GigaDevice Semiconductor Inc.

GD32A50x
Arm® Cortex®-M33 32-bit MCU

For GD32A503xx

User Manual

Revision 1.8

(Aug. 2025)

Table of Contents

Table of Contents **2**

List of Figures **16**

List of Tables **22**

1. System and memory architecture **25**

1.1. Arm® Cortex®-M33 processor **25**

1.2. System architecture **26**

1.3. Memory map **28**

1.3.1. On-chip SRAM memory 32

1.3.2. On-chip Flash memory 34

1.4. Boot configuration **34**

1.5. System configuration controller **34**

1.6. System configuration registers **36**

1.6.1. System configuration register 0 (SYSCFG_CFG0) 36

1.6.2. System configuration register 1 (SYSCFG_CFG1) 36

1.6.3. EXTI sources selection register 0 (SYSCFG_EXTISS0) 37

1.6.4. EXTI sources selection register 1 (SYSCFG_EXTISS1) 38

1.6.5. EXTI sources selection register 2 (SYSCFG_EXTISS2) 40

1.6.6. EXTI sources selection register 3 (SYSCFG_EXTISS3) 41

1.6.7. System configuration register 2 (SYSCFG_CFG2) 42

1.6.8. System status register (SYSCFG_STAT) 43

1.6.9. System configuration register 3 (SYSCFG_CFG3) 44

1.6.10. TIMER input source select register (SYSCFG_TIMERINSEL) 45

1.7. Device electronic signature **47**

1.7.1. Memory density information 47

1.7.2. Unique device ID (96 bits) 48

2. Flash memory controller (FMC) **50**

2.1. Overview **50**

2.2. Characteristics **50**

2.3. Function overview **50**

2.3.1. Flash memory architecture 50

2.3.2. Error Checking and Correcting (ECC) 53

2.3.3. Read operations 54

2.3.4. Dual bank architecture with read-while-write (RWW) capability 55

2.3.5. Unlock the FMC_CTLx register 55

2.3.6. Page erase 56

2.3.7. Mass erase 57

2.3.8. Main flash programming 58
2.3.9. Main Flash Fast Programming 60
2.3.10. Check blank command 63
2.3.11. OTP programming 63
2.3.12. Shared RAM 63
2.3.13. Data Flash operation 64
2.3.14. Option bytes 0 erase 64
2.3.15. Option bytes programming 65
2.3.16. Option bytes description 66
2.3.17. Erase / program protection 68
2.3.18. Security protection 70
2.3.19. Error description 71
2.4. Register definition 73
2.4.1. Wait state register (FMC_WS) 73
2.4.2. ECC control and status register (FMC_ECCCS) 74
2.4.3. Unlock key register 0 (FMC_KEY0) 76
2.4.4. Status register 0 (FMC_STAT0) 76
2.4.5. Control register 0 (FMC_CTL0) 77
2.4.6. Address register 0 (FMC_ADDR0) 78
2.4.7. Option byte unlock key register (FMC_OBKEY) 79
2.4.8. Unlock key register 1 (FMC_KEY1) 79
2.4.9. Status register 1 (FMC_STAT1) 80
2.4.10. Control register 1 (FMC_CTL1) 81
2.4.11. Address register 1 (FMC_ADDR1) 83
2.4.12. Option byte status register (FMC_OBSTAT) 83
2.4.13. Erase / Program protection register 0 (FMC_WP0) 84
2.4.14. Erase / Program protection register 1 (FMC_WP1) 84
2.4.15. Option byte 1 control and status register (FMC_OB1CS) 85
2.4.16. Product ID register (FMC_PID) 85
3. Power management unit (PMU) 87
3.1. Overview 87
3.2. Characteristics 87
3.3. Function overview 87
3.3.1. V_{DD} / V_{DDA} power domain 88
3.3.2. 1.1V power domain 92
3.3.3. Power saving modes 92
3.4. Register definition 95
3.4.1. Control register (PMU_CTL) 95
3.4.2. Control and status register (PMU_CS) 96
4. Backup registers (BKP) 99
4.1. Overview 99

4.2. Characteristics 99
4.3. Function overview 99
4.3.1. RTC clock calibration 99
4.3.2. Tamper detection 99
4.4. Register definition 101
4.4.1. Backup data register x (BKP_DATAx) (x= 0..9) 101
4.4.2. RTC signal output control register (BKP_OCTL) 101
4.4.3. Tamper pin control register (BKP_TPCTL) 102
4.4.4. Tamper control and status register (BKP_TPCS) 103
5. Reset and clock unit (RCU) 105
5.1. Reset control unit (RCTL) 105
5.1.1. Overview 105
5.1.2. Function overview 105
5.2. Clock control unit (CCTL) 106
5.2.1. Overview 106
5.2.2. Characteristics 108
5.2.3. Function overview 108
5.3. Register definition 113
5.3.1. Control register (RCU_CTL) 113
5.3.2. Configuration register 0 (RCU_CFG0) 115
5.3.3. Interrupt register (RCU_INT) 118
5.3.4. APB2 reset register (RCU_APB2RST) 121
5.3.5. APB1 reset register (RCU_APB1RST) 123
5.3.6. AHB enable register (RCU_AHBEN) 124
5.3.7. APB2 enable register (RCU_APB2EN) 126
5.3.8. APB1 enable register (RCU_APB1EN) 128
5.3.9. Backup domain control register (RCU_BDCTL) 130
5.3.10. Reset source /clock register (RCU_RSTSCK) 131
5.3.11. AHB reset register (RCU_AHBRST) 134
5.3.12. Configuration register 1 (RCU_CFG1) 136
5.3.13. Configuration register 2 (RCU_CFG2) 137
5.3.14. Voltage key register (RCU_VKEY) 138
5.3.15. Deep-sleep mode voltage register (RCU_DSV) 138
6. Interrupt / event controller (EXTI) 140
6.1. Overview 140
6.2. Characteristics 140
6.3. Interrupts function overview 140
6.4. External interrupt and event (EXTI) block diagram 144
6.5. External Interrupt and Event function overview 144

6.6. Register definition 147
6.6.1. Interrupt enable register (EXTI_INTEN) 147
6.6.2. Event enable register (EXTI_EVEN) 147
6.6.3. Rising edge trigger enable register (EXTI_RTEN) 148
6.6.4. Falling edge trigger enable register (EXTI_FTEN) 148
6.6.5. Software interrupt event register (EXTI_SWIEV) 148
6.6.6. Pending register (EXTI_PD) 149
7. Trigger selection controller (TRIGSEL) 150
7.1. Overview 150
7.2. Characteristics 150
7.3. Function overview 150
7.4. Internal connect 151
7.5. Register definition 156
7.5.1. Trigger selection for EXTOUT0 register (TRIGSEL_EXTOUT0) 156
7.5.2. Trigger selection for EXTOUT1 register (TRIGSEL_EXTOUT1) 157
7.5.3. Trigger selection for ADC0 register (TRIGSEL_ADC0) 158
7.5.4. Trigger selection for ADC1 register (TRIGSEL_ADC1) 158
7.5.5. Trigger selection for DAC register (TRIGSEL_DAC) 159
7.5.6. Trigger selection for TIMER0_ITI register (TRIGSEL_TIMER0IN) 159
7.5.7. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN) 160
7.5.8. Trigger selection for TIMER7_ITI register (TRIGSEL_TIMER7IN) 161
7.5.9. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN) 162
7.5.10. Trigger selection for TIMER19_ITI register (TRIGSEL_TIMER19IN) 163
7.5.11. Trigger selection for TIMER19_BRKIN register (TRIGSEL_TIMER19BRKIN) 164
7.5.12. Trigger selection for TIMER20_ITI register (TRIGSEL_TIMER20IN) 165
7.5.13. Trigger selection for TIMER20_BRKIN register (TRIGSEL_TIMER20BRKIN) 166
7.5.14. Trigger selection for TIMER1_ITI register (TRIGSEL_TIMER1IN) 167
7.5.15. Trigger selection for MFCOM register (TRIGSEL_MFCOM) 168
7.5.16. Trigger selection for CAN0 register (TRIGSEL_CAN0) 169
7.5.17. Trigger selection for CAN1 register (TRIGSEL_CAN1) 170
8. General-purpose and alternate-function I/Os (GPIO and AFIO) 171
8.1. Overview 171
8.2. Characteristics 171
8.3. Function overview 171
8.3.1. GPIO pin configuration 173
8.3.2. External interrupt/event lines 173
8.3.3. Alternate functions (AF) 173
8.3.4. Additional functions 173
8.3.5. Input configuration 174
8.3.6. Output configuration 174
8.3.7. Analog configuration 175

8.3.8. Alternate function (AF) configuration 175
8.3.9. GPIO locking function 176
8.3.10. GPIO single cycle toggle function 176
8.4. Register definition 177
8.4.1. Port control register (GPIOx_CTL, x=A..F) 177
8.4.2. Port output mode register (GPIOx_OMODE, x=A..F) 178
8.4.3. Port output speed register (GPIOx_OSPD, x=A..F) 180
8.4.4. Port pull-up/down register (GPIOx_PUD, x=A..F) 182
8.4.5. Port input status register (GPIOx_ISTAT, x=A..F) 183
8.4.6. Port output control register (GPIOx_OCTL, x=A..F) 184
8.4.7. Port bit operate register (GPIOx_BOP, x=A..F) 184
8.4.8. Port configuration lock register (GPIOx_LOCK, x=A..F) 185
8.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A..F) 185
8.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A..F) 187
8.4.11. Bit clear register (GPIOx_BC, x=A..F) 188
8.4.12. Port bit toggle register (GPIOx_TG, x=A..F) 188
9. Multi-function communication Interface (MFCOM) 190
9.1. Overview 190
9.2. Characteristics 190
9.3. Block diagram 190
9.4. Function overview 191
9.4.1. Clocking and resets 191
9.4.2. Shifter 191
9.4.3. Timer 192
9.4.4. Pin 195
9.4.5. Interrupts and DMA requests 195
9.4.6. Triggers 196
9.5. Register definition 197
9.5.1. Control register (MFCOM_CTL) 197
9.5.2. Pin data register (MFCOM_PINDATA) 197
9.5.3. Shifter status register (MFCOM_SSTAT) 198
9.5.4. Shifter error register (MFCOM_SERR) 198
9.5.5. Timer status register (MFCOM_TMSTAT) 199
9.5.6. Shifter status interrupt enable register (MFCOM_SSIEN) 199
9.5.7. Shifter error interrupt enable register (MFCOM_SEIEN) 200
9.5.8. Timer status interrupt enable register (MFCOM_TMSIEN) 200
9.5.9. Shifter status DMA enable register (MFCOM_SSDMAEN) 201
9.5.10. Shifter control x register (MFCOM_SCTLx) 201
9.5.11. Shifter configuration x register (MFCOM_SCFGx) 202
9.5.12. Shifter buffer x register (MFCOM_SBUFx) 203
9.5.13. Shifter buffer x bit swapped register (MFCOM_SBUFBISx) 204
9.5.14. Shifter buffer x byte swapped register (MFCOM_SBUFBYSx) 204

9.5.15. Shifter buffer x bit byte swapped register (MFCOM_SBUFBBSx) 205
9.5.16. Timer control x register (MFCOM_TMCTLx) 205
9.5.17. Timer configuration x register (MFCOM_TMCFGx) 207
9.5.18. Timer compare x register (MFCOM_TMCMPx) 209
10. CRC calculation unit (CRC) 210
10.1. Overview 210
10.2. Characteristics 210
10.3. Function overview 211
10.4. Register definition 212
10.4.1. Data register (CRC_DATA) 212
10.4.2. Free data register (CRC_FDATA) 212
10.4.3. Control register (CRC_CTL) 213
10.4.4. Initialization data register (CRC_IDATA) 213
10.4.5. Polynomial register (CRC_POLY) 214
11. Direct memory access controller (DMA) 215
11.1. Overview 215
11.2. Characteristics 215
11.3. Block diagram 216
11.4. Function overview 216
11.4.1. DMA operation 216
11.4.2. Peripheral handshake 218
11.4.3. Arbitration 219
11.4.4. Address generation 219
11.4.5. Circular mode 219
11.4.6. Memory to memory mode 219
11.4.7. Channel configuration 219
11.4.8. Interrupt 220
11.4.9. DMA request mapping 221
11.5. Register definition 222
11.5.1. Interrupt flag register (DMA_INTF) 222
11.5.2. Interrupt flag clear register (DMA_INTC) 222
11.5.3. Channel x control register (DMA_CHxCTL) 223
11.5.4. Channel x counter register (DMA_CHxCNT) 225
11.5.5. Channel x peripheral base address register (DMA_CHxPADDR) 226
11.5.6. Channel x memory base address register (DMA_CHxMADDR) 226
12. DMA request multiplexer (DMAMUX) 228
12.1. Overview 228
12.2. Characteristics 228
12.3. Block diagram 229

12.4. Function overview 229
12.4.1. DMAMUX signals 230
12.4.2. DMAMUX request multiplexer 230
12.4.3. DMAMUX request generator 233
12.4.4. Channel configurations 234
12.4.5. Interrupt 234
12.4.6. DMAMUX mapping 234
12.5. Register definition 239
12.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG) 239
12.5.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF) 240
12.5.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC) 240
12.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG) 241
12.5.5. Request generator interrupt flag register (DMAMUX_RG_INTF) 242
12.5.6. Request generator interrupt flag clear register (DMAMUX_RG_INTC) 242
13. Debug (DBG) 244
13.1. Introduction 244
13.2. JTAG/SW function overview 244
13.2.1. Switch JTAG or SW interface 244
13.2.2. Pin assignment 244
13.2.3. JTAG daisy chained structure 245
13.2.4. Debug reset 245
13.2.5. JEDEC-106 ID code 245
13.3. Debug hold function overview 245
13.3.1. Debug support for power saving mode 245
13.3.2. Debug support for TIMER, I2C, WWDGT and FWDGT 246
13.4. Registers definition 247
13.4.1. ID code register (DBG_ID) 247
13.4.2. Control register (DBG_CTL) 247
14. Analog-to-digital converter (ADC) 250
14.1. Overview 250
14.2. Characteristics 250
14.3. Pins and internal signals 251
14.4. Function overview 252
14.4.1. Foreground calibration function 252
14.4.2. ADC clock 253
14.4.3. ADC enable 253
14.4.4. Routine sequence 253
14.4.5. Operation modes 253
14.4.6. Conversion result threshold monitor 256
14.4.7. Data storage mode 257

14.4.8. Sample time configuration 257
14.4.9. External trigger configuration 258
14.4.10. DMA request 258
14.4.11. ADC internal channels 258
14.4.12. Programmable resolution (DRES) 259
14.4.13. On-chip hardware oversampling 260
14.5. ADC sync mode 261
14.5.1. Free mode 262
14.5.2. Routine parallel mode 262
14.5.3. Routine follow-up fast mode 263
14.5.4. Routine follow-up slow mode 264
14.6. ADC interrupts 265
14.7. Register definition 266
14.7.1. Status register (ADC_STAT) 266
14.7.2. Control register 0 (ADC_CTL0) 267
14.7.3. Control register 1 (ADC_CTL1) 269
14.7.4. Sample time register 0 (ADC_SAMPT0) 270
14.7.5. Sample time register 1 (ADC_SAMPT1) 271
14.7.6. Watchdog high threshold register 0 (ADC_WDHT0) 272
14.7.7. Watchdog low threshold register 0 (ADC_WDLT0) 272
14.7.8. Routine sequence register 0 (ADC_RSQ0) 273
14.7.9. Routine sequence register 1 (ADC_RSQ1) 273
14.7.10. Routine sequence register 2 (ADC_RSQ2) 274
14.7.11. Routine data register (ADC_RDATA) 275
14.7.12. Oversample control register (ADC_OVSAMPCTL) 275
14.7.13. Watchdog 1 channel selection register (ADC_WD1SR) 277
14.7.14. Watchdog threshold register 1 (ADC_WDT1) 277
15. Digital-to-analog converter (DAC) 279
15.1. Overview 279
15.2. Characteristics 279
15.3. Function overview 280
15.3.1. DAC enable 280
15.3.2. DAC output buffer 280
15.3.3. DAC data configuration 280
15.3.4. DAC trigger 281
15.3.5. DAC conversion 281
15.3.6. DAC noise wave 281
15.3.7. DAC output voltage 282
15.3.8. DMA request 282
15.4. Register definition 284
15.4.1. DACx control register 0 (DAC_CTL0) 284

