing 1 to the channel index bit of CHEN and writing 0 to the channel index bit of CHCTRL

v. Configure shield channel by writing 1 to the channel index bit of CHCTRL

vi. Set shield channel output by writing 0 to PS. SHDPSEL[5:4]

vii. Set non measurement channel output by writing 0 to PS. NMPSEL[1:0]

viii. Set scan frequency pattern to one frequency by writing 1 to FRECFG.MCA0[0]

ix. Disable frequency judgment by writing by writing 0 to OPT.FREJFEN[1]

x. Set scan mode to self capacitance and multiple scan by writing CTSM_DCTRL.MD0[6] and CTSM_DCTRL.MD[7]

xi. Disable automatic judgment by writing 0 to OPT.AJFEN[8]

xii. Disable correction by writing 0 to OPT.CCOCFEN[0]

xiii. Enable interrupts by writing 0xFFF to INTEN

d. Judgment initial by writing 1 to SYSCFG.CTSM_INT.JINIT[12] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]

e. Sensor offset initial by writing 1 to SYSCFG.CTSM_INT.SOINIT[16] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]

f. Start normal scan by writing 1 to SYSCFG.CTSM_INT.MSTRT[1] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]

g. Scan all sensor offset, read measurement result from buffer that starts at MEASURE_BUFFER_ADDRESS and size is MEASURE_BUFFER_SIZE bytes.

h. Stop scan by writing 1 to SYSCFG.CTSM_INT.MSTOP[0] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]

6. CTSM initial

a. Set sensor pulse mode to CTSM_DCTRL. SDPSEL[10], 0 means random mode, 1 means high resolution mode;

b. If sensor pulse is random mode, should set the following parameters:

i. Set pseudorandom number mode to CTSM_DCTRL.PROFF[19], 0 means disable, 1 means enable;

if pseudorandom number mode is enabled, config the following parameters:

- pseudorandom generate cycle by writing CTSM_DCTRL.PHT[16:15]
- pseudorandom update cycle by writing CTSM_DCTRL.PHS[14:11]

ii. Set spread spectrum mode to CTSM_DCTRL.SOFF[18], 0 means disable, 1

means enable;

If spread spectrum is enabled, config the following parameter:

- Spread spectrum period by writing CTSM_DIV.SS_DIV[3:0]

iii. Set SUCLK SS mode to AD.SSCTRL[26:24],

- b’000 means disable,
- b’001 means SS frequency adjust 0x04,
- b’010 means SS frequency adjust 0x08,
- b’011 means SS frequency adjust 0x0C,
- b’100 means SS frequency adjust 0x10,
- b’101 means SS frequency adjust 0x14,
- b’110 means SS frequency adjust 0x18,
- b’111 means SS frequency adjust 0x1C,

If SUCLK SS is enabled, config the following parameter:

- period N+1 to AD.SSNUM[23:8]

c. Set current range to CTSM_ACTRL.CMD[13:12],

b’00 means 80uA,
b’01 means 40uA,
b’10 means 20uA and b’11 means 160uA.

d. Disable tuning by writing 0 to OPT.TUNEN[2]

e. Enable interrupt by writing 1 to ICTL.IRQINTENH[31];

f. Config scan;

i. Write scan number to SCANNUM.SCANNUM[15:0];

ii. Set measurement period to 384us by writing counts to CTSM_TS.MSNUM[31:16];

iii. Set clock divider by writing divider to CTSM_DIV.BDIV[11:4];

iv. Enable scan channel by writing 1 to the channel index bit of CHEN and writing transfer mode to the channel index bit of CHCTRL, 0 means receive, 1 means transmit;

v. Configure shield channel by writing 1 to the channel index bit of CHCTRL

vi. Set shield channel output to PS. SHDPSEL[5:4], b’00 means output 0, b’01 means output 1, b’10 means output pulse;

vii. Set non measurement channel output by writing 0 to PS. NMPSEL[1:0], b’00 means output 0, b’01 means output 1, b’10 means output pulse;

viii. Set scan frequency pattern to FRECFG.MCAx[3:0];

ix. Set frequency judgment mode to OPT.FREJFEN[1], b’0 means disable, b’1 means enable;

x. Set scan mode to CTSM_DCTRL.MD0[6], b’0 means self capacitance and b’1 means mutual capacitance, and CTSM_DCTRL.MD[7], b’0 means single scan, b’1 means multiple scan;

xi. Set automatic judgment to OPT.AJFEN[8], b’0 means disable, b’1 means enable;

If automatic judgment is enabled, the following parameter should be set:

- Judgment condition to AJCTRL.JC[21:20]

  2'b00: Judges as "touch" when there is one or more touch judgments

  2'b01: Judges as "touch" when there are two or more touch judgments

  2'b10: Judges as "touch" when there are three or more touch judgments

  2'b11: Judges as "touch" only when there are four touch judgments

xii. Set correction mode to OPT.CCOCFEN[0], b'0 means disable, b'1 means enable;

xiii. Enable interrupts by writing 0xFFF to INTEN

7. Judgment initial by writing 1 to SYSCFG.CTSM_INT.JINIT[12] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]
8. Sensor offset initial by writing 1 to SYSCFG.CTSM_INT.SOINIT[16] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]
9. Start normal scan by writing 1 to SYSCFG.CTSM_INT.MSTRT[1] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]
10. Wait interrupt to get measurement result and judgment result. Read measurement result from buffer that starts at MEASURE_BUFFER_ADDRESS and size is MEASURE_BUFFER_SIZE bytes. Read judgment result from buffer that starts at JUDGMEN_BUFFER_ADDRESS and size is JUDGMENT_BUFFER_SIZE.
11. Stop scan by writing 1 to SYSCFG.CTSM_INT.MSTOP[0] and writing 1 to SYSCFG.CTSM_INT.INTGEN[31]