15.4.2. DACx software trigger register (DAC_SWT) 285
15.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH) 286
15.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH) 286
15.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH) 287
15.4.6. DACx_OUT0 data output register (DAC_OUT0_DO) 287
15.4.7. DACx status register 0 (DAC_STAT0) 288
16. Watchdog timer (WDGT) 289
16.1. Free watchdog timer (FWDGT) 289
16.1.1. Overview 289
16.1.2. Characteristics 289
16.1.3. Function overview 289
16.1.4. Register definition 292
16.2. Window watchdog timer (WWDGT) 296
16.2.1. Overview 296
16.2.2. Characteristics 296
16.2.3. Function overview 296
16.2.4. Register definition 299
17. Real-time clock (RTC) 301
17.1. Overview 301
17.2. Characteristics 301
17.3. Function overview 301
17.3.1. RTC reset 302
17.3.2. RTC reading 302
17.3.3. RTC configuration 302
17.3.4. RTC flag assertion 303
17.4. Register definition 305
17.4.1. RTC interrupt enable register (RTC_INTEN) 305
17.4.2. RTC control register (RTC_CTL) 305
17.4.3. RTC prescaler high register (RTC_PSCH) 306
17.4.4. RTC prescaler low register (RTC_PSCL) 307
17.4.5. RTC divider high register (RTC_DIVH) 307
17.4.6. RTC divider low register (RTC_DIVL) 307
17.4.7. RTC counter high register (RTC_CNTH) 308
17.4.8. RTC counter low register (RTC_CNTL) 308
17.4.9. RTC alarm high register (RTC_ALRMH) 309
17.4.10. RTC alarm low register (RTC_ALRML) 309
18. TIMER 310
18.1. Advanced timer (TIMERx, x=0, 7, 19, 20) 311
18.1.1. Overview 311
18.1.2. Characteristics 311
18.1.3. Block diagram 312

18.1.4. Function overview 312
18.1.5. Registers definition (TIMERx, x=0, 7, 19, 20) 347
18.2. General level0 timer (TIMERx, x=1) 404
18.2.1. Overview 404
18.2.2. Characteristics 404
18.2.3. Block diagram 404
18.2.4. Function overview 405
18.2.5. Registers definition (TIMERx, x=1) 421
18.3. Basic timer (TIMERx, x=5, 6) 444
18.3.1. Overview 444
18.3.2. Characteristics 444
18.3.3. Block diagram 444
18.3.4. Function overview 444
18.3.5. Registers definition (TIMERx, x=5, 6) 448
19. Universal synchronous/asynchronous receiver /transmitter (USART) 453
19.1. Overview 453
19.2. Characteristics 453
19.3. Function overview 454
19.3.1. USART frame format 455
19.3.2. Baud rate generation 456
19.3.3. USART transmitter 456
19.3.4. USART receiver 458
19.3.5. Use DMA for data buffer access 459
19.3.6. Hardware flow control 461
19.3.7. Multi-processor communication 462
19.3.8. LIN mode 463
19.3.9. Synchronous mode 464
19.3.10. IrDA SIR ENDEC mode 465
19.3.11. Half-duplex communication mode 467
19.3.12. Smartcard (ISO7816-3) mode 467
19.3.13. ModBus communication 469
19.3.14. Receive FIFO 469
19.3.15. Wakeup from Deep-sleep mode 470
19.3.16. USART interrupts 470
19.4. Register definition 473
19.4.1. Control register 0 (USART_CTL0) 473
19.4.2. Control register 1 (USART_CTL1) 475
19.4.3. Control register 2 (USART_CTL2) 477
19.4.4. Baud rate generator register (USART_BAUD) 480
19.4.5. Prescaler and guard time configuration register (USART_GP) 481
19.4.6. Receiver timeout register (USART_RT) 482
19.4.7. Command register (USART_CMD) 482

19.4.8. Status register (USART_STAT) 483
19.4.9. Interrupt status clear register (USART_INTC) 487
19.4.10. Receive data register (USART_RDATA) 488
19.4.11. Transmit data register (USART_TDATA) 488
19.4.12. USART coherence control register (USART_CHC) 489
19.4.13. USART receive FIFO control and status register (USART_RFCS) 490
20. Inter-integrated circuit interface (I2C) 491
20.1. Overview 491
20.2. Characteristics 491
20.3. Function overview 491
20.3.1. Clock requirements 492
20.3.2. I2C communication flow 493
20.3.3. Noise filter 495
20.3.4. I2C timings configuration 496
20.3.5. I2C reset 498
20.3.6. Data transfer 498
20.3.7. I2C slave mode 500
20.3.8. I2C master mode 505
20.3.9. SMBus support 510
20.3.10. SMBus mode 513
20.3.11. Use DMA for data transfer 514
20.3.12. I2C error and interrupts 515
20.3.13. I2C debug mode 515
20.4. Register definition 517
20.4.1. Control register 0 (I2C_CTL0) 517
20.4.2. Control register 1 (I2C_CTL1) 519
20.4.3. Slave address register 0 (I2C_SADDR0) 521
20.4.4. Slave address register 1 (I2C_SADDR1) 522
20.4.5. Timing register (I2C_TIMING) 523
20.4.6. Timeout register (I2C_TIMEOUT) 524
20.4.7. Status register (I2C_STAT) 525
20.4.8. Status clear register (I2C_STATC) 527
20.4.9. PEC register (I2C_PEC) 528
20.4.10. Receive data register (I2C_RDATA) 529
20.4.11. Transmit data register (I2C_TDATA) 529
20.4.12. Control register 2 (I2C_CTL2) 530
21. Serial peripheral interface/Inter-IC sound (SPI/I2S) 531
21.1. Overview 531
21.2. Characteristics 531
21.2.1. SPI characteristics 531
21.2.2. I2S characteristics 531

21.3. SPI function overview 532
21.3.1. SPI block diagram 532
21.3.2. SPI signal description 532
21.3.3. SPI clock timing and data format 533
21.3.4. NSS function 534
21.3.5. SPI operating modes 535
21.3.6. DMA function 544
21.3.7. CRC function 544
21.3.8. SPI interrupts 545
21.4. I2S function overview 546
21.4.1. I2S block diagram 546
21.4.2. I2S signal description 547
21.4.3. I2S audio standards 547
21.4.4. I2S clock 555
21.4.5. Operation 556
21.4.6. DMA function 560
21.4.7. I2S interrupts 560
21.5. Register definition 562
21.5.1. Control register 0 (SPI_CTL0) 562
21.5.2. Control register 1 (SPI_CTL1) 564
21.5.3. Status register (SPI_STAT) 565
21.5.4. Data register (SPI_DATA) 566
21.5.5. CRC polynomial register (SPI_CRCPOLY) 567
21.5.6. RX CRC register (SPI_RCRC) 567
21.5.7. TX CRC register (SPI_TCRC) 568
21.5.8. I2S control register (SPI_I2SCTL) 568
21.5.9. I2S clock prescaler register (SPI_I2SPSC) 570
21.5.10. Quad-SPI mode control register (SPI_QCTL) of SPI0 571
22. Comparator (CMP) 572
22.1. Overview 572
22.2. Characteristics 572
22.3. Function overview 572
22.3.1. CMP clock 573
22.3.2. CMP I / O configuration 573
22.3.3. CMP operating mode 574
22.3.4. CMP hysteresis 574
22.3.5. CMP register write protection 575
22.3.6. CMP output blanking 575
22.3.7. CMP voltage scaler function 576
22.3.8. CMP interrupt 576
22.4. Register definition 577
22.4.1. CMP Control/status register (CMPx_CS) 577

23. Controller area network (CAN) 580

23.1. Overview 580

23.2. Characteristics 580

23.3. Function overview 581

23.3.1. Mailbox descriptor 582
23.3.2. Rx FIFO descriptor 587
23.3.3. Communication modes 593
23.3.4. Power saving modes 594
23.3.5. Data transmission 595
23.3.6. Data reception 599
23.3.7. Data reception in Pretended Networking mode 606
23.3.8. CAN FD operation 609
23.3.9. Errors and states 612
23.3.10. Communication parameters 615
23.3.11. Interrupts 618

23.4. Example for a typical configuration flow of CAN 618

23.5. CAN registers 621

23.5.1. Control register 0 (CAN_CTL0) 621
23.5.2. Control register 1 (CAN_CTL1) 623
23.5.3. Timer register (CAN_TIMER) 625
23.5.4. Receive mailbox public filter register (CAN_RMPUBF) 625
23.5.5. Error register 0 (CAN_ERR0) 626
23.5.6. Error register 1 (CAN_ERR1) 626
23.5.7. Interrupt enable register (CAN_INTEN) 629
23.5.8. Status register (CAN_STAT) 630
23.5.9. Control register 2 (CAN_CTL2) 631
23.5.10. CRC for classical frame register (CAN_CRCC) 633
23.5.11. Receive FIFO public filter register (CAN_RFIFOPUBF) 634
23.5.12. Receive FIFO identifier filter matching number register (CAN_RFIFOIFMN) 634
23.5.13. Bit timing register (CAN_BT) 635
23.5.14. Receive FIFO/mailbox private filter x register (CAN_RFIFOMPFx)(x=0..31) 635
23.5.15. Pretended Networking mode control register 0 (CAN_PN_CTL0) 636
23.5.16. Pretended Networking mode timeout register (CAN_PN_TO) 637
23.5.17. Pretended Networking mode status register (CAN_PN_STAT) 638
23.5.18. Pretended Networking mode expected identifier 0 register (CAN_PN_EID0) 639
23.5.19. Pretended Networking mode expected DLC register (CAN_PN_EDLC) 639
23.5.20. Pretended Networking mode expected data low 0 register (CAN_PN_EDL0) 640
23.5.21. Pretended Networking mode expected data low 1 register (CAN_PN_EDL1) 641
23.5.22. Pretended Networking mode identifier filter / expected identifier 1 register (CAN_PN_IFEID1) 641
23.5.23. Pretended Networking mode data 0 filter / expected data high 0 register (CAN_PN_DF0EDH0) 642

23.5.24. Pretended Networking mode data 1 filter / expected data high 1 register (CAN_PN_DF1EDH1) 643
23.5.25. Pretended Networking mode received wakeup mailbox x control status information register (CAN_PN_RWMxCS)(x=0..3) 644
23.5.26. Pretended Networking mode received wakeup mailbox x identifier register (CAN_PN_RWMxI)(x=0..3) 644
23.5.27. Pretended Networking mode received wakeup mailbox x data 0 register (CAN_PN_RWMxD0)(x=0..3) 645
23.5.28. Pretended Networking mode received wakeup mailbox x data 1 register (CAN_PN_RWMxD1)(x=0..3) 645
23.5.29. FD control register (CAN_FDCTL) 646
23.5.30. FD bit timing register (CAN_FDBT) 647
23.5.31. CRC for classical and FD frame register (CAN_CRCCFD) 648
24. Appendix 649
24.1. List of abbreviations used in register 649
24.2. List of terms 649
24.3. Available peripherals 650
25. Revision history 651

List of Figures

Figure 1-1. The structure of the Cortex®-M33 processor....26
Figure 1-2. Series system architecture of GD32A50x series....28
Figure 1-3. ECC decoder....33
Figure 2-1. Process of page erase operation....57
Figure 2-2. Process of mass erase operation....58
Figure 2-3. Process of word program operation....60
Figure 2-4. Process of fast programming operation....62
Figure 3-1. Power supply overview....88
Figure 3-2. Waveform of the POR / PDR....89
Figure 3-3. Waveform of the BOR....90
Figure 3-4. Waveform of the LVD threshold....91
Figure 3-5. Waveform of the OVD threshold....91
Figure 5-1. The system reset circuit....106
Figure 5-2. Clock tree....107
Figure 5-3. HXTAL clock source....108
Figure 5-4. HXTAL clock source in bypass mode....109
Figure 6-1. Block diagram of EXTI....144
Figure 7-1. TRIGSEL main composition example....151
Figure 8-1. Basic structure of a general-pupose I/O....172
Figure 8-2. Basic structure of Input configuration....174
Figure 8-3. Basic structure of Output configuration....174
Figure 8-4. Basic structure of Analog configuration....175
Figure 8-5. Basic structure of Alternate function configuration....176
Figure 9-1. MFCOM block diagram....190
Figure 9-2. Shifter microarchitecture....192
Figure 10-1. Block diagram of CRC calculation unit....210
Figure 11-1. Block diagram of DMA....216
Figure 11-2. Handshake mechanism....218
Figure 11-3. DMA interrupt logic....220
Figure 12-1. Block diagram of DMAMUX....229
Figure 12-2. Synchronization mode....231
Figure 12-3. Event generation....232
Figure 14-1. ADC module block diagram....252
Figure 14-2. Single operation mode....253
Figure 14-3. Continuous operation mode....254
Figure 14-4. Scan operation mode, continuous disable....255
Figure 14-5. Scan operation mode, continuous enable....255
Figure 14-6. Discontinuous operation mode....256
Figure 14-7. Data storage mode of 12-bit resolution....257
Figure 14-8. Data storage mode of 10-bit resolution....257
Figure 14-9. Data storage mode of 8-bit resolution....257

Figure 14-10. Data storage mode of 6-bit resolution 257
Figure 14-11. 20-bit to 16-bit result truncation 260
Figure 14-12. Numerical example with 5-bits shift and rounding 261
Figure 14-13. ADC sync block diagram 262
Figure 14-14. Routine parallel mode on 16 channels 263
Figure 14-15. Routine follow-up fast mode on 1 channel in continuous operation mode 264
Figure 14-16. Routine follow-up slow mode on 1 channel 264
Figure 15-1. DAC block diagram 279
Figure 15-2. DAC LFSR algorithm 282
Figure 15-3. DAC triangle noise wave 282
Figure 16-1. Free watchdog block diagram 290
Figure 16-2. Window watchdog timer block diagram 296
Figure 16-3. Window watchdog timing diagram 297
Figure 17-1. Block diagram of RTC 302
Figure 17-2. RTC second and alarm waveform example (RTC_PSC = 3, RTC_ALRM = 2) 303
Figure 17-3. RTC second and overflow waveform example (RTC_PSC= 3) 304
Figure 18-1. Advanced timer block diagram 312
Figure 18-2. Timing chart of internal clock divided by 1 313
Figure 18-3. Timing chart of PSC value change from 0 to 2 314
Figure 18-4. Timing chart of up counting mode, PSC=0/2 315
Figure 18-5. Timing chart of up counting mode, change TIMERx_CAR on the go 316
Figure 18-6. Timing chart of down counting mode, PSC=0/2 317
Figure 18-7. Timing chart of down counting mode, change TIMERx_CAR on the go 317
Figure 18-8. Timing chart of center-aligned counting mode 319
Figure 18-9. Repetition counter timing chart of center-aligned counting mode 320
Figure 18-10. Repetition counter timing chart of up counting mode 320
Figure 18-11. Repetition counter timing chart of down counting mode 321
Figure 18-12. Channel 0 input capture principle 322
Figure 18-13. Multi mode channel 0 input capture principle 322
Figure 18-14. Channel output compare principle (when MCHxMSEL = 2'00, x=0, 1, 2, 3) . 324
Figure 18-15. Channel output compare principle (when MCHxMSEL = 2'01, x=0, 1, 2, 3) . 324
Figure 18-16. Channel output compare principle (with complementary output when MCHxMSEL = 2'11, x=0,1,2,3) 324
Figure 18-17. Output-compare under three modes 326
Figure 18-18. EAPWM timechart 327
Figure 18-19. CAPWM timechart 327
Figure 18-20. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 329
Figure 18-21. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 329
Figure 18-22. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 329
Figure 18-23. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 330
Figure 18-24. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 330
Figure 18-25. Four Channels outputs in Composite PWM mode 331
Figure 18-26. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00) 332

Figure 18-27. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00) .332
Figure 18-28. Channel output complementary PWM with dead-time insertion.....335
Figure 18-29. Break function diagram 336
Figure 18-30. Output behavior of the channel in response to a break (the break high active) 337
Figure 18-31. Counter behavior with CI0FE0 polarity non-inverted in mode 2.....338
Figure 18-32. Counter behavior with CI0FE0 polarity inverted in mode 2.....339
Figure 18-33. Hall sensor is used for BLDC motor 340
Figure 18-34. Hall sensor timing between two timers.....340
Figure 18-35. Restart mode.....341
Figure 18-36. Pause mode.....342
Figure 18-37. Event mode 342
Figure 18-38. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x99 343
Figure 18-39. TIMER0 master/slave mode example 344
Figure 18-40. Triggering TIMER0 with enable signal of TIMER1.....345
Figure 18-41. Triggering TIMER0 and TIMER1 with TIMER1's CI0 input.....346
Figure 18-42. General Level 0 timer block diagram 405
Figure 18-43. Timing chart of internal clock divided by 1 406
Figure 18-44. Timing chart of PSC value change from 0 to 2 407
Figure 18-45. Timing chart of up counting mode, PSC=0/2 408
Figure 18-46. Timing chart of up counting, change TIMERx_CAR on the go 408
Figure 18-47. Timing chart of down counting mode, PSC=0/2 409
Figure 18-48. Timing chart of down counting mode, change TIMERx_CAR on the go.....410
Figure 18-49. Timing chart of center-aligned counting mode.....411
Figure 18-50. Channels input capture principle 412
Figure 18-51. Channel output compare principle (x=0,1,2,3) 413
Figure 18-52. Output-compare under three modes.....414
Figure 18-53. EAPWM timechart.....415
Figure 18-54. CAPWM timechart 415
Figure 18-55. Counter behavior with CI0FE0 polarity non-inverted in mode 2.....417
Figure 18-56. Counter behavior with CI0FE0 polarity inverted in mode 2.....417
Figure 18-57. Restart mode.....418
Figure 18-58. Pause mode.....418
Figure 18-59. Event mode 419
Figure 18-60. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x99 420
Figure 18-61. Basic timer block diagram.....444
Figure 18-62. Timing chart of internal clock divided by 1 445
Figure 18-63. Timing chart of PSC value change from 0 to 2 445
Figure 18-64. Timing chart of up counting mode, PSC=0/2 446
Figure 18-65. Timing chart of up counting mode, change TIMERx_CAR on the go.....447
Figure 19-1. USART module block diagram.....455
Figure 19-2. USART character frame (8 bits data and 1 stop bit).....455
Figure 19-3. USART transmit procedure 457
Figure 19-4. Oversampling method of a receive frame bit (OSB=0).....458

Figure 19-5. Configuration step when using DMA for USART transmission 460
Figure 19-6. Configuration step when using DMA for USART reception 461
Figure 19-7. Hardware flow control between two USARTs 461
Figure 19-8. Hardware flow control 462
Figure 19-9. Break frame occurs during idle state 464
Figure 19-10. Break frame occurs during a frame 464
Figure 19-11. Example of USART in synchronous mode 465
Figure 19-12. 8-bit format USART synchronous waveform (CLEN=1) 465
Figure 19-13. IrDA SIR ENDEC module 466
Figure 19-14. IrDA data modulation 466
Figure 19-15. ISO7816-3 frame format 467
Figure 19-16. USART Receive FIFO structure 470
Figure 19-17. USART interrupt mapping diagram 472
Figure 20-1. I2C module block diagram 492
Figure 20-2. Data validation 493
Figure 20-3. START and STOP signal 494
Figure 20-4. I2C communication flow with 10-bit address (Master Transmit) 494
Figure 20-5. I2C communication flow with 7-bit address (Master Transmit) 495
Figure 20-6. I2C communication flow with 7-bit address (Master Receive) 495
Figure 20-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0) 495
Figure 20-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1) 495
Figure 20-9. Data hold time 496
Figure 20-10. Data setup time 496
Figure 20-11. Data transmission 498
Figure 20-12. Data reception 499
Figure 20-13. I2C initialization in slave mode 502
Figure 20-14. Programming model for slave transmitting when SS=0 503
Figure 20-15. Programming model for slave transmitting when SS=1 504
Figure 20-16. Programming model for slave receiving 505
Figure 20-17. I2C initialization in master mode 506
Figure 20-18. Programming model for master transmitting (N<=255) 507
Figure 20-19. Programming model for master transmitting (N>255) 508
Figure 20-20. Programming model for master receiving (N<=255) 509
Figure 20-21. Programming model for master receiving (N>255) 510
Figure 20-22. SMBus Master Transmitter and Slave Receiver communication flow 514
Figure 20-23. SMBus Master Receiver and Slave Transmitter communication flow 514
Figure 21-1. Block diagram of SPI 532
Figure 21-2. SPI timing diagram in normal mode 533
Figure 21-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0) 534
Figure 21-4. A typical full-duplex connection 537
Figure 21-5. A typical simplex connection (Master: Receive, Slave: Transmit) 537
Figure 21-6. A typical simplex connection (Master: Transmit only, Slave: Receive) 537

Figure 21-7. A typical bidirectional connection 537
Figure 21-8. Timing diagram of TI master mode with discontinuous transfer 539
Figure 21-9. Timing diagram of TI master mode with continuous transfer 540
Figure 21-10. Timing diagram of TI slave mode 540
Figure 21-11. Timing diagram of NSS pulse with continuous transmit 541
Figure 21-12. Timing diagram of quad write operation in Quad-SPI mode 542
Figure 21-13. Timing diagram of quad read operation in Quad-SPI mode 543
Figure 21-14. Block diagram of I2S 546
Figure 21-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 547
Figure 21-16. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 548
Figure 21-17. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 548
Figure 21-18. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 548
Figure 21-19. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 548
Figure 21-20. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 548
Figure 21-21. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 549
Figure 21-22. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 549
Figure 21-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) ... 549
Figure 21-24. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) ... 549
Figure 21-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) ... 550
Figure 21-26. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) ... 550
Figure 21-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) ... 550
Figure 21-28. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) ... 550
Figure 21-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) ... 550
Figure 21-30. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) ... 550
Figure 21-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 551
Figure 21-32. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 551
Figure 21-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 551
Figure 21-34. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 551
Figure 21-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 552
Figure 21-36. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 552
Figure 21-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 552
Figure 21-38. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 552
Figure 21-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 552
Figure 21-40. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 553
Figure 21-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 553
Figure 21-42. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 553

Figure 21-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 553
Figure 21-44. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 553
Figure 21-45. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 554
Figure 21-46. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 554
Figure 21-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 554
Figure 21-48. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 554
Figure 21-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 554
Figure 21-50. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 555
Figure 21-51. Block diagram of I2S clock generator 555
Figure 21-52. I2S initialization sequence 557
Figure 21-53. I2S master reception disabling sequence 559
Figure 22-1. CMP block diagram 573
Figure 22-2. CMP hysteresis 575
Figure 22-3. The CMP outputs signal blanking 576
Figure 23-1. CAN module block diagram 581
Figure 23-2. Transmitter delay 612
Figure 23-3. CAN bit time 615

List of Tables

Table 1-1. Bus Interconnection Matrix 26
Table 1-2. Memory map of GD32A50x devices 29
Table 2-1. Base address and size for 384 KB flash memory 51
Table 2-2. Base address and size for 256 KB flash memory 51
Table 2-3. Base address and size for 128KB flash memory 52
Table 2-4. The relation between WSCNT and AHB clock frequency when LDO is 1.1V 54
Table 2-5. Option bytes 0 66
Table 2-6. Option bytes 1 68
Table 2-7. OB_BK0WP bit for pages protected 69
Table 2-8. OB_BK1WP bit for pages protected 69
Table 2-9. OB_DFWP bit for pages protected 70
Table 2-10. PGSERR conditions 71
Table 2-11. PGAERR conditions 72
Table 2-12. PGERR conditions 72
Table 2-13. WPERR conditions 72
Table 3-1. Power saving mode summary 94
Table 5-1. Clock source select 111
Table 5-2. Core domain voltage selected in Deep-sleep mode 112
Table 6-1. NVIC exception types in Cortex®-M33 141
Table 6-2. Interrupt vector table 141
Table 6-3. EXTI source 145
Table 7-1. Trigger input bit fields selection 151
Table 7-2. TRIGSEL input and output mapping 153
Table 8-1. GPIO configuration table 172
Table 9-1. Mode of shifter 192
Table 9-2. MFCOM interrupts and DMA requests 196
Table 11-1. DMA transfer operation 217
Table 11-2. interrupt events 220
Table 12-1. DMAMUX signals 230
Table 12-2. Interrupt events 234
Table 12-3. Request multiplexer input mapping 235
Table 12-4. Trigger input mapping 237
Table 12-5. Synchronization input mapping 238
Table 13-1. Pin assignment 245
Table 14-1. ADC internal input signals 251
Table 14-2. ADC input pins definition 251
Table 14-3. External trigger source for routine sequence 258
Table 14-4. t_{CONV} timings depending on resolution 259
Table 14-5. Maximum output results for N and M (Grayed values indicates truncation) 261
Table 14-6. ADC sync mode table 262
Table 15-1. DAC I/O description 280

Table 15-2. DAC triggers and outputs summary 280
Table 15-3. Triggers of DAC 281
Table 16-1. Min/max FWDGT timeout period at 40KHz (IRC40K) 291
Table 16-2. Min-max timeout value at 50 MHz (f_{PCLK1}) 297
Table 18-1. Timers (TIMERx) are divided into three sorts 310
Table 18-2. Advanced timer channel description 312
Table 18-3.The Composite PWM pulse width 328
Table 18-4. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 334
Table 18-5. Counting direction in different quadrature decoder mode 338
Table 18-6. Examples of slave mode 341
Table 18-7. Counting direction in different quadrature decoder mode 416
Table 18-8. Examples of slave mode 417
Table 19-1. Description of USART important pins 454
Table 19-2. Configuration of stop bits 456
Table 19-3. USART interrupt requests 470
Table 20-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors) 492
Table 20-2. Data setup time and data hold time 497
Table 20-3. Communication modes to be shut down 499
Table 20-4. SMBus with PEC configuration 512
Table 20-5. I2C error flags 515
Table 20-6. I2C interrupt events 515
Table 21-1. SPI signal description 532
Table 21-2. Quad-SPI signal description 533
Table 21-3. NSS function in slave mode 534
Table 21-4. NSS function in master mode 535
Table 21-5. SPI operating modes 535
Table 21-6. SPI interrupt requests 546
Table 21-7. I2S bitrate calculation formulas 555
Table 21-8. Audio sampling frequency calculation formulas 556
Table 21-9. Direction of I2S interface signals for each operation mode 556
Table 21-10. I2S interrupt 561
Table 22-1. CMP inputs and outputs summary 573
Table 23-1. Mailbox descriptor with 64 byte payload 582
Table 23-2. Data bytes for DLC 584
Table 23-3. Mailbox Rx CODE 584
Table 23-4. Mailbox Tx CODE 585
Table 23-5. Mailbox size 587
Table 23-6. Rx FIFO descriptor 588
Table 23-7. Mailbox arbitration value(32 bit) when local priority disabled 597
Table 23-8. Mailbox arbitration value(35 bit) when local priority enabled 597
Table 23-9. Rx mailbox matching 604
Table 23-10. Rx FIFO matching 605
Table 23-11. Interrupt events 618

Table 23-12. Rx FIFO filter element number 632
Table 24-1. List of abbreviations used in register 649
Table 24-2. List of terms 649
Table 25-1. Revision history 651

1. System and memory architecture

The GD32A50x series are 32-bit general-purpose microcontrollers based on the Arm® Cortex®-M33 processor. The Arm® Cortex®-M33 processor includes two AHB buses known as Code and System buses. All memory accesses of the Arm® Cortex®-M33 processor are executed on these two buses according to the different purposes and the target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

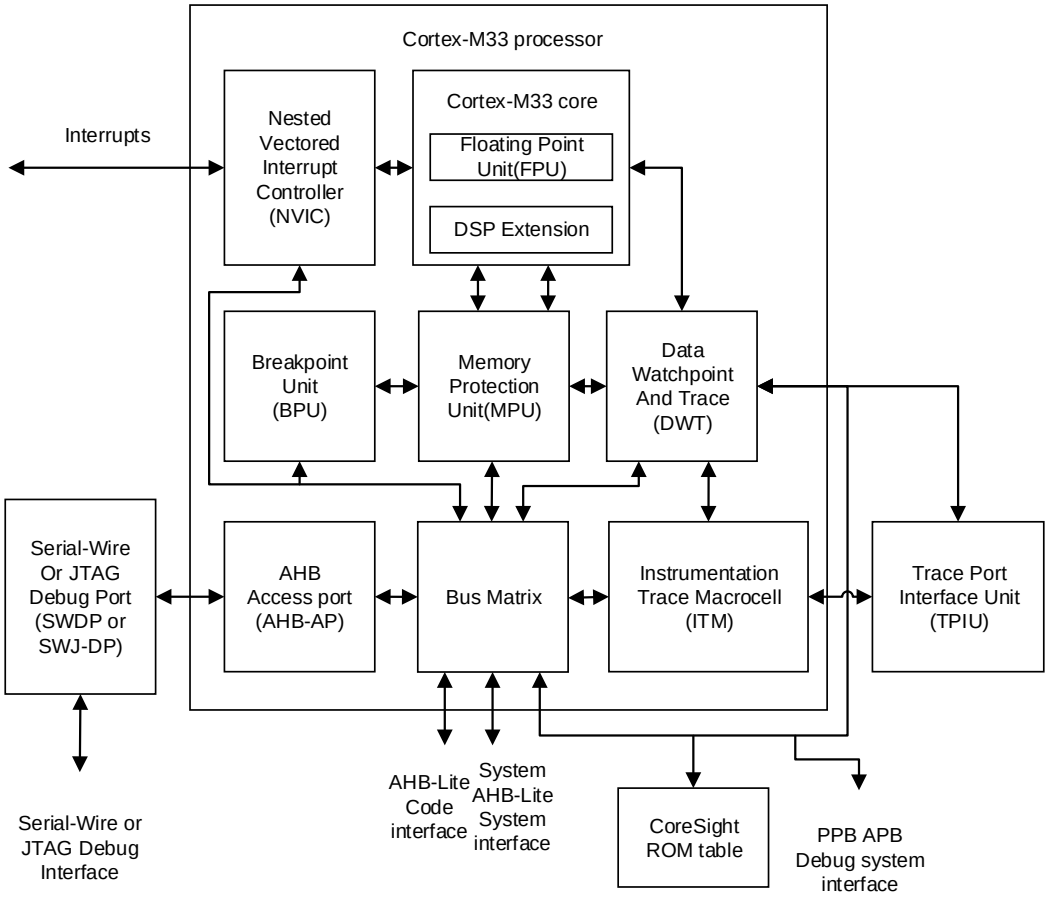
1.1. Arm® Cortex®-M33 processor

The Cortex®-M33 processor is a 32-bit processor that possesses low interrupt latency and low-cost debug. The characteristics of integrated and advanced make the Cortex®-M33 processor suitable for market products that require microcontrollers with high performance and low power consumption. The Cortex®-M33 processor is based on the Armv8 architecture and supports a powerful and scalable instruction set including general data processing I/O control tasks, advanced data processing bit field manipulations and DSP. Some system peripherals listed below are also provided by Cortex®-M33:

- Internal Bus Matrix connected with Code bus, System bus, and Private Peripheral Bus (PPB) and debug accesses
- Nested Vectored Interrupt Controller (NVIC)
- Breakpoint Unit (BPU)
- Data Watchpoint and Trace (DWT)
- Instrumentation Trace Macrocell (ITM)
- Serial Wire JTAG Debug Port (SWJ-DP)
- Trace Port Interface Unit (TPIU)
- Memory Protection Unit (MPU)
- Floating Point Unit (FPU)
- DSP Extension (DSP)

The following figure ***Figure 1-1. The structure of the Cortex®-M33 processor*** shows the Arm® Cortex®-M33 processor block diagram. For more information, refer to the Arm® Cortex®-M33 Technical Reference Manual.