Notes:

1. Buffer address and size list in following table.

MEASURE_BUFFER_ADDRESS is 0x8002C000
MEASURE_BUFFER_SIZE is 0x100 bytes
JUDGMENT_BUFFER_ADDRESS is 0x8002C1E0
JUDGMENT_BUFFER_SIZE is 0x40 bytes

Buffer	Address	Size (Bytes)
Measurement buffer	0x8002C000	0x100
Judgment buffer	0x8002C1E0	0x40

2. Buffer data format are shown in the following table.

Buffer Word Index	Byte 3	Byte 2	Byte 1	Byte 0
Measurement data[0]: 0	Second data [0]		Prime data [0]	
Measurement data[0]: 1	Second UC count[0]		Prime UC count[0]	
..	..	..	..	..

Buffer Word Index	Byte 3	Byte 2	Byte 1	Byte 0
Judgment data[1:0]	..	Judgment date [1]	..	Judgment data [0]
..	..	..	..	..

25.4.2 Correction

Calculate coefficient to correct measurement raw data. It is called when CTSM is open.

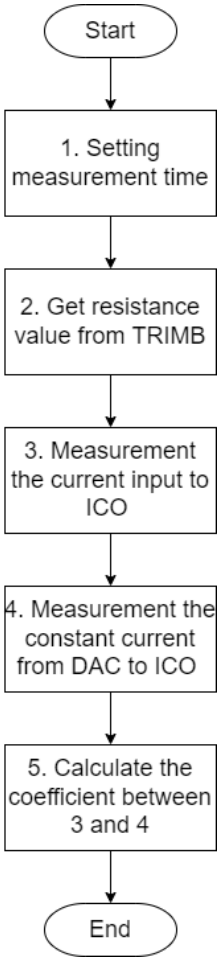


Figure 68 Correct process

When get data or tuning sensor offset, it may correct raw data and average corrected data.

25.4.3 Measurement

Measure current on selected channels.

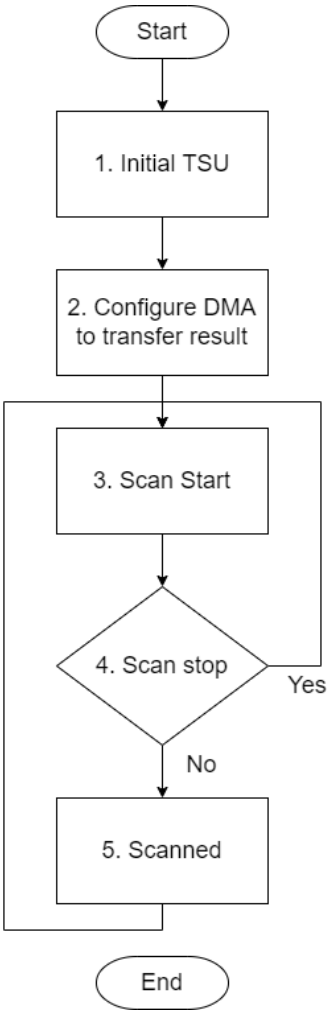


Figure 69 Measurement

25.4.4 Tuning sensor offset

After measurement is completed, sensor offset need to be tuning. It is also called when TSU is open.

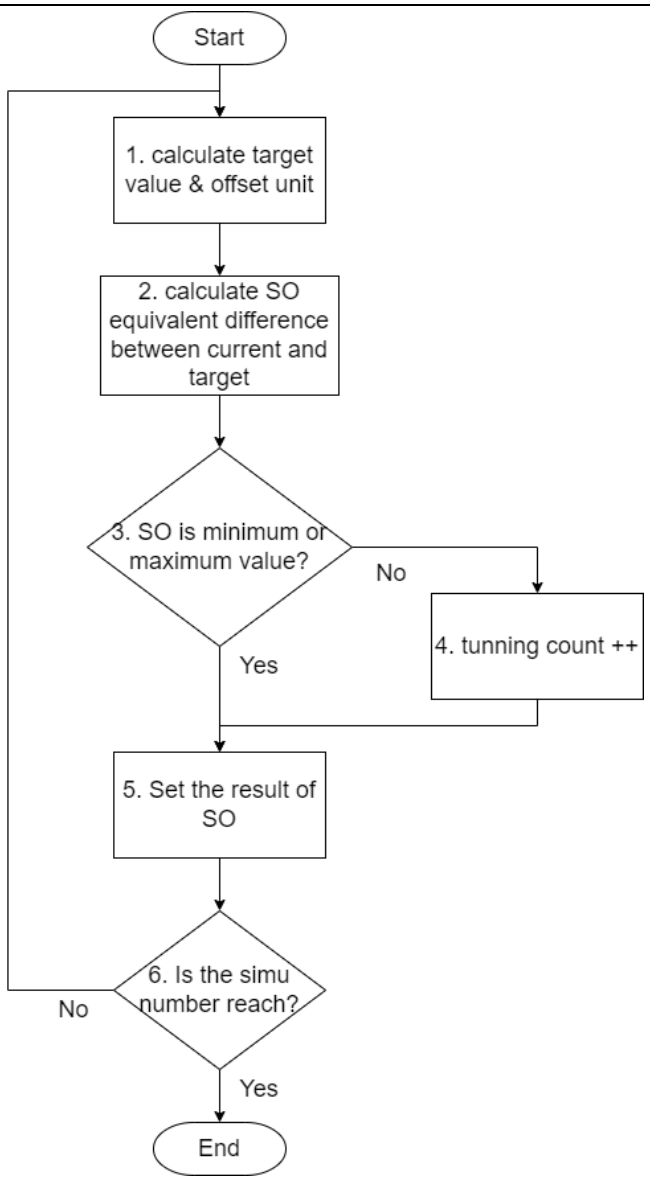


Figure 70 Tunning sensor offset

25.4.5 Auto correction table

When TSU is open and get data, update auto correct table for TSU is auto correction is enabled. Correction.DAC_value and Correction.ref_value is an item in table by writing ACT register.

25.4.6 Interrupts

Register setting request interrupt (WR)
Measurement result read request interrupt (RD)
Measurement end interrupt (CLP)

25.4.7 Register setting request

Update measurement channel settings (SO) got each channel measurement.

25.4.8 Measurement data read request

Request to read SCNT when a channel measurement is completed

25.4.9 Measurement end

Output when all enabled channels measurement are completed.

25.5 CTSM register

25.5.1 CTSM_INT_CTSM2HOST (interrupt request, used by CTSM system)

Base address: 0x8002_ F000
Address offset: 0x00
Type: Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	**INTGEN**: interrupt generate.	W
<30:11>	Reserved.	W
<10>	**TCORRCPLT**:	W
<9>	**CORRCPLT**:	W
<8>	**IDLE**: Idle	W
<7>	**ERROR**: error report	W
<6>	**JCLPT**: judgment completed	W
<5>	**SCLPT**: scan completed	W
<4>	**MCLPT**: measurement completed	W
<3>	**BUSY**: busy	W
<2>	**JRRD**: judgment result read request	W
<1>	**MARRD**: measurement average result read request	W
<0>	**MRRD**: measurement result read request	W

25.5.2 CTSM_INT_HOST2CTSM (interrupt status, used by CTSM system)

Base address: 0x8002_ F000
Address offset: 0x04
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31>	**INTGEN**: interrupt indicator, from host to CTSM	R
<30:15>	Reserved.	R
<16>	**SOINIT**: Sensor offset initial	R
<15>	**MSCAN**: measurement scan	R
<14>	**AFC**: AF calibration	R
<13>	**GETST**: get CTSM status	R
<12>	**JINIT**: judgment initial	R
<11:7>	Reserved.	R
<6>	**IDCSTRT**: internal current measurement start	R
<5>	**TCSTRT**: temperature correction start	R

Bit	Description	Type
<4>	**DIASTRT**: diagnosis start	R
<3>	**DCSTRT**: current measurement start	R
<2>	**CSTRT**: correction start	R
<1>	**MSTRT**: measurement start	R
<0>	**MSTOP**: measurement stop	R

25.5.3 CTSM_SYSCFG (used by CTSM system)

Base address: 0x8002_ F000
Address offset: 0x08
Type: Write
Reset value: 0x0000_0001

Bit	Description	Type
<31>	**INTCLR**: clear HOST interrupt request	W
<30:16>	Reserved.	W
<15:8>	**AFCRDIV[8:0]**: AFC reference clock divider	W
<7:1>	Reserved.	W
<0>	**RST**: CTSM reset	W

25.5.4 CTSM_PSC (used by CTSM system)

Base address: 0x8002_ F000
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_0002

Bit	Description	Type
<31:16>	**DWA_DIV[15:0]**: DWA divider 16’h00: divide 1 16’h01: divide 1 16’h02: divide 2 … 16’hxx: divide n …	R/W
<15:0>	**DIG_DIV[15:0]**: CTS_DIG divider 16’h00: divide 1 16’h01: divide 1 16’h02: divide 2 … 16’hxx: divide n …	R/W

25.5.5 CTSM_DCTRL

Base address: 0x8002_ F010
Address offset: 0x00
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:28>	Reserved	R/W
<27>	**CDEIE**: CTS DWA error interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<26>	**ADEIE**: AFC DWA error interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<25>	**AEIE**: AFC Calibration error interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<24>	**ANIE**: AFC Calibration completed interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<23>	**ALIE**: AFC Calibration completed in loop interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<22>	**SCOIE**: SC overflow interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<21>	**UCOIE**: UC overflow interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<20>	**MDIE**: Measurement completed interrupt enable. 1'b0: disabled 1'b1: enabled	R/W
<19>	**PROFF**: Pseudo-random number off 1'b0: Pseudo-random number is enabled. 1'b1: Pseudo-random number is disabled.	R/W
<18>	**SOFF**: Spread-spectrum off 1'b0: Spread-spectrum is enabled. 1'b1: Spread-spectrum is disabled.	R/W
<17>	Reserved.	R/W
<16:15>	**PHT[1:0]**: Pseudo-random number generation cycle	R/W