Figure 1-1. The structure of the Cortex®-M33 processor



Note: Although the GD32A50x series has an ITM module, it does not support the output of trace data.

1.2. System architecture

A 32-bit multilayer bus is implemented in the GD32A50x devices, which enables parallel access paths between multiple masters and slaves in the system. The multilayer bus consists of an AHB interconnect matrix, one AHB bus and two APB buses. The interconnection relationship of the AHB interconnect matrix is shown below. In the following table, “1” indicates the corresponding master is able to access the corresponding slave through the AHB interconnect matrix, while the blank means the corresponding master cannot access the corresponding slave through the AHB interconnect matrix.

This architecture is shown in ***Table 1-1. Bus Interconnection Matrix***.

Table 1-1. Bus Interconnection Matrix

	CBUS	SBUS	DMA0	DMA1
FMC	1	0	1	1
SRAM	1	1	1	1

	CBUS	SBUS	DMA0	DMA1
AHB1	0	1	1	1
AHB2	0	1	1	1

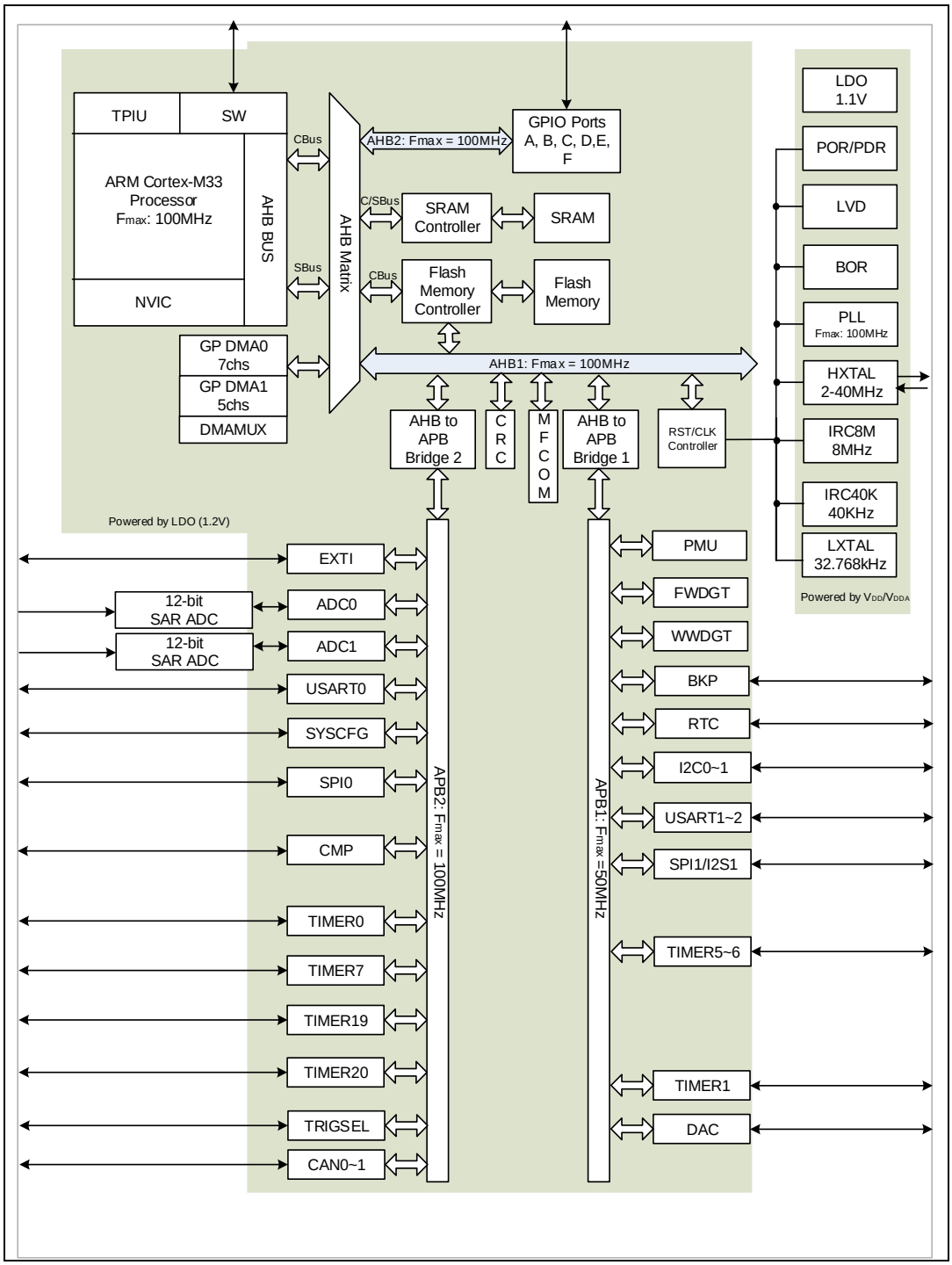
As is shown above, there are several masters connected with the AHB interconnect matrix, including CBUS, SBUS, DMA0 and DMA1. CBUS is the code bus of the Cortex®-M33 core, which is used for any instruction fetch and data access to the Code region. Similarly, SBUS is the system bus of the Cortex®-M33 core, which is used for instruction/vector fetches, data loading/storing and debugging access of the system regions. The system regions include the internal SRAM region and the Peripheral region. DMA0 and DMA1 are the buses of DMA0 and DMA1 respectively.

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM, AHB1, AHB2. FMC is the bus interface of the flash memory controller. SRAM is on-chip static random access memories. AHB1 is the AHB bus connected with all of the AHB slaves except GPIO. AHB2 is the AHB bus connected with GPIO. While APB1 and APB2 are the two APB buses connected with all of the APB slaves.

The two APB buses connect with all the APB peripherals. APB1 is up to 50MHz, APB2 operates at full speed (up to 100MHz depending on the device).

These are interconnected using a multilayer AHB bus architecture as shown in ***Figure 1-2. Series system architecture of GD32A50x series*** below:

Figure 1-2. Series system architecture of GD32A50x series



1.3. Memory map

The Arm® Cortex®-M33 processor is structured in Harvard architecture which can use separate buses to fetch instructions and load/store data. Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space which is the maximum address range of the Cortex®-M33 since the bus address width is 32-bit. Additionally, a pre-defined memory map is provided by the Cortex®-M33 processor to reduce

the software complexity of repeated implementation of different device vendors. In the map, some regions are used by the Arm® Cortex®-M33 system peripherals which can not be modified. However, the other regions are available to the vendors. ***Table 1-2. Memory map of GD32A50x devices*** shows the memory map of the GD32A50x devices, including Code, SRAM, peripheral, and other pre-defined regions. Almost each peripheral is allocated 1KB of space. This allows simplifying the address decoding for each peripheral.

Table 1-2. Memory map of GD32A50x devices

Pre-defined Regions	Bus	Address	Peripherals
		0xE004 4400 - 0xE00F FFFF	Cortex M33 internal peripherals
		0xE004 4000 – 0xE004 43FF	DBG
		0xE000 0000 – 0xE004 3FFF	Cortex M33 internal peripherals
External RAM		0x6000 0000 - 0x9FFF FFFF	Reserved
Peripheral	AHB1	0x5000 0000 - 0x5FFF FFFF	Reserved
	AHB2	0x4800 1800 - 0x4FFF FFFF	Reserved
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	GPIOE
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4003 8C00 - 0x47FF FFFF	Reserved
		0x4003 8400 - 0x4003 8BFF	MFCOM
		0x4002 3400 - 0x4003 83FF	Reserved
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	Reserved
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	Reserved
		0x4002 1800 - 0x4002 1BFF	Reserved
		0x4002 1400 - 0x4002 17FF	Reserved
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	Reserved
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 C000 - 0x4001 FFFF	Reserved
		0x4001 B000 - 0x4001 BFFF	CAN1
		0x4001 A000 - 0x4001 AFFF	CAN0
		0x4001 8800 - 0x4001 9FFF	Reserved
		0x4001 8400 - 0x4001 87FF	TRIGSEL
		0x4001 8000 - 0x4001 83FF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 5800 - 0x4001 7BFF	Reserved
		0x4001 5400 - 0x4001 57FF	TIMER20
		0x4001 5000 - 0x4001 53FF	TIMER19
		0x4001 4C00 - 0x4001 4FFF	Reserved
		0x4001 4800 - 0x4001 4BFF	Reserved
		0x4001 4400 - 0x4001 47FF	Reserved
		0x4001 4000 - 0x4001 43FF	Reserved
		0x4001 3C00 - 0x4001 3FFF	Reserved
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	ADC1
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	Reserved
		0x4001 1C00 - 0x4001 1FFF	Reserved
		0x4001 1800 - 0x4001 1BFF	Reserved
		0x4001 1400 - 0x4001 17FF	Reserved
		0x4001 1000 - 0x4001 13FF	Reserved
		0x4001 0C00 - 0x4001 0FFF	Reserved
		0x4001 0800 - 0x4001 0BFF	Reserved
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
	APB1	0x4000 DC00 - 0x4000 FFFF	Reserved
		0x4000 D800 - 0x4000 DBFF	Reserved
		0x4000 D400 - 0x4000 D7FF	Reserved
		0x4000 D000 - 0x4000 D3FF	Reserved
		0x4000 CC00 - 0x4000 CFFF	Reserved
		0x4000 C800 - 0x4000 CBFF	Reserved
		0x4000 C400 - 0x4000 C7FF	Reserved
		0x4000 C000 - 0x4000 C3FF	Reserved
		0x4000 8800 - 0x4000 BFFF	Reserved
		0x4000 8400 - 0x4000 87FF	Reserved
		0x4000 8000 - 0x4000 83FF	Reserved
		0x4000 7C00 - 0x4000 7FFF	Reserved
		0x4000 7800 - 0x4000 7BFF	Reserved
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	BKP
		0x4000 6800 - 0x4000 6BFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x4000 6400 - 0x4000 67FF	Reserved
		0x4000 6000 - 0x4000 63FF	Reserved
		0x4000 5C00 - 0x4000 5FFF	Reserved
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	Reserved
		0x4000 4C00 - 0x4000 4FFF	Reserved
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	Reserved
		0x4000 3C00 - 0x4000 3FFF	Reserved
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	Reserved
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	Reserved
		0x4000 2000 - 0x4000 23FF	Reserved
		0x4000 1C00 - 0x4000 1FFF	Reserved
		0x4000 1800 - 0x4000 1BFF	Reserved
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	Reserved
		0x4000 0800 - 0x4000 0BFF	Reserved
		0x4000 0400 - 0x4000 07FF	Reserved
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2000 D000 - 0x3FFF FFFF	Reserved
		0x2000 C000 - 0x2000 CFFF	Shared SRAM(4KB)
		0x2000 8000 - 0x2000 BFFF	SRAM(48KB)
		0x2000 6000 - 0x2000 7FFF	
		0x2000 4000 - 0x2000 5FFF	
		0x2000 0000 - 0x2000 3FFF	
Code		0x1FFF FC10 - 0x1FFF FFFF	Reserved
		0x1FFF FC00 - 0x1FFF FC0F	Reserved
		0x1FFF F818 - 0x1FFF BFFF	Reserved
		0x1FFF F800 - 0x1FFF F817	Option Bytes (24B)
		0x1FFF B000 - 0x1FFF F7FF	System memory(18KB)
		0x1FFF 7400 - 0x1FFF AFFF	Reserved
		0x1FFF 7000 - 0x1FFF 73FF	OTP(1KB)
		0x0A00 D000 - 0x1FFF 6FFF	Reserved
		0x0A00 C000 - 0x0A00 CFFF	Shared SRAM(4KB)

Pre-defined Regions	Bus	Address	Peripherals
		0x0A00 8000 - 0x0A00 BFFF	SRAM(48KB)
		0x0A00 6000 - 0x0A00 7FFF	
		0x0A00 4000 - 0x0A00 5FFF	
		0x0A00 0000 - 0x0A00 3FFF	
		0x08C0 1000 - 0x09FF FFFF	Reserved
		0x08C0 0000 - 0x08C0 0FFF	Reserved
		0x0881 0000 - 0x08BF FFFF	Reserved
		0x0880 0000 - 0x0880 FFFF	DFlash(64KB)
		0x0808 0000 - 0x0871 FFFF	Reserved
		0x0806 0000 - 0x0807 FFFF	Reserved
		0x0802 0000 - 0x0805 FFFF	Main Flash memory
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0006 0000 - 0x07FF FFFF	Reserved
		0x0002 0000 - 0x0005 FFFF	Aliased to Flash or system memory
		0x0001 0000 - 0x0001 FFFF	
		0x0000 0000 - 0x0000 FFFF	

Note: 0x20000000-0x2000BFFF and 0x0A000000-0x0A00BFFF are two different logical addresses mapped to the same SRAM. 0x2000C000-0x2000CFFF and 0x0A00C000-0x0A00CFFF are two different logical addresses mapped to the same shared SRAM.

1.3.1. On-chip SRAM memory

The GD32A50x series contain up to 48KB of on-chip SRAM which starts at the address 0x2000 0000. It supports byte, half-word (16 bits), and word (32 bits) accesses.

ECC

When reading and writing SRAM, it supports 7-bit ECC function. It can correct 1 bit error and detect multiple bits (two bits) error.