Bit	Description	Type
	2’b00: 255 cycles 2’b01: 63 cycles 2’b10: 31 cycles 2’b11: 3 cycles	
<14:11>	**PHS[3:0]**: Pseudo-random number update cycle. Set the shift period of the linear feedback shift register (LFSR). Output per (n+1) cycles.	R/W
<10>	**SDPSEL**: Sensor driver pulse select 1’b0: Random pulse mode 1’b1: High resolution pulse mode	R/W
~~<9:8>~~	~~**CLKSEL[1:0]**: Monitor clock select~~ ~~2’b00: Low is output~~ ~~2’b01: Sensor CCO clock divided by 4 (16)~~ ~~2’b10: Setting prohibited~~ ~~2’b11: SUCLK divided by 4 (16)~~	~~R/W~~
<7>	**MD**: Measurement mode select 1 1’b0: Self-capacitance method 1’b1: Mutual capacitance method	R/W
<6>	**MD0**: Measurement mode select 0 (just for external core) 1’b0: Single scan mode 1’b1: multi-scan mode	R/W
~~<5>~~	~~**MSS**: Measurement operation start enable select~~ ~~1’b0: Software enable~~ ~~1’b1: Hardware enable~~	~~R/W~~
<4>	Reserved.	R/W
<3>	**CTSDWAEN**: CTS DWA enable. 1’b0: disabled 1’b1: enabled	R/W
<2>	**AFCDWAEN**: AFC DWA enable. 1’b0: disabled 1’b1: enabled	R/W
<1>	**AFCEN**: AFC enable. 1’b0: disabled 1’b1: enabled	R/W
<0>	**STRT**: Measurement operation start 1’b0: Measurement stop 1’b1: Measurement start	R/W

25.5.6 CTSM_TS

Base address: 0x8002_ F010
Address offset: 0x04

Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**MSNUM[15:0]** : Measurement Period Setting. Measurement is performed for (n + 1) cycles of operation clock.	R/W
<15:0>	**WTNUM[15:0]**: Sensor stabilization wait time setting The stabilization wait time is (n + 1) cycles of operation clock.	R/W

25.5.7 CTSM_DIV

Base address: 0x8002_ F010

Address offset: 0x08

Type: Read/Write

Reset value: 0x0000_0000

Bit	Description	Type
<31:12>	Reserved	R/W
<11:4>	**BDIV[7:0]**: Base clock setting - Random pulse mode (SDPSEL = 0) When the set value is n, the base clock frequency is the operating clock divided by 2(n + 1). - High resolution pulse mode (SDPSEL = 1) When the set value is n, the base clock frequency is SUCLK divided by n + 1.	R/W
<3:0>	**SSDIV[3:0]**: Spread Spectrum Sampling Period Control. When the set value is n, the spread spectrum sampling period is SUCLK divided by n + 1.	R/W

25.5.8 PS

Base address: 0x8002_ F010
Address offset: 0x0C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:29>	Reserved	R/W
<28:24>	**TXCH[4:0]**: The index of transmit channel to be measured	R/W
<23:21>	Reserved	R/W
<20:16>	**RXCH[4:0]**: The index of receive channel to be measured	R/W
<15:10>	Reserved	R/W
<9>	**SHD**: Shield pin enable 1'b0: disabled 1'b1: enabled	R/W
<8>	**TXINV**: Transmit Pin Inverting Output. - Self-capacitance method (MD = 0) 1'b0: Outputs a signal with the **same** phase as the receive pin 1'b1: Outputs a signal with the **opposite** phase to the receive pin - Mutual capacitance method (MD = 1) 1'b0: Outputs a signal with the **same** phase as the receive pin at the first time, and outputs the signal with the **opposite** phase to the receive pin 1'b1: Outputs a signal with the **opposite** phase to the receive pin at the first time, and outputs the signal with the **same** phase as the receive pin	R/W
<7:6>	Reserved	R/W
<5:4>	**SHDPSEL[1:0]**: Shield pin state 2'b00: 0 (VSS) 2'b01: 1 (VDD) 2'b10: pulse Others: reserved	R/W
<3:2>	Reserved	R/W
<1:0>	**NMPSEL[1:0]**: Non-measure pin state 2'b00: 0(VSS) 2'b01: 1 (VDD) 2'b10: pulse Others: reserved	R/W

25.5.9 DF (no-used)

Base address: 0x8002_ F010
Address offset: 0x10
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	Reserved	R/W
<23:16>	**ADJRES[7:0]**: Calibration result	R/W
<15:11>	Reserved.	R/W
<10>	**ANIF**: AFC completed interrupt flag 1'b0: No interrupt. (write 0 to clear interrupt) 1'b1: interrupt.	R/W
<9>	**ALIF**: AFC completed in loop interrupt flag 1'b0: No interrupt. (write 0 to clear interrupt) 1'b1: interrupt.	R/W
<8>	**MDIF**: Measurement completed interrupt flag 1'b0: No interrupt. (write 0 to clear interrupt) 1'b1: interrupt.	R/W
<7:5>	Reserved.	R/W
<4>	**CDEIF**: CTS DWA error interrupt flag 1'b0: No interrupt. (write 0 to clear interrupt) 1'b1: interrupt.	R/W
<3>	**ADEIF**: AFC DWA error interrupt flag 1'b0: No interrupt. (write 0 to clear interrupt) 1'b1: interrupt.	R/W
<2>	**AEIF**: AFC error interrupt flag 1'b0: No interrupt. (write 0 to clear interrupt) 1'b1: interrupt.	R/W
<1>	**UCOVF**: Sensor unit clock counter overflow flag 1'b0: Not overflow (write 0 to clear) 1'b1: Overflow (hardware writes 1)	R/W
<0>	**SCOVF**: Sensor counter overflow flag 1'b0: Not overflow (write 0 to clear) 1'b1: Overflow (hardware writes 1)	R/W

25.5.10 SR

Base address: 0x8002_ F010
Address offset: 0x14
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**CST[7:0]**: Current Scan status.	R
<23:16>	**AST[7:0]**: Action status.	R
<15:8>	**MST[7:0]**: Measure status.	R
<7:6>	**MFC[1:0]**: Multi-clock counter. Only write 00. Just for external core 2'b00: SUCLK0 2'b01: SUCLK1 2'b10: SUCLK2 2'b11: SUCLK3	R
<5:4>	Reserved	R/W
<3:0>	**STC[2:0]**: Measurement status 4'b0000: Status 0. (idle) 4'b0001: Status 1. (wait) 4'b0010: Status 2. (measure) 4'b0011: Status 3. (sync) 4'b0100: Status 4. (read) 4'b0101: Status 5. (wait) 4'b0110: Status 6. (measure) 4'b0111: Status 7. (sync) 4'b1000: Status 8. (read)	R

25.5.11 CNT (no-used)

Base address: 0x8002_ F010
Address offset: 0x18
Type: Read
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**UC[15:0]**: Sensor unit counter. These bits indicate the count value of the clock (twice the frequency of SUCLK) output from the CCO of SUCLK. When an overflow occurs, these bits become FFFFh.	R
<15:0>	**SC[15:0]**: Sensor counter. These bits indicate the measurement result. When an overflow occurs, these bits become FFFFh.	R

.

25.5.12 FRECFG

Base address: 0x8002_ F010
Address offset: 0x1C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:4>	Reserved	R/W
<3>	**MCA3**: Muti clock 3 enable 1'b0: Disabled 1'b1: Enabled	R/W
<2>	**MCA2**: Muti clock 2 enable 1'b0: Disabled 1'b1: Enabled	R/W
<1>	**MCA1**: Muti clock 1 enable 1'b0: Disabled 1'b1: Enabled	R/W
<0>	**MCA0**: Muti clock 0 enable 1'b0: Disabled 1'b1: Enabled	R/W

25.5.13 CHEN

Base address: 0x8002_ F010
Address offset: 0x20
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**CHEN[x]**: Channel x enable. X is 31 to 0. 1'b0: Disabled 1'b1: Enabled	R/W

25.5.14 CHCTRL

Base address: 0x8002_ F010
Address offset: 0x24
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**CHCTRL[x]**: Channel x transmit/receive control. X is 31 to 0. 1'b0: Receive 1'b1: Transmit	R/W

25.5.15 ACTRL

Base address: 0x8002_ F010
Address offset: 0x40
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:17>	Reserved	R/W
<16>	**SENSDB**: Double the touch sense current. 1'b0: the touch sense current for sense cco is unit 1'b1: the touch sense current for sense cco is double R/W	R/W
<15:14>	**LOAD[1:0]**: Measurement load control 2'b00: Constant current load mode (for normal measurement) 2'b01: No load mode 2'b10: Constant current load mode (for calibration) 2'b11: Resistive load mode (for calibration)	R/W
<13:12>	**CMD[1:0]**: selects the measurement current range of the measurement power supply 2'b00: 80uA 2'b01: 40uA 2'b10: 20uA 2'b11: 160uA	R/W
<11>	**CCOBS**: CCO Bias current Select. 1'b0: idac bias for reference cco 1'b1: idac bias for sense cco	R/W
<10:9>	**OCTUN[1:0]**: selects the load resistance in resistive load mode 2'b00: 10k ohm 2'b01: 20k ohm 2'b10: 40k ohm 2'b11: 5k ohm	R/W
<8>	**ANAE**: cts analog module enable 1'b0: disable all ctsm circuits, 1'b1: enable	R/W
<7>	**DTE**: enable over current and over voltage detector 1'b0: disable 1'b1: enable	R/W
<6>	**CCMBE**: enable charge current measure block 1'b0: disable 1'b1: enable	R/W
<5>	**CCOCLKEN**: CCOCLK Enable 1'b0: disable	R/W

Bit	Description	Type
	1’b1: enable	
<4>	**SUCLKEN**: SUCLK Enable 1’b0: SUCLK is stopped. 1’b1: SUCLK is oscillated.	R/W
<3>	**TON**: turn on the tscap io switcher 1’b0: off 1’b1: on	R/W
<2>	**TDCON**: turn on the dcback switcher in tscap io 1’b0: off 1’b1: on	R/W
<1>	**DCME**: DC measure Enable 1’b0: disable 1’b1: enable	R/W
<0>	**PON**: measurement power supply enable 1’b0: Power off 1’b1: Power on	R/W

25.5.16 ACFG (no-used)

Base address: 0x8002_ F010
Address offset: 0x44
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:9>	Reserved	R/W
<8>	**FCLR**: ov_flag and oc_flag clear ~~(ICOMP0 and ICOMP1 reset)~~ 1’b0: No effect. 1’b1: Clear ~~(ICOMP0 and ICOMP1).~~	R/W
<7:0>	**RTRIM[7:0]**: These bits are used to adjust the resistance value of the internal reference resistor	R/W

25.5.17 PE (no-used)

Base address: 0x8002_ F010
Address offset: 0x48
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**TXE[31:0]**: enable tsn<x> pin for measure for channel enable 1’b0: off	R/W