It must be written before reading SRAM, otherwise it may cause ECC error. Unaligned read operations will be performed in accordance with 32-bit read operations. Non-aligned write operations will produce a read-modify-write process. For example, when 16-bit data is written into SRAM, firstly the another 16-bit data is read out from the SRAM, and the 16-bit that need to be written are combined to a 32-bit data, and finally the 32-bit data is written into the SRAM together. Therefore, when initializing SRAM, it can only be written in a 32-bit width.

The ECC module is composed of an encoder and a decoder.

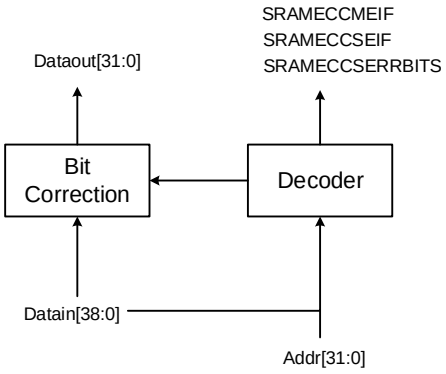
Encoder: When performing a SRAM write operation, a 7-bit ECC code will be generated and written into the SRAM together with the data.

Decoder: When performing a SRAM read operation, it uses the same algorithm as the

encoder to decode and generate a 7-bit ECC code. The ECC code includes ECC error status and information which specific bit of the 32-bit data has single bit error.

The decoder is shown in the figure *Figure 1-3. ECC decoder* below:

Figure 1-3. ECC decoder



EEIC

The EEIC(ECC Error Interrupt Control) module provides the function of ECC error status management and ECC interrupt configuration.

Single bit correction error event

When a single-bit correction error event is detected in SRAM, EEIC:

(1) The SRAMECCSEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG3 records the address where the single-bit error correction event occurred.

Multi-bits (two bits) non-correction error event

When a multi-bits (two bits) non-correction error event is detected in SRAM, EEIC:

(1) The SRAMECCMEIF bit int SYSCFG_STAT register will be set. Software can clear it by writing 1.
(2) The SYSCFG_CFG3 records the address where the two bits non-correction error event occurred.

Single bit error correction interrupt

Set the SRAMECCSEIE bit in SYSCFG_CFG3 register. When a single-bit error correctable event is detected, a corresponding interrupt will be generated.

Multi-bits (two bits) non-correction error interrupt

Set the SRAMECCMEIE bit in SYSCFG_CFG3 register. When a multi-bits (two bits) error non-correction event is detected, a NMI interrupt will be generated.

1.3.2. On-chip Flash memory

The devices provide high-density on-chip flash memory, which is structured as follows:

- Up to 384KB of main Flash memory.
- Up to 18KB of information blocks for the boot loader.
- Option bytes to configure the device

Refer to ***Flash memory controller (FMC)*** for more details.

1.4. Boot configuration

The GD32A50x series provide up to two kinds of boot sources which can be selected by the BOOT0 pin. The value on the pin is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT0 pin after a power-on reset or a system reset to select the required boot source. Once the pin has been sampled, it is free and can be used for other purposes.

After power-on sequence or a system reset, the Arm® Cortex®-M33 processor fetches the top-of-stack value from address 0x0000 0000 and the base address of boot code from 0x0000 0004 in sequence. Then, it starts executing code from the base address of boot code.

According to the selected boot source, either the main flash memory (original memory space beginning at 0x0800 0000) or the system memory (original memory space beginning at 0x1FFF B000) is aliased in the boot memory space which begins at the address 0x0000 0000.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory. For GD32A503xxxx and GD32A503xxxx0E, the boot loader can be activated through one of the following interfaces: USART0 (PA10 and PA11), LIN (PA3 and PA4), and CAN0 (PB13 and PB14).

Note:

（1） GD32A503xxxx supports booting from the main FLASH or system memory. (Determined by BOOT0, when BOOT0 = 0, it boots from the main FLASH; when BOOT0 = 1, it boots from system memory.)

（2） GD32A503xxxx0E supports booting from the main FLASH or system memory. (Determined by BOOT0, when BOOT0 = 0, it boots from the main FLASH; when BOOT0 = 1, it boots from system memory.)

（3） GD32A503xxxx1E only supports booting from the main FLASH. (It always boots from the main FLASH, and BOOT0 can be any value.)

1.5. System configuration controller

The main purposes of the system configuration controller (SYSCFG) are the following:

- Remapping of some I/O ports
- Managing the external interrupt line connection to the GPIOs

1.6. System configuration registers

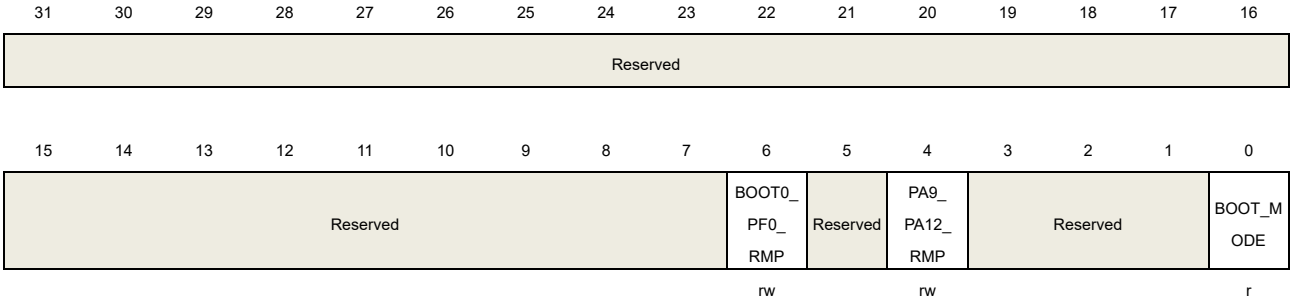
SYSCFG base address: 0x4001 0000

1.6.1. System configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00

Reset value: 0x0000 000X (X indicates BOOT_MODE may be any value according to the BOOT0 pin after reset.)

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	BOOT0_PF0_RMP	BOOT0 and PF0 remapping bit This bit is set and cleared by software. It controls the mapping of either BOOT0 or PF0 function on the BOOT0 pin. 0: No remap (BOOT0 function is mapping on the BOOT0 pin) 1: Remap (PF0 function is mapping on the BOOT0 pin)
5	Reserved	Must be kept at reset value.
4	PA9_PA12_RMP	PA9 and PA12 remapping bit for small packages (32 pins). This bit is set and cleared by software. It controls the mapping of either PA9/12 or PA10/11 pin pair on small pin-count packages. 0: No remap (pin pair PA9/12 mapped on the pins) 1: Remap (pin pair PA9/12 mapped instead of PA10/11)
3:1	Reserved	Must be kept at reset value.
0	BOOT_MODE	Boot mode Bit0 is mapping to the BOOT0 pin. 0: Boot from the main flash 1: Boot from the system flash memory

1.6.2. System configuration register 1 (SYSCFG_CFG1)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC0CH 9RMP	ADC0CH 8RMP	ADC1CH 15RMP	ADC1CH 14RMP	Reserved											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31	ADC0CH9RMP	ADC0 channel 9 remapping bit 0: ADC0_IN9 is mapping on PB1. 1: ADC0_IN9 is mapping on PC6.
30	ADC0CH8RMP	ADC0 channel 8 remapping bit 0: ADC0_IN8 is mapping on PB2. 1: ADC0_IN8 is mapping on PC7.
29	ADC1CH15RMP	ADC1 channel 15 remapping bit 0: ADC1_IN15 is mapping on PB13. 1: ADC1_IN15 is mapping on PD14.
28	ADC1CH14RMP	ADC1 channel 14 remapping bit 0: ADC1_IN14 is mapping on PB14. 1: ADC1_IN14 is mapping on PD15.
27:0	Reserved	Must be kept at reset value.

1.6.3. EXTI sources selection register 0 (SYSCFG_EXTISS0)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI3_SS[3:0]	EXTI 3 sources selection

		X000: PA3 pin X001: PB3 pin X010: PC3 pin X011: PD3 pin X100: PE3 pin X101: PF3 pin X110: reserved X111: reserved
11:8	EXTI2_SS[3:0]	EXTI 2 sources selection X000: PA2 pin X001: PB2 pin X010: PC2 pin X011: PD2 pin X100: PE2 pin X101: PF2 pin X110: reserved X111: reserved
7:4	EXTI1_SS[3:0]	EXTI 1 sources selection X000: PA1 pin X001: PB1 pin X010: PC1 pin X011: PD1 pin X100: PE1 pin X101: PF1 pin X110: reserved X111: reserved
3:0	EXTI0_SS[3:0]	EXTI 0 sources selection X000: PA0 pin X001: PB0 pin X010: PC0 pin X011: PD0 pin X100: PE0 pin X101: PF0 pin X110: reserved X111: reserved

1.6.4. EXTI sources selection register 1 (SYSCFG_EXTISS1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI7_SS[3:0]	EXTI 7 sources selection X000: PA7 pin X001: PB7 pin X010: PC7 pin X011: PD7 pin X100: PE7 pin X101: PF7 pin X110: reserved X111: reserved
11:8	EXTI6_SS[3:0]	EXTI 6 sources selection X000: PA6 pin X001: PB6 pin X010: PC6 pin X011: PD6 pin X100: PE6 pin X101: PF6 pin X110: reserved X111: reserved
7:4	EXTI5_SS[3:0]	EXTI 5 sources selection X000: PA5 pin X001: PB5 pin X010: PC5 pin X011: PD5 pin X100: PE5 pin X101: PF5 pin X110: reserved X111: reserved
3:0	EXTI4_SS[3:0]	EXTI 4 sources selection X000: PA4 pin X001: PB4 pin X010: PC4 pin X011: PD4 pin X100: PE4 pin X101: PF4 pin

X110: reserved

X111: reserved

1.6.5. EXTI sources selection register 2 (SYSCFG_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI11_SS[3:0]	EXTI 11 sources selection X000: PA11 pin X001: PB11 pin X010: PC11 pin X011: PD11 pin X100: PE11 pin X101: reserved X110: reserved X111: reserved
11:8	EXTI10_SS[3:0]	EXTI 10 sources selection X000: PA10 pin X001: PB10 pin X010: PC10 pin X011: PD10 pin X100: PE10 pin X101: reserved X110: reserved X111: reserved
7:4	EXTI9_SS[3:0]	EXTI 9 sources selection X000: PA9 pin X001: PB9 pin X010: PC9 pin X011: PD9 pin X100: PE9 pin X101: reserved

		X110: reserved X111: reserved
3:0	EXTI8_SS[3:0]	EXTI 8 sources selection X000: PA8 pin X001: PB8 pin X010: PC8 pin X011: PD8 pin X100: PE8 pin X101: reserved X110: reserved X111: reserved

1.6.6. EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI15_SS[3:0]	EXTI 15 sources selection X000: PA15 pin X001: PB15 pin X010: PC15 pin X011: PD15 pin X100: PE15 pin X101: reserved X110: reserved X111: reserved
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection X000: PA14 pin X001: PB14 pin X010: PC14 pin X011: PD14 pin X100: PE14 pin X101: reserved

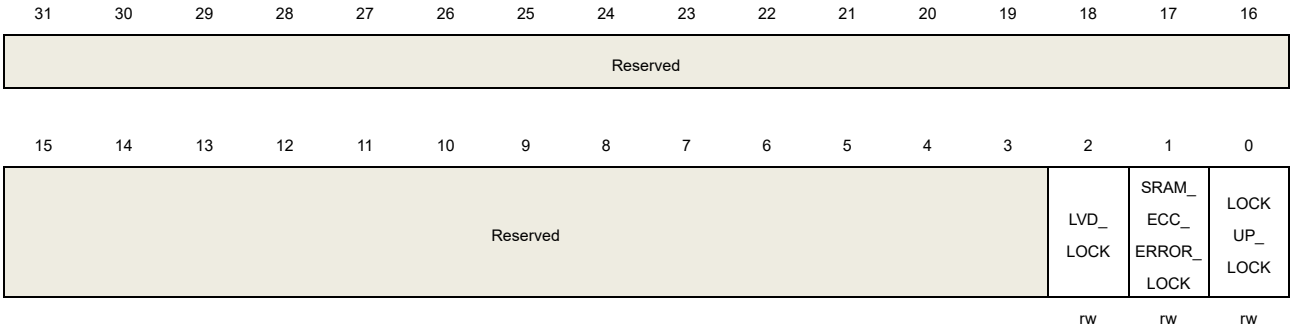
		X110: reserved X111: reserved
7:4	EXTI13_SS[3:0]	EXTI 13 sources selection X000: PA13 pin X001: PB13 pin X010: PC13 pin X011: PD13 pin X100: PE13 pin X101: reserved X110: reserved X111: reserved
3:0	EXTI12_SS[3:0]	EXTI 12 sources selection X000: PA12 pin X001: PB12 pin X010: PC12 pin X011: PD12 pin X100: PE12 pin X101: reserved X110: reserved X111: reserved

1.6.7. System configuration register 2 (SYSCFG_CFG2)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	LVD_LOCK	LVD lock This bit is set by software and cleared by a system reset. 0: The LVD interrupt is disconnected from the break input of TIMER0/7/19/20. LVDE and LVDT[2:0] in the PMU_CTL register can be programmed. 1: The LVD interrupt is connected from the break input of TIMER0/7/19/20. LVDE

		and LVDT[2:0] in the PMU_CTL register are read only.
1	SRAM_ECC ERROR_LOCK	SRAM ECC check error lock This bit is set by software and cleared by a system reset. 0: The SRAM ECC check error is disconnected from the break input of TIMER0/7/19/20. 1: The SRAM ECC check error is connected from the break input of TIMER0/7/19/20.
0	LOCKUP_LOCK	Cortex-M33 LOCKUP output lock This bit is set by software and cleared by a system reset. 0: The Cortex-M33 LOCKUP output is disconnected from the break input of TIMER0/7/19/20. 1: The Cortex-M33 LOCKUP output is connected from the break input of TIMER0/7/19/20.

1.6.8. System status register (SYSCFG_STAT)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											NMIPINIF	CKMNMIIF	FLASHECCIF	SRAMECCSEIF	SRAMECCMEIF
											r	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	NMIPINIF	Interrupt flag from NMI pin 0: no input 1: interrupt input from nmi pin
3	CKMNMIIF	HXTAL clock moniotor NMI interrupt flag The software can clear it by writing 1. 0: no HXTAL clock moniotor error 1: HXTAL clock moniotor is detected.
2	FLASHECCIF	Flash ECC NMI interrupt flag The software can clear it by writing 1. 0: no Flash ECC error 1: Flash ECC error is detected.
1	SRAMECCSEIF	SRAM single bit correction error interrupt flag

		The software can clear it by writing 1. 0: no SRAM single bit correction error event is detected. 1: SRAM single bit correction error event is detected.
0	SRAMECCMEIF	SRAM multi-bits (two bits) non-correction error interrupt flag The software can clear it by writing 1. 0: no SRAM non-correction error event is detected. 1: SRAM non-correction error event is detected. **Note:** SRAM multi-bits (two bits) non-correction ECC error will cause a NMI interrupt when the SRAMECCMEIE bit is set.

1.6.9. System configuration register 3 (SYSCFG_CFG3)

Address offset: 0x28
Reset value: 0xXXXX X00F

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAMECCEADDR[13:0]														SRAMECCSERRBITS[5:4]	
r														r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAMECCSERRBITS[3:0]				Reserved							NMIPINIE	CKMNMIIE	FLASHECCIE	SRAMECCSEIE	SRAMECCMEIE
r											rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:18	SRAMECCEADDR[13:0]	Record the faulting system address (SRAMECCEADDR = SRAM address[15:0] >> 2) where the last SRAM ECC event on SRAM occurred.
17:12	SRAMECCSERRBITS[5:0]	Which one bit has an SRAM ECC single-bit correctable error 0: no error 1: bit 0 ... 32: bit 31
11:5	Reserved	Must be kept at reset value.
4	NMIPINIE	NMI pin interrupt enable 0: disable 1: enable
3	CKMNMIIE	HXTAL clock moniotor NMI interrupt enable 0: disable 1: enable
2	FLASHECCIE	Flash ECC NMI enable

		0: disable 1: enable
1	SRAMECCSEIE	SRAM single bit correction error interrupt enable 0: SRAM single bit correction error interrupt is disabled. 1: SRAM single bit correction error interrupt is enabled.
0	SRAMECCMEIE	SRAM multi-bits (two bits) non-correction error NMI interrupt enable 0: SRAM non-correction error NMI interrupt is disabled. 1: SRAM non-correction error NMI interrupt is enabled.

1.6.10. TIMER input source select register (SYSCFG_TIMERINSEL)

Address offset: 0x2C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER0_ETI_SEL [1:0]		TIMER7_ETI_SEL [1:0]		Reserved		TIMER19_ETI_SEL [1:0]		TIMER20_ETI_SEL [1:0]		TIMER0_ BRKIN0_ SEL	TIMER0_ BRKIN1_ SEL	TIMER0_ BRKIN2_ SEL	TIMER0_ BRKIN3_ SEL	TIMER7_ BRKIN0_ SEL	TIMER7_ BRKIN1_ SEL
rw		rw				rw		rw		rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER7_ BRKIN2_ SEL	TIMER7_ BRKIN3_ SEL	Reserved				TIMER19 _BRKIN0 _ SEL	TIMER19 _BRKIN1 _ SEL	TIMER19 _BRKIN2 _ SEL	TIMER19 _BRKIN3 _ SEL	TIMER20 _BRKIN0 _ SEL	TIMER20 _BRKIN1 _ SEL	TIMER20 _BRKIN2 _ SEL	TIMER20 _BRKIN3 _ SEL	Reserved	TIMER7_ CH0N_ SEL
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw		rw

Bits	Fields	Descriptions
31:30	TIMER0_ETI_SEL[1:0]	TIMER0 external trigger select 00: timer external trigger 0 01: timer external trigger 1 10: timer external trigger 2 11: Reserved
29:28	TIMER7_ETI_SEL[1:0]	TIMER7 external trigger select 00: timer external trigger 0 01: timer external trigger 1 10: timer external trigger 2 11: Reserved
27:26	Reserved	Must be kept at reset value.
25:24	TIMER19_ETI_SEL[1:0]	TIMER19 external trigger select 00: timer external trigger 0 01: timer external trigger 1

		10: timer external trigger 2 11: Reserved
23:22	TIMER20_ETI_SEL[1:0]	TIMER20 external trigger select 0: timer external trigger 0 1: timer external trigger 1 10: timer external trigger 2 11: Reserved
21	TIMER0_BRKIN0_SEL	TIMER0 break input 0 select 0: from GPIO pin 1: from TRIGSEL
20	TIMER0_BRKIN1_SEL	TIMER0 break input 1 select 0: from GPIO pin 1: from TRIGSEL
19	TIMER0_BRKIN2_SEL	TIMER0 break input 2 select 0: from GPIO pin 1: from TRIGSEL
18	TIMER0_BRKIN3_SEL	TIMER0 break input 3 select 0: from GPIO pin 1: from TRIGSEL
17	TIMER7_BRKIN0_SEL	TIMER7 break input 0 select 0: from GPIO pin 1: from TRIGSEL
16	TIMER7_BRKIN1_SEL	TIMER7 break input 1 select 0: from GPIO pin 1: from TRIGSEL
15	TIMER7_BRKIN2_SEL	TIMER7 break input 2 select 0: from GPIO pin 1: from TRIGSEL
14	TIMER7_BRKIN3_SEL	TIMER7 break input 3 select 0: from GPIO pin 1: from TRIGSEL
13:10	Reserved	Must be kept at reset value.
9	TIMER19_BRKIN0_SEL	TIMER19 break input 0 select 0: from GPIO pin 1: from TRIGSEL
8	TIMER19_BRKIN1_SEL	TIMER19 break input 1 select 0: from GPIO pin 1: from TRIGSEL

7	TIMER19_BRKIN2_SEL	TIMER19 break input 0 select 0: from GPIO pin 1: from TRIGSEL
6	TIMER19_BRKIN3_SEL	TIMER19 break input 3 select 0: from GPIO pin 1: from TRIGSEL
5	TIMER20_BRKIN0_SEL	TIMER20 break input 0 select 0: from GPIO pin 1: from TRIGSEL
4	TIMER20_BRKIN1_SEL	TIMER20 break input 1 select 0: from GPIO pin 1: from TRIGSEL
3	TIMER20_BRKIN2_SEL	TIMER20 break input 2 select 0: from GPIO pin 1: from TRIGSEL
2	TIMER20_BRKIN3_SEL	TIMER20 break input 3 select 0: from GPIO pin 1: from TRIGSEL
1	Reserved	Must be kept at reset value.
0	TIMER7_CH0N_SEL	TIMER7 Channel 0 complementary input select 0: TIMER7_CH0N_IN 1: exclusive OR of TIMER7_CH0_IN,TIMER7_CH0N_IN,and TIMER0_CH0_IN

1.7. Device electronic signature

The device electronic signature contains memory density information and the 96-bit unique device ID. It is stored in the information block of the Flash memory. The 96-bit unique device ID is unique for any device. It can be used as serial numbers, or part of security keys, etc.

1.7.1. Memory density information

Base address: 0x1FFF F7E0
The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM_DENSITY[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_DENSITY[15:0]															

r

Bits	Fields	Descriptions
31:16	SRAM_DENSITY	SRAM density
	[15:0]	The value indicates the on-chip SRAM density of the device in Kbytes.
		Example: 0x0008 indicates 8 Kbytes.
15:0	FLASH_DENSITY	Flash memory density
	[15:0]	The value indicates the Flash memory density of the device in Kbytes.
		Example: 0x0020 indicates 32 Kbytes.

1.7.2. Unique device ID (96 bits)

Base address: 0x1FFF F7E8

The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[15:0]															
r															

Bits	Fields	Descriptions
31:0	UNIQUE_ID[31:0]	Unique device ID

Base address: 0x1FFF F7EC

The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[63:48]															
r															

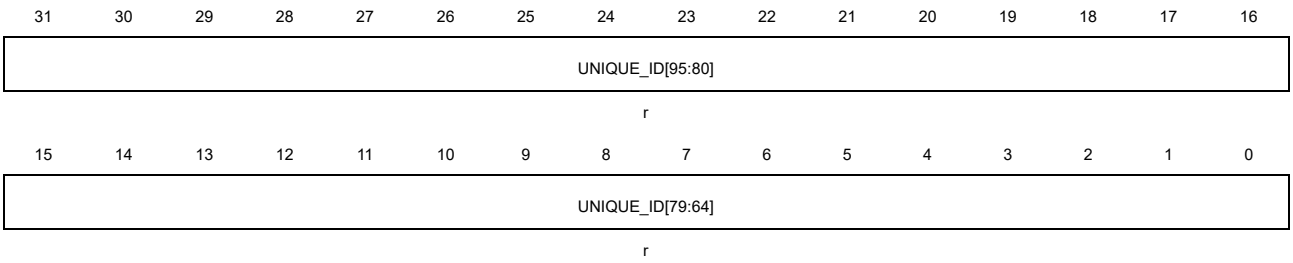
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[47:32]															
r															

Bits	Fields	Descriptions
31:0	UNIQUE_ID[63:32]	Unique device ID

Base address: 0x1FFF F7F0

The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:0	UNIQUE_ID[95:64]	Unique device ID

2. Flash memory controller (FMC)

2.1. Overview

The flash memory controller, FMC, provides all the necessary functions for the on-chip flash memory. A little waiting time is needed while CPU executes instructions stored from the 384K bytes of the flash. It also provides page erase, mass erase, and program operations for flash memory.

2.2. Characteristics

- Up to 384KB of on-chip flash memory for instruction and data. Up to 1KB OTP. Up to 64KB Extend flash.
  - bank0: 256KB
  - bank1: 128KB
  - Extend Block: 64KB, which can be used for data flash
  - OTP: 1KB
  - Shared RAM: 4KB used for basic SRAM or fast program buffer
- Dual bank architecture for read-while-write (RWW) capability.
- ECC with single bit error corrected and double bit errors detected.
- 0~3 waiting time within bank0 / bank1 / Data Flash when CPU executes instructions and read data.
- Pre-fetch buffer to speed read operations.
- Cache with 1K bytes which organized as 32 cache line of 4X64 bits.
- The flash page size is 1KB.
- Double word programming, page erase and mass erase operation.
- 1KB OTP (one-time program) block used for user data storage.
- 24B option bytes block for user application requirements.
- 4B option bytes 1.
- Option bytes are uploaded to the option byte control registers when the system is reset.
- Flash security protection to prevent illegal code / data access.
- Page erase / program protection to prevent unexpected operation.
- Fast program support.

2.3. Function overview

2.3.1. Flash memory architecture

The flash memory consists of up to 384 KB main flash, which is organized into 384 pages with 1KB capacity, an 18 KB information block for the boot loader, up to 64KB Extend Flash Block which can be configured as Data Flash. Each page of main flash memory can be erased

individually. *Table 2-1. Base address and size for 384 KB flash memory* shows the base address and size.

Table 2-1. Base address and size for 384 KB flash memory

Block		Name	Address	size(bytes)
Main Flash Block	bank0	Page 0	0x0800 0000 - 0x0800 03FF	1KB
		Page 1	0x0800 0400 - 0x0800 07FF	1KB
		Page 2	0x0800 0800 - 0x0800 0BFF	1KB
		.	.	.
		.	.	.
		.	.	.
		Page 255	0x0803 FC00 - 0x0803 FFFF	1KB
	bank1	Page 256	0x0804 0000 - 0x0804 03FF	1KB
		Page 257	0x0804 0400 - 0x0804 07FF	1KB
		Page 258	0x0804 0800 - 0x0804 0BFF	1KB
		.	.	.
		.	.	.
		.	.	.
		Page 383	0x0805 FC00 - 0x0805 FFFF	1KB
Extend Flash Block		Data Flash(1)	0x0880 0000 - 0x0880 FFFF	64KB
Shared RAM		Fast program SRAM	-	4KB
		Basic SRAM	Refer to *Table 1-2. Memory map of GD32A50x devices*.	
Information Block		Bootloader(2)	0x1FFF B000 - 0x1FFF F7FF	18KB
Option byte Block		Option bytes 0	0x1FFF F800 - 0x1FFF F817	24B
		Option bytes 1	0x4002 2068	4B
One-time program Block		OTP bytes(3)	0x1FFF 7000~0x1FFF 73FF	1KB

Note: 1. When the Extend Flash Block is used for Data Flash, it must be configured by option bytes1.

2. The Information Block stores the boot loader. This block cannot be programmed or erased by user.

3. 1 Kbyte (128 double word) OTP (one-time programmable) data area is for user, the OTP area is only operated by bank1 register. The OTP data cannot be erased and can be written only once. If any bit has been set 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000 0000.

Table 2-2. Base address and size for 256 KB flash memory

Block		Name	Address	size(bytes)
Main Flash Block	bank0	Page 0	0x0800 0000 - 0x0800 03FF	1KB
		Page 1	0x0800 0400 - 0x0800 07FF	1KB
		Page 2	0x0800 0800 - 0x0800 0BFF	1KB

Block		Name	Address	size(bytes)
		. . .	. . .	. . .
		Page 255	0x0803 FC00 - 0x0803 FFFF	1KB
Extend Flash Block		Data Flash(1)	0x0880 0000 - 0x0880 FFFF	64KB
Shared RAM		Fast program SRAM	-	4KB
		Basic SRAM	Refer to ***Table 1-2. Memory map of GD32A50x devices***.	
Information Block		Bootloader(2)	0x1FFF B000 - 0x1FFF F7FF	18KB
Option byte Block		Option bytes 0	0x1FFF F800 - 0x1FFF F817	24B
		Option bytes 1	0x4002 2068	4B
One-time program Block		OTP bytes(3)	0x1FFF 7000~0x1FFF 73FF	1KB

Note: 1. When the Extend Flash Block is used for Data Flash, it must be configured by option bytes1.

2. The Information Block stores the boot loader. This block cannot be programmed or erased by user.

3. 1 Kbyte (128 double word) OTP (one-time programmable) data area is for user, the OTP area is only operated by bank 1 register. The OTP data cannot be erased and can be written only once. If any bit has been set 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000 0000.

Table 2-3. Base address and size for 128KB flash memory

Block		Name	Address	size(bytes)
Main Flash Block	bank0	Page 0	0x0800 0000 - 0x0800 03FF	1KB
		Page 1	0x0800 0400 - 0x0800 07FF	1KB
		Page 2	0x0800 0800 - 0x0800 0BFF	1KB
		. . .	. . .	. . .
		Page 127	0x0801 FC00 - 0x0801 FFFF	1KB
Extend Flash Block		Data Flash(1)	0x0880 0000 - 0x0880 7FFF	32KB
Shared RAM		Fast program SRAM	-	2KB
		Basic SRAM	Refer to ***Table 1-2. Memory map of GD32A50x devices***.	
Information Block		Bootloader(2)	0x1FFF B000 - 0x1FFF F7FF	18KB
Option byte Block		Option bytes 0	0x1FFF F800 - 0x1FFF F817	24B
		Option bytes 1	0x4002 2068	4B
One-time program Block		OTP bytes(3)	0x1FFF 7000~0x1FFF 73FF	1KB

Note: 1. When the Extend Flash Block is used for Data Flash, it must be configured by option

bytes1.

2. The Information Block stores the boot loader. This block cannot be programmed or erased by user.

3. 1 Kbyte (128 double word) OTP (one-time programmable) data area is for user, the OTP area is only operated by bank 1 register. The OTP data cannot be erased and can be written only once. If any bit has been set 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000 0000.

2.3.2. Error Checking and Correcting (ECC)

The ECC mechanism supports:

- One error detection and correction
- Two errors detection

When one error is detected and corrected:

- When occurred in option bytes 0 (load option bytes 0 to register after reset) / option bytes 1, the error is corrected without any notice.
- When occurred in other space including read option bytes 0 by 0x1FFFF80x, the ECCCOR bit in FMC_ECCCS register will be set. If the ECCCORIE bit in FMC_ECCCS register is set, an interrupt is generated. The OTP_ECC / DF_ECC / SYS_ECC / BK1_ECC / OB0_ECC notice the space that error occurred. The ECCADDR notice the offset address in each space.

When two errors are detected:

- When occurred in option bytes 0 (load option bytes 0 to register after reset), the OB0ECCDET bit in FMC_ECCCS register will be set. If the ECCDETIE bit in FMC_ECCCS register is set, an interrupt is generated. And the data return all F.
- When occurred in option bytes 1, the OB1ECCDET bit in FMC_ECCCS register will be set. If the ECCDETIE bit in FMC_ECCCS register is set, an interrupt is generated. And the data return all F.
- When occurred in other space including read option bytes 0 at 0x1FFFF80x, the ECCDET bit in FMC_ECCCS register and FLASHECCIF in SYSCFG_STAT register will be set. And the NMI interrupt will be generated if FLASHECCIE bit in SYSCFG_CFG3 register is set. The OTP_ECC / DF_ECC / SYS_ECC / BK1_ECC / OB0_ECC notice the space which error occurred. The ECCADDR notice the offset address in each space. And the data return all F. The ECCDET bit in FMC_ECCCS register and FLASHECCIF bit in SYSCFG_STAT register can be cleared by writing 1 to ECCDET bit in FMC_ECCCS register or FLASHECCIF in SYSCFG_STAT register.