Bit	Description	Type
	1’b1: on	

25.5.18 AD

Base address: 0x8002_ F010
Address offset: 0x4C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:28>	Reserved	R/W
<28>	**CTSDWABYS**: CTS DWA bypass enable 1’b0: not bypass 1’b1: bypass	R/W
<27>	Reserved.	R/W
<26:24>	**SSCTRL[2:0]**: SUCLK spread-spectrum control 3’b000: **SUADJ[7:0]**+ 00h 3’b001: **SUADJ[7:0]**+ 04h 3’b010: **SUADJ[7:0]**+ 08h 3’b011: **SUADJ[7:0]**+ 0Ch 3’b100: **SUADJ[7:0]**+ 10h 3’b101: **SUADJ[7:0]**+ 14h 3’b110: **SUADJ[7:0]**+ 18h 3’b111: **SUADJ[7:0]**+ 1Ch	R/W
<23:8>	**SSNUM[15:0]**: SUCLK spread-spectrum period number. (n+1) cycles	R/W
<7:0>	**SO[7:0]**: Sensor offset adjustment. These bits adjust the input offset current of sensor CCO. Idacoffset_h[15:0]: current dac for offset high current inject, 16 cells with 10uA each idacoffset_l[9:0]: current dac for offset low current inject, 10 cells with 1.25uA each	R/W

25.5.19 AF (no-used)

Base address: 0x8002_ F010
Address offset: 0x50
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:2>	Reserved	R/W
<1>	**OVF**: Over-voltage detection flag 1'b0: Normal. 1'b1: Over-voltage is detected.	R/W
<0>	**OCF**: Over-current detection flag 1'b0: Normal. 1'b1: Over-current is detected.	R/W

25.5.20 AFCCTRL

Base address: 0x8002_ F010
Address offset: 0x54
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	Reserved	R/W
<29:16>	**RFCNT[13:0]**: AFC reference clock period. The value is 64, when reference clock is 4MHz; The value is 1536, when reference clock is 96MHz.	R/W
<15:8>	**DIFFCNT[7:0]**: Calibration reference. Default value is 16.	R/W
<7:4>	Reserved.	R/W
<3>	**AFCDWABYS**: AFC DWA bypass. 1'b0: no bypass 1'b1: bypass	R/W
<2>	**AFCBYS**: AFC bypass. 1'b0: no bypass 1'b1: bypass	R/W
<1:0>	Reserved.	R/W

25.5.21 AFCCFG

Base address: 0x8002_ F010
Address offset: 0x58
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31>	Reserved	R/W
<30>	**CBCLKS**: Reference clock. 1'b0: SUCLK 1'b1: CCOCLK	R/W
<29:28>	Reserved	R/W
<27:8>	**STDCNT[19:0]**: Calibration count. (no used)	R/W
<7:0>	**SUADJ[7:0]**: SUCLK frequency. Set the initial value of the frequency of SUCLK. (no used)	R/W

25.5.22 SUCLK0

Base address: 0x8002_ F010
Address offset: 0x60
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**SUADJ3[7:0]**: SUCLK3 Frequency Setting. Set the initial value of the frequency of SUCLK3.	R/W
<23:16>	**SUADJ2[7:0]**: SUCLK2 Frequency Setting. Set the initial value of the frequency of SUCLK2.	R/W
<15:8>	**SUADJ1[7:0]**: SUCLK1 Frequency Setting. Set the initial value of the frequency of SUCLK1.	R/W
<7:0>	**SUADJ0[7:0]**: SUCLK0 Frequency Setting. Set the initial value of the frequency of SUCLK0.	R/W

25.5.23 SUCLK1

Base address: 0x8002_ F010
Address offset: 0x64
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**FREQ0[7:0]**: SUCLK0 Frequency Setting. Set the initial value of the frequency of SUCLK0.	R/W

25.5.24 SUCLK2

Base address: 0x8002_ F010
Address offset: 0x68
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**FREQ1[7:0]**: SUCLK2 Frequency Setting. Set the initial value of the frequency of SUCLK2.	R/W

25.5.25 SUCLK3

Base address: 0x8002_ F010
Address offset: 0x6C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**FREQ2[7:0]**: SUCLK2 Frequency Setting. Set the initial value of the frequency of SUCLK2.	R/W

25.5.26 SUCLK4

Base address: 0x8002_ F010
Address offset: 0x70
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:8>	Reserved	R/W
<7:0>	**FREQ3[7:0]**: SUCLK3 Frequency Setting. Set the initial value of the frequency of SUCLK3.	R/W

25.5.27 TRIM0

Base address: 0x8002_ F010
Address offset: 0x74
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**TRESULT4[7:0]**: AFC adjust time (default 128us)	R/W
<23:16>	**SUADJD[7:0]**: SUCLK Frequency Adjustment. These bits are used to adjust the SUCLK frequency in random pulse mode (CTSUCRA.SDPSEL = 0). The value is written so that the frequency of SUCLK becomes about 32 MHz.	R/W
<15:8>	**DACTRIM[7:0]**: Offset Current Adjustment. These bits are used to adjust the coefficient of the current DAC for the current measurement oscillator.	R/W
<7:0>	**RTRIM[7:0]**: Reference Resistor Adjust Input. These bits are used to adjust the resistance value of the internal reference resistor. The value is written so that the specified current value is reached at 25°C.	R/W

25.5.28 TRIM1

Base address: 0x8002_ F010
Address offset: 0x78
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	**TRESULT3[7:0]**: Test Result 3 Storage. These bits stores the coefficient of variation when the load resistance for the LDO is 60 kΩ.	R/W
<23:16>	**TRESULT2[7:0]**: Test Result 2 Storage. These bits stores the coefficient of variation when the load resistance for the LDO is 30 kΩ.	R/W
<15:8>	**TRESULT1[7:0]**: Test Result 1 Storage. These bits stores the coefficient of	R/W

Bit	Description	Type
	variation when the load resistance for the LDO is 15 kΩ.	
<7:0>	**TRESULT0[7:0]**: Test Result 0 Storage. These bits stores the coefficient of variation when the load resistance for the LDO is 7.5 kΩ.	R/W