Note:

1. Data in Flash memory are 72-bits words: 8 bits are added per double word (64 bits), but the added 8bits are calculated by hardware and can not be accessed by user.

2. For a virgin data 0xFF FFFF FFFF FFFF FFFF, ECC is not supported.

3. When the original data programmed is 0xFFFF FFFF FFFF FFFF, the written ECC code is not 0xF. If programming other values again, since the programmed area is not all F, it cannot be written, resulting in the programming error PGERR. The area needs to be erased before performing the programming operation again.

The OTP_ECC / DF_ECC / SYS_ECC / BK1_ECC / OB0_ECC / ECCADDR notice the first error occurred in single bit error and double bit errors. If ECCCOR or ECCDET set, the value will not change even if a new error occurred.

When an ECC error is reported, a new read at the error address may not generate an ECC error if the data is still present in the current buffer / prefetch buffer / cache, even if ECCCOR and ECCDET are cleared.

2.3.3. Read operations

The flash can be addressed directly as a common memory space. Any instruction fetch and the data access from the flash are through the AHB BUS from the CPU.

Wait state added:

The WSCNT bits in the FMC_WS register needs to be configured correctly depend on the AHB clock frequency when reading the flash memory. The relation between WSCNT and AHB clock frequency is show as the ***Table 2-4. The relation between WSCNT and AHB clock frequency when LDO is 1.1V***.

Table 2-4. The relation between WSCNT and AHB clock frequency when LDO is 1.1V

AHB clock frequency	WSCNT configured
<= 25MHz	0 (0 wait state added)
<= 50MHz	1 (1 wait state added)
<= 75MHz	2 (2 wait state added)
<= 100MHz	3 (3 wait state added)

If system reset occurs, the AHB clock frequency is 8MHz and the WSCNT is 0.

Note:

1. If it is necessary to increase the AHB clock frequency, firstly, refer to ***Table 2-4. The relation between WSCNT and AHB clock frequency when LDO is 1.1V***, configure the WSCNT bits according to the target AHB clock frequency. Then, increase the AHB clock frequency to the target frequency. It is forbidden to increase the AHB clock frequency before configuring the WSCNT.

2. If it is necessary to decrease the AHB clock frequency, firstly, decrease the target AHB clock frequency. Then refer to ***Table 2-4. The relation between WSCNT and AHB clock frequency when LDO is 1.1V***, configure the WSCNT bits according the target AHB clock frequency. It is forbidden to configure the WSCNT bits before decrease the AHB clock frequency.

Considering that the wait state is added, the read efficiency is very low (such as add 3 wait state when 100MHz). In order to speed up the read access, there are some functions performed as below.

Current buffer:

The current buffer is always enabled. Each time read from flash memory, 64-bit data will be get and store in current buffer. The CPU only need 32-bit or 16-bit buffer in each read operation. So in the case of sequential code, the next data can get from current buffer without repeat fetch from flash memory.

Pre-fetch buffer:

The pre-fetch buffer is enabled by setting the PFEN bit in the FMC_WS register. In the case of sequential code, when CPU executes the current buffer data (64-bit), 32-bit needs at least 2 clocks and 16-bit needs at least 4 clocks. In this case, pre-fetch the data of next double-word address from flash memory and store to Pre-fetch buffer. So when the CPU finishes the current buffer and needs execute the next data, the pre-fetch buffer hits.

ICODE / DCODE cache:

Cache is enabled by set the IDCEN bit in the FMC_WS register. The cache has 1K bytes which organized as 32 cache lines, each cache lines is 4 x 64bits.

If the data is in cache (cache hit), the CPU read data from cache without any wait state. If the data is not in cache (cache miss) and not in current buffer / Pre-fetch buffer, the cache line fetch data from flash memory and copied it to cache. If all cache line filled, LRU (least recently used) policy used to replace the cache line.

Read flash by DMA is not cacheable.

Read option byte is not cacheable.

2.3.4. Dual bank architecture with read-while-write (RWW) capability

The Flash memory features a dual bank architecture based on bank 0(256 Kbytes) and bank 1 (up to 128 Kbytes) / Bootloader / Data Flash / OTP / Option byte. This architecture supports the RWW (read-while-write) capability. It means that while a read or program operation is performed in a bank, the other bank can be accessed for another operation (read or program) without the need to wait for the end of operation on the first bank.

For Bootloader / Data Flash / OTP / Option byte, the RWW capability is same as bank1.

2.3.5. Unlock the FMC_CTLx register

After reset, the FMC_CTLx(x=0,1) register is not accessible in write mode, except for the OBRLD bit, which is used for reloading the option byte, and the LK bit in FMC_CTLx register is 1. An unlocking sequence consists of two write operations to the FMC_KEYx register can

open the access to the FMC_CTLx register. The two write operations are writing 0x45670123 and 0xCDEF89AB to the FMC_KEYx register. After the two write operations, the LK bit in FMC_CTLx register is cleared by hardware. The software can lock the FMC_CTLx again by setting the LK bit in FMC_CTLx register. If there is any wrong operations on the FMC_KEYx register, the LK bit in FMC_CTLx register will be set, and the FMC_CTLx register will be locked, then it will generate a bus error.

The OB0PG bit and OB0ER bit in FMC_CTL1 are also protected by FMC_OBKEY register. The unlocking sequence includes two write operations, which are writing 0x45670123 and 0xCDEF89AB to FMC_OBKEY register, then the OBWEN bit in FMC_CTL1 register is set by hardware. The software can clear OBWEN bit to protect the OB0PG bit and OB0ER bit in FMC_CTL1 register again.

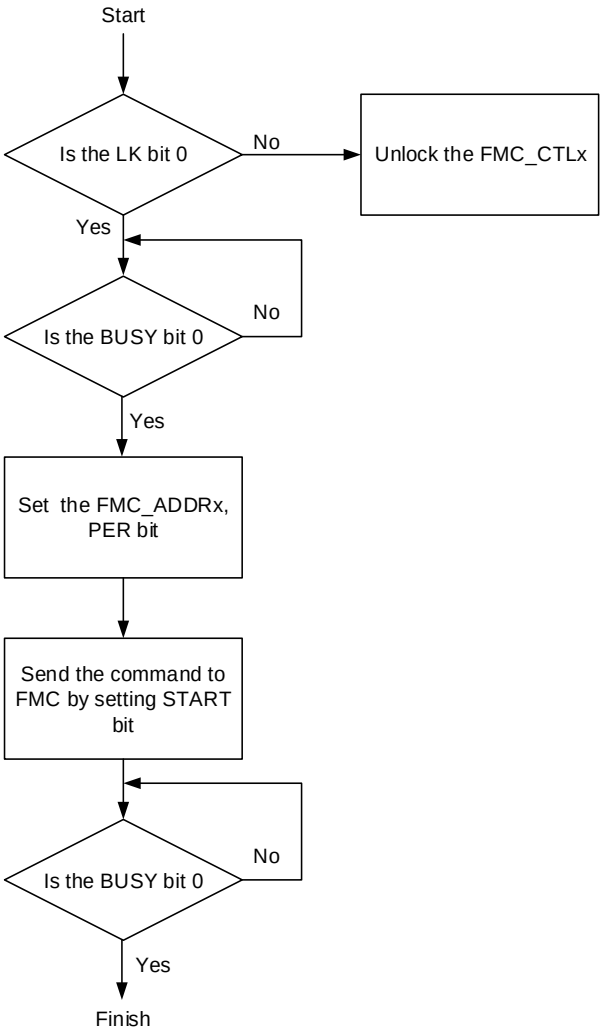
2.3.6. Page erase

The FMC provides a page erase function which is used for initializing the contents of a main flash memory page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the register for a page erase operation.

- Unlock the FMC_CTLx register if necessary.
- Check the BUSY bit in FMC_STATx register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Write the page address into the FMC_ADDRx register.
- Write the page erase command into PER bit in FMC_CTLx register.
- Send the page erase command to the FMC by setting the START bit in FMC_CTLx register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STATx register.
- Read and verify the page if required using a DBUS access.

When the operation is executed successfully, the ENDF in FMC_STATx register will be set. An interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTLx register is set. Note that a correct target page address must be confirmed. Otherwise, the software may run out of control if the target erase page is being used for fetching codes or accessing data. The FMC will not provide any notification when it occurs. Additionally, the page erase operation will be ignored on protected pages. Flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTLx register is set. The software can check the WPERR bit in the FMC_STATx register to detect this condition in the interrupt handler. The ***Figure 2-1. Process of page erase operation*** shows the page erase operation flow.

Figure 2-1. Process of page erase operation



2.3.7. Mass erase

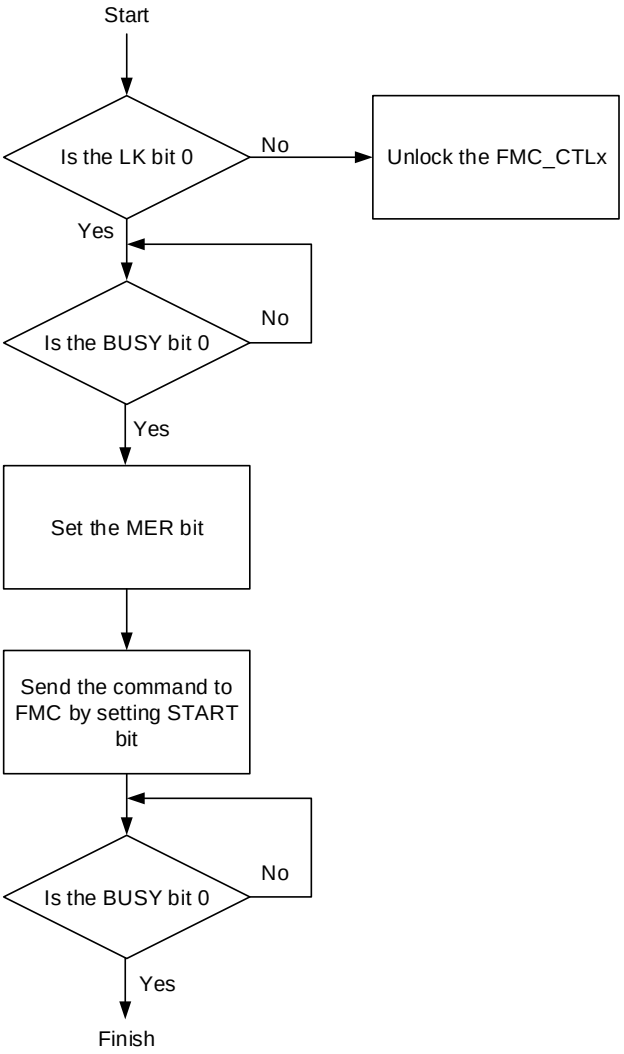
The FMC provides a complete erase function which is used for initializing the Main Flash Block contents. The following steps show the mass erase register access sequence.

- Unlock the FMC_CTLx register if necessary.
- Check the BUSY bit in FMC_STATx register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Write the mass erase command into MER bit in FMC_CTLx register.
- Send the mass erase command to the FMC by setting the START bit in FMC_CTLx register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STATx register.
- Read and verify the flash memory if required using a DBUS access.

When the operation is executed successfully, an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTLx register is set, and the ENDF in FMC_STATx register is set.

Since all flash data will be reset to a value of 0xFFFF FFFF, the mass erase operation can be implemented using a program that runs in SRAM or by using the debugging tool to access the FMC registers directly. Additionally, the mass erase operation will be ignored if any page is erase / program protected. In this condition, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTLx register is set. The software can check the WPERR bit in the FMC_STATx register to detect this condition in the interrupt handler. The *Figure 2-2. Process of mass erase operation* indicates the mass erase operation flow.

Figure 2-2. Process of mass erase operation



Note: The mass erase of bank 1 is split to 128 page erase by hardware, so the mass erase time is longer than bank 0.

2.3.8. Main flash programming

The FMC provides a 32-bit word programming function by DBUS which is used to modify the main flash memory contents. While actually, the data program to flash memory is 64-bits. The following steps show the register access sequence of the programming operation.

- Unlock the FMC_CTLx register if necessary.
- Check the BUSY bit in FMC_STATx register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Write the word program command into the PG bit in FMC_CTLx register.
- Write the data to be programed by DBUS with desired absolute address (0x08XX XXXX). The DBUS write twice to form a 64-bit data and then the 64-bit data program to flash memory. The data to be programed must double-word alignment.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STATx register.
- Read and verify the flash memory if required using a DBUS access.

When the operation is executed successfully, an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTLx register is set, and the ENDF in FMC_STATx register is set. Note that before the double word programming operation you should check the address that it has been erased. If the address has not been erased, PGERR bit will set when programming the address even if programming 0x0. Additionally, the program operation will be ignored on protected pages. Flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTLx register is set. The software can check the PGERR bit in the FMC_STATx register to detect this condition in the interrupt handler.

In the following cases, the PGAERR bit in the FMC_STATx register will be set.

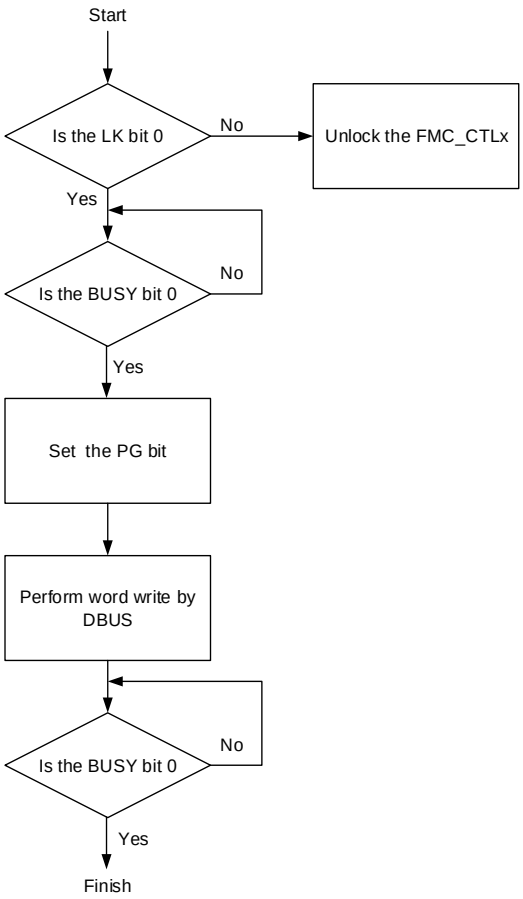
The DBUS program do not use 32-bit write.

The DBUS write is not alignment. If DBUS program is 32-bit, the second DBUS write must double-word alignment and belong to same double-word address.

Note: If the program is not write total 64bits, the data is not program to the flash memory without any notice.

In these conditions, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTLx register is set. The software can check the PGERR bit, PGAERR bit or WPERR bit in the FMC_STATx register to detect which condition occurred in the interrupt handler. The ***Figure 2-3. Process of word program operation*** displays the word programming operation flow.

Figure 2-3. Process of word program operation



Note: Reading the flash should be avoided when a program / erase operation is ongoing in the same bank.