25.5.29 OPT

Base address: 0x8002_ F010
Address offset: 0x7C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:24>	Reserved	R/W
<23:20>	**AWIN[3:0]**: the number of the moving average window. The windows size is 2^n. Reset value: 0	R/W
<19:16>	**SCACTB[3:0]**: Sensor Counter Automatic Correction Table Number Setting. Set the number of the sensor counter automatic correction table to be accessed. The range of values is 0 to 11. When a value is written to the CTSUSCNTACT register, the value of these bits is automatically incremented.	R/W
<15:12>	Reserved	R/W
<11:9>	**SYSCLK[2:0]**: CTSU system clock setting. 3'b000: 1MHz 3'b001: 2MHz 3'b010: 4MHz 3'b011: 8MHz 3'b100: 16MHz 3'b101: 32MHz 3'b110: 64MHz other: Reserved	
<8>	**AJFEN**: Automatic Judgment Function Enable. 1'b0: Automatic judgment function is disabled. 1'b1: Automatic judgment function is enabled.	R/W
<7,6>	Reserved	R/W
<5>	**MTUCFEN**: Mutual Capacitance Calculation Enable. 1'b0: Do not subtract the first measurement data from the second measurement data 1'b1: Subtract the first measurement data from the second measurement data	R/W
<4>	**DTCLESS**: Data Transfer Request Disable. 1'b0: Data transfer request is enabled. 1'b1: Data transfer request is disabled.	R/W
<3>	Reserved	R/W

Bit	Description	Type
<2>	**TUNEN**: SO tuning enable 1'b0: SO tuning is disabled. 1'b1: SO tuning is enabled.	R/W
<1>	**FREJEN**: Frequency judgment enable 1'b0: Frequency judgment is disabled. 1'b1: Frequency judgment is enabled.	R/W
<0>	**CCOCFEN**: CCO Characteristics Correction Function Enable. 1'b0: CCO characteristics correction function is disabled. 1'b1: CCO characteristics correction function is enabled.	R/W

25.5.30 SCANNUM

Base address: 0x8002_ F010
Address offset: 0x80
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**SCANCNT[15:0]**: Scan counter	R/W
<15:0>	**SCANNUM[15:0]**: Continuous scan number. the number of measurement is (n+1).	R/W

25.5.31 AJCTRL

Base address: 0x8002_ F010
Address offset: 0x84
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:28>	Reserved	R/W
<27:24>	**AJMMAT[3:0]**: Measured Value Moving Average Number Setting. These bits specify the number of moving averages of the measured value. If the set value is n, the smoothing factor is ½n. The setting range is 0000b to 1011b.	R/W
<23:22>	Reserved	R/W
<21:20>	**JC[1:0]**: Judgment Condition Setting. 2'b00: Judges as "touch" when there is one or more touch judgments 2'b01: Judges as "touch" when there are two or more touch judgments 2'b10: Judges as "touch" when there are three or more touch judgments 2'b11: Judges as "touch" only when there are four touch judgments	R/W
<19:17>	Reserved	R/W
<16>	Reserved.	R/W

Bit	Description	Type
<15:8>	**THOT[7:0]**: Touch Judgment Criterion Setting. These bits specify the number of consecutive "touch" detections before a "touch" is reported. If the set value is m, the AJRSLT.TJRn flag becomes 1 when "touch" is detected m + 1 times in a row.	R/W
<7:0>	**TLOT[7:0]**: Non-Touch Judgment Criterion Setting. These bits specify the number of consecutive "non-touch" detections before a "non-touch" is reported. If the set value is m, the AJRSLT.TJRn flag becomes 0 when "non-touch" is detected m + 1 times in a row.	R/W

25.5.32 AJTHR

Base address: 0x8002_ F010
Address offset: 0x88
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:16>	**AJTHH[15:0]**: Upper Threshold Setting. Set the touch judgment threshold as a relative value from the baseline. The value is a signed integer between –32768 and 32767.	R/W
<15:0>	**AJTHL[15:0]**: Lower Threshold Setting. Set the non-touch judgment threshold as a relative value from the baseline. The value is a signed integer between –32768 and 32767.	R/W

25.5.33 MARSLT (no-used) (INTEN for CTSMtoHost interrupts)

Base address: 0x8002_ F010
Address offset: 0x8C
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:5>	**AJMMR[26:0]**: Moving Average Result. These bits store the moving average calculation result for the measured value. The value is a fixed-point number with 16 bits for the integer part and 11 bits for the decimal part.	R/W
<4>	Reserved	R/W
<3:0>	**AJMMATI[3:0]**: Moving Average Count. These bits indicate the number of moving averages for the current measurement. Set these bits to 0000b at the first measurement.	R/W

25.5.34 CTSM_AJRSLT (no-used)

Base address: 0x8002_ F010
Address offset: 0x90
Type: Read/Write
Reset value: 0x0000_0000

Bit	Description	Type
<31:0>	**AJRCHx**: Auto judgment result.	R/W

25.5.35 CTSM register map

The base address of the CTSM registers is 0x8002_F010.
The following table gives the CTSM register map and reset values.