When programming double word, the ECC byte calculated from the 64 bits will be added following the 64 bits, then the total 72 bits will be programmed at a time, even if the double word is 0xFFFF FFFF FFFF FFFF.

If the program / erase operation is interrupted by a power down, reset, ect, the contents in flash will not be guaranteed and leave in an indeterminate state. So appropriate measures should be taken to avoid data loss by interrupt of program / erase.

2.3.9. Main Flash Fast Programming

The FMC provides a fast programming function by DBUS which is used to modify the main flash memory contents. A row (32 double-word) could be programmed to main flash memory in this mode to reduce the page programming time by eliminating the need for verifying the flash locations before they are programmed and to avoid rising and falling time of high voltage for each word.

The following steps show the register access sequence of the programming operation.

- 1. Set Shared RAM to fast program mode by configuring the SRAMCMD bits to "01".

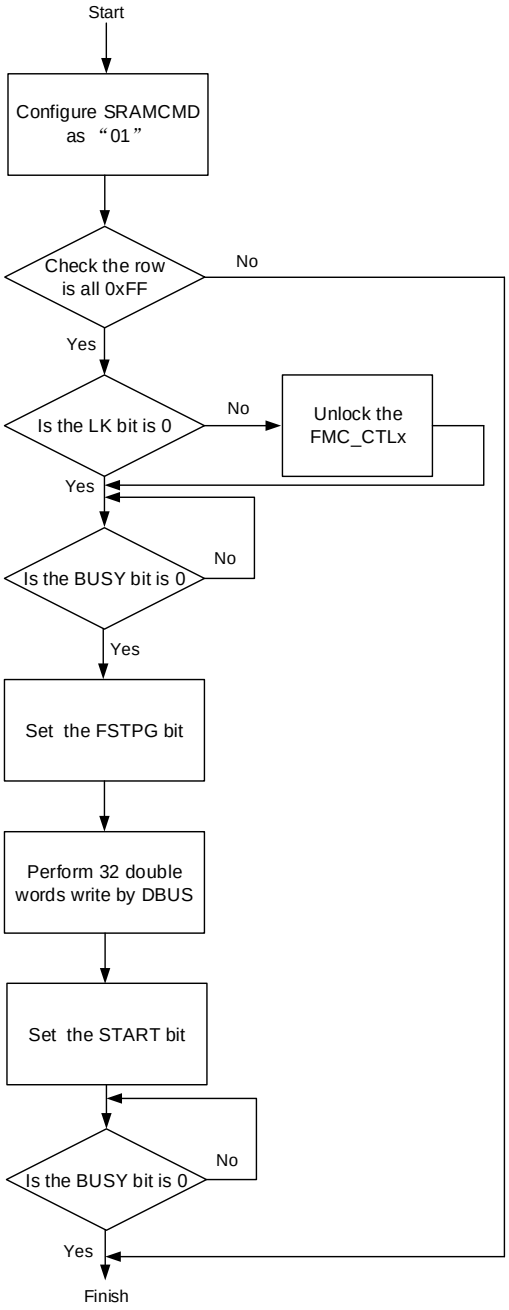
- 2. Check the row (32 double-word) in flash to confirm all data in flash is all 0xFF. The check blank command can be used to check the page the row in.
- 3. Unlock the FMC_CTLx register if necessary.
- 4. Check the BUSY bit in FMC_STATx register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- 5. Set the FSTPG bit in FMC_CTLx register.
- 6. Write the one row data (32 double-word) to be programed by BUS with desired absolute address (0x08XX XXXX). The 32 double-word should be written sequentialy and continuously. Or else the PGERR bit in FMC_CTLx register will be set after setting the START bit.
- 7. Set START bit to launch fast program operation to flash.
- 8. Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STATx register.
- 9. Read and verify the Flash memory if required using a BUS access. It is recommended to flush the cache if cache is enabled.

When the operation is executed successfully, the ENDF in FMC_STATx register is set, and an interrupt will be generated if the ENDIE bit in the FMC_CTLx register is set. In sequential fast programming, if the row is confirmed erased, repeat 6~8 to fast program next row.

The program operation will be ignored on erase / program protected pages and WPERR bit in FMC_STATx will be set.

In these conditions, a flash operation error interrupt will be generated if the ERRIE bit in the FMC_CTLx register is set. The software can check the PGSERR / PGAERR / WPERR / PGERR bit in the FMC_STATx register to detect which condition occurred in the interrupt handler. ***Figure 2-4. Process of fast programming operation*** shows the fast programming operation flow.

Figure 2-4. Process of fast programming operation



Note: 1. The 32 double-word must be written successively.

2. The 32 double-word must be aligned.

3. Because the fast program do not check 0xFF in flash macro by hardware, the software must check 0xFF first and don’t program one row twice or more between 2 erases. If program one row twice or more between 2 erases, unpredictable result may occur.

4. Between setting FSTPG and START, read operation is allowed, but the read address should not be the address of the row where to be programmed, otherwise the data read out may be old data.

5. The cache must be flushed before fast programming.

2.3.10. Check blank command

The check blank command is used to check if the flash area which is specified by FMC_ADDRx and CBCMDLEN bits in FMC_CTLx register are all 0xFF or not. Configure the check blank command by setting the CBCMD bit in FMC_CTLx register and send the check blank command to the FMC by setting the START bit in FMC_CTLx register, the BUSY bit will be set, and the hardware will check if the flash area are all 0xFF or not. If the flash area are all 0xFF, the ENDF will be set, and the BUSY bit will be cleared, or else the CBCMDERR will be set. The check blank command only support for bank0 / bank1 / data flash.

Note: The flash area to be checked must be in one page and should not exceed 1KB boundary.

2.3.11. OTP programming

The OTP programming opearation flow is as same as the main flash programming. The OTP block can only be programed once and cannot be erased. The OTP area is only operated by bank 1 registers (FMC_KEY1 / FMC_STAT1 / FMC_CTL1 / FMC_ADDR1).

Note: It must ensure the OTP programming sequence completely without any unexpected interrupt, such as system reset or power down. If unexpected interrupt occurs, there is very little probability of corrupt the data stored in flash memory.

2.3.12. Shared RAM

There are 4KB Shared RAM which can be used for basic SRAM or fast program SRAM.

Basic SRAM

The basic SRAM command is sent by configuring the SRAMCMD bits as "10". After sending the basic SRAM command, the Shared RAM initializes to all zeros. If the basic SRAM is ready, the BRAMRDY bit in FMC_WS register will be set and SRAMCMD will be cleared. Otherwise, wait until the command has been finished. The basic SRAM can be used for system SRAM but without ECC.

Fast program SRAM

The fast program SRAM command is sent by configuring the SRAMCMD bits as "01". After sending the fast program SRAM command, the Shared RAM initializes to all 1. If the fast program SRAM is ready, the PRAMRDY bit in FMC_WS register will be set and SRAMCMD will be cleared. Otherwise, wait until the command has been finished.

Note: When configuring the SRAMCMD bits, it is required to check which SRAM is available currently. The configuration cannot be repeated. For example, if it is currently basic SRAM, the SRAMCMD should not be configured as "10" again. The current type of SRAM can be checked by BRAMRDY / PRAMRDY bits in FMC_WS register.

2.3.13. Data Flash operation

The data flash size is configured by option bytes 1.

Read

The read of Data Flash is same as main flash. Refer to ***Read operations***.

Program

The Program is same as bank 1 program. Refer to ***Main flash programming***.

Page erase

The page erase is same as bank 1 page erase. Refer to ***Page erase***.

Mass erase

The mass erase is same as bank 1 mass erase. Refer to ***Mass erase***.

Note: 1. The mass erase is split to page erase by hardware. The erase time is longer than normal mass erase.

2. The mass erase command is setting MERDF bit in FMC_CTL1 register.

Data flash fast program

The fast program of data flash is same as bank 1 fast program. Refer to ***Main Flash Fast Programming***.

2.3.14. Option bytes 0 erase

The FMC provides an erase function which is used to initialize the option bytes 0 block in flash. The following steps show the erase sequence.

- Unlock the FMC_CTL1 register if necessary.
- Check the BUSY bit in the FMC_STAT1 register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has finished.
- Unlock the option bytes operation bit OBWEN in the FMC_CTL1 register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL1 register.
- Set the OB0ER bit in the FMC_CTL1 register.
- Send the option bytes 0 erase command to the FMC by setting the START bit in the FMC_CTL1 register.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT1 register.
- Read and verify the flash memory using a DBUS access if required.

When the operation is executed successful, the ENDF bit in the FMC_STAT1 register is set, and an interrupt will be generated if the ENDIE bit in the FMC_CTL1 register is set.

2.3.15. Option bytes programming

Option bytes 0 programming

The FMC provides a 64-bit double word programming function which is used for modifying the option byte 0 contents. The following steps show the programming operation sequence.

- Unlock the FMC_CTL1 register if necessary.
- Check the BUSY bit in FMC_STAT1 register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Unlock the OBWEN bit in FMC_CTL1 register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL1 register.
- Write the program command into the OB0PG bit in FMC_CTL1 register.
- Write the 64-bit data to be programmed by DBUS with desired absolute address.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT1 register.
- Read and verify the flash memory if required using a DBUS access.

When the operation is executed successfully, an interrupt will be generated if the ENDIE bit in the FMC_CTL1 register is set, and the ENDF in FMC_STAT1 register is set. Note that to check the address that it has been erased before the word programming operation. If the address has not been erased, PGERR bit in the FMC_STAT1 register will be set when programming the address even if programming 0x0. The end of this operation is indicated by the ENDF bit in the FMC_STAT1 register.

Option bytes 1 programming

The following steps show the modify operation sequence.

- Unlock the FMC_CTL1 register if necessary.
- Check the BUSY bit in FMC_STAT1 register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Unlock the OBWEN bit in FMC_CTL1 register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL1 register.
- Write the value to LKVAL / EPLOAD / EPSIZE / EFALC in FMC_OB1CS register.
- Write the OB1START in FMC_OB1CS register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT1 register.
- Lauch a system reset or setting the OBRLD bit in FMC_CTL1 register before accessing data flash.
- Read and verify the flash memory if necessary.

Note:

1. The option bytes 1 block of flash memory reloaded to FMC_OB1CS registers after each system reset or OBRLD bit set in FMC_CTL1 register.

2. The modified option bytes 1 effect only after a system reset or setting the OBRLD bit in FMC_CTL1 register.

3. After the system reset or setting the OBRLD bit in FMC_CTL1 register, if the LKVAL is 0x33CC, the OB1LK bit in FMC_OB1CS register will be set and the option bytes 1 cannot be modified any more.

4. If the unexpected value is written in option bytes 1, the PGSERR bit will be set after setting the OB1START bit and the operation will be ignored.

5. To modify the EPSIZE / EFALC bits, the extend flash will be erased by hardware first, then modify the option bytes 1.

6. After modifying the option byte 1, it is strongly recommended to perform a power reset.

2.3.16. Option bytes description

Option bytes 0 description

The option bytes 0 block of flash memory reloaded to FMC_OBSTAT and FMC_WP0 registers after each system reset or OBRLD bit set in FMC_CTL1 register, and the option bytes 0 work. The option complement bytes are the opposite of option bytes 0. When option bytes 0 reload, if the option complement bytes and option bytes 0 does not match, the OBERR bit in FMC_OBSTAT register is set, and the option byte is set to 0xFF. The ***Table 2-5. Option bytes 0*** is the detail of option bytes 0.

Table 2-5. Option bytes 0

Address	Name	Description
0x1fff f800	OB_SPC	option bytes 0 security protection value 0xA5 : no security protection any value except 0xA5 or 0xCC : protection level low 0xCC : protection level high
0x1fff f801	OB_SPC_N	OB_SPC complement value
0x1fff f802	OB_USER	[7]: BOR_TH (Brown out reset threshold) 0: Have BOR, brownout threshold level (2.6V) (factory value) 1: No BOR [6:5]: reserved [4]: OTA 0: When configured boot from main memory, if the BB is 0, all data will be copied from bank1 to bank0 and then boot from bank0. 1: no effect. [3]: BB

Address	Name	Description
		0: when OTA is 1, boot from bank1 or bank0 if bank1 is void, when configured boot from main memory 1: boot from bank0, when configured boot from main memory [2]: nRST_STDBY 0: generate a reset instead of entering standby mode 1: no reset when entering standby mode [1]: nRST_DPSLP 0: generate a reset instead of entering deep-sleep mode 1: no reset when entering deep-sleep mode [0]: nWDG_HW 0: hardware free watchdog 1: software free watchdog
0x1fff f803	OB_USER_N	OB_USER complement value
0x1fff f804	OB_DATA[7:0]	user defined data bit 7 to 0
0x1fff f805	OB_DATA_N[7:0]	OB_DATA complement value bit 7 to 0
0x1fff f806	OB_DATA[15:8]	user defined data bit 15 to 8
0x1fff f807	OB_DATA_N[15:8]	OB_DATA complement value bit 15 to 8
0x1fff f808	OB_BK0WP[7:0]	Page erase / program protection of bank0 bit 7 to 0 0: protection active 1: unprotected
0x1fff f809	OB_BK0WP_N[7:0]	OB_BK0WP complement value bit 7 to 0
0x1fff f80a	OB_BK0WP[15:8]	Page erase / program protection of bank0 bit 15 to 8
0x1fff f80b	OB_BK0WP_N[15:8]	OB_BK0WP complement value bit 15 to 8
0x1fff f80c	OB_BK0WP[23:16]	Page erase / program protection of bank0 bit 23 to 16
0x1fff f80d	OB_BK0WP_N[23:16]	OB_BK0WP complement value bit 23 to 16
0x1fff f80e	OB_BK0WP[31:24]	Page erase / program protection of bank0 bit 31 to 24
0x1fff f80f	OB_BK0WP_N[31:24]	OB_BK0WP complement value bit 31 to 24
0x1fff f810	OB_BK1WP[7:0]	Page erase / program protection of bank1 bit 7 to 0
0x1fff f811	OB_BK1WP_N[7:0]	OB_BK1WP complement value bit 7 to 0
0x1fff f812	OB_DFWP[7:0]	Page erase / program protection of data flash bit 7 to 0
0x1fff f813	OB_DFWP_N[7:0]	OB_DFWP complement value bit 7 to 0
0x1fff f814	Reserved	Must be kept at 0xFF. Or else, it may cause unpredictable consequence.
0x1fff f815	Reserved	Complement value bit 7 to 0 of the value saved at address 0x1fff f814.

Option bytes 1 description

Table 2-6. Option bytes 1 shows the detail of option bytes 1 in different flash memory capacity.

Table 2-6. Option bytes 1

<table>
<tr><th>Address</th><th>Name</th><th>Description</th></tr>
<tr><td>[31:16]</td><td>LKVAL[15:0]</td><td>Option bytes 1 Lock Value
0x33CC: Lock Option bytes 1, the Option bytes 1 cannot be modified any more.
other value : Do not lock Option bytes 1</td></tr>
<tr><td>[15]</td><td>EPLOAD</td><td>Must be configured as 0. Or else, it may cause unpredictable consequence.</td></tr>
<tr><td>[11:8]</td><td>EPSIZE[3:0]</td><td>Must be configured as 0xF. Or else, it may cause unpredictable consequence.</td></tr>
<tr><td>[7:4]</td><td>EFALC[3:0]</td><td>Default value is 0xF.
When the extend memory is used for data flash, these bits must be configured as follows according to flash memory capacity