Table 64 TSU register map and reset values

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DCTRL	Res	Res	Res	Res	CDIE	ADEIE	AEIE	ANIE	ALIE	SCOIE	UCOIE	MDIE	PROFF	SOFF	Res		PHT[1:0]				PHS[3:0]	SDPSEL		CLKSEL[1:0]	MD	MD0	MSS	Res	CTSDWAEN	AFCDWAEN	AFCEN	STRT
	Reset value	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
0x04	TS	MSNUM[15:0]																WTNUM[15:0]															
	Reset value	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
0x08	DIV	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BDIV[7:0]								SSDIV[3:0]			
	Reset value	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	1
0x0C	PS	Res	Res	Res	TXCH[4:0]					Res	Res	Res	RXCH[4:0]					Res	Res	Res	Res	Res	Res	SHD	TXINV	Res	Res	SHPSEL[1:0]		Res	Res	NMPSEL[1:0]	
	Reset value	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	1	0
0x10	DF	Res	Res	Res	Res	Res	Res	Res	Res	ADJRES[7:0]								Res	Res	Res	Res	Res	ANIF	ALIF	MDIF	Res	Res	Res	CDEIF	ADEIF	AEIF	UCOVF	SCOVF
	Reset value	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
0x14	SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MFC[1:0]		Res	Res	Res	Res	Res			STC[2:0]
	Reset value	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
0x18	CNT	UC[15:0]																SC[15:0]															
	Reset value	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
0x1C	FRECFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MCA3	MCA2	MCA1	MCA0
	Reset value	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
0x20	CHEN	CH31_EN	CH30_EN	CH29_EN	CH28_EN	CH27_EN	CH26_EN	CH25_EN	CH24_EN	CH23_EN	CH22_EN	CH21_EN	CH20_EN	CH19_EN	CH18_EN	CH17_EN	CH16_EN	CH15_EN	CH14_EN	CH13_EN	CH12_EN	CH11_EN	CH10_EN	CH9_EN	CH8_EN	CH7_EN	CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN
	Reset value	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
0x24	CHCTRL	CH31_CTRL	CH30_CTRL	CH29_CTRL	CH28_CTRL	CH27_CTRL	CH26_CTRL	CH25_CTRL	CH24_CTRL	CH23_CTRL	CH22_CTRL	CH21_CTRL	CH20_CTRL	CH19_CTRL	CH18_CTRL	CH17_CTRL	CH16_CTRL	CH15_CTRL	CH14_CTRL	CH13_CTRL	CH12_CTRL	CH11_CTRL	CH10_CTRL	CH9_CTRL	CH8_CTRL	CH7_CTRL	CH6_CTRL	CH5_CTRL	CH4_CTRL	CH3_CTRL	CH2_CTRL	CH1_CTRL	CH0_CTRL
	Reset value	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
0x40	ACTRL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LOAD[1:0]		CMD[1:0]		CCOBS	OCTUN[1:0]		ANAE	DTE	CCMBE	CCOCLKEN	SUCLKEN	TON	TDCON	DCME	PON

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value	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
0x44	ACFG	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FCLR	RTRIM[7:0]							
	Reset value	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
0x48	PE	TXE[31:0]																															
	Reset value	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
0x4C	AD	Res	Res	Res	Res	Res	CTSDWABYS	SSCTRL[1:0]		SSNUM[15:0]																SO[7:0]							
	Reset value	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
0x50	AF	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OVF	OCF
	Reset value	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
0x54	AFCCTRL	Res	Res	RFCNT[13:0]														DIFFCNT[7:0]								Res	Res	Res	Res	AFCDWABYS	AFCBYS	Res	Res
	Reset value	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
0x58	AFCCFG	Res	CBCLKS	Res	Res	STDCNT[19:0]																				SUADJ[7:0]							
	Reset value	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
0x60	SUCLK0	SUADJ3[7:0]								SUADJ2[7:0]								SUADJ1[7:0]								SUADJ0[7:0]							
	Reset value	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
0x64	SUCLK1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STDCNT0[19:0]																			
	Reset value	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
0x68	SUCLK2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STDCNT1[19:0]																			
	Reset value	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
0x6C	SUCLK3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STDCNT2[19:0]																			
	Reset value	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
0x70	SUCLK4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	STDCNT3[19:0]																			
	Reset value	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
0x74	TRIM0	TRESULT4[7:0]								SUADJD[7:0]								DACTRIM[7:0]								RTRIM[7:0]							
	Reset value	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
0x78	TRIM1	TRESULT3[7:0]								TRESULT2[7:0]								TRESULT1[7:0]								TRESULT0[7:0]							
	Reset value	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
0x7C	OPT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	AWIN	SCACTB[3:0]				Res	Res	Res	Res	SYSCLK[2:0]			AJFEN	Res	Res	MTUCFEN	DTCLESS	Res	TUNEN	FREJEN	CCOCFEN
	Reset value	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

Offset	Register name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x80	SCANNUM	SCANCNT[15:0]																SCANNUM[15:0]															
	Reset value	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
0x84	AJCTRL	Res	Res	Res	Res				AJMMAT[3:0]	Res	Res		JC[1:0]	Res	Res	Res	Res	THOT[7:0]								TLOT[7:0]							
	Reset value	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
0x88	AJTHR	AJTHH[15:0]																AJTHL[15:0]															
	Reset value	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
0x8C	MARSLT	AJMMR[26:11]																											Res	AJMMATI[3:0]			
	Reset value	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
0x90	AJRSLT	AJRCH31	AJRCH30	AJRCH29	AJRCH28	AJRCH27	AJRCH26	AJRCH25	AJRCH24	AJRCH23	AJRCH22	AJRCH21	AJRCH20	AJRCH19	AJRCH18	AJRCH17	AJRCH16	AJRCH15	AJRCH14	AJRCH13	AJRCH12	AJRCH11	AJRCH10	AJRCH9	AJRCH8	AJRCH7	AJRCH6	AJRCH5	AJRCH4	AJRCH3	AJRCH2	AJRCH1	AJRCH0
	Reset value	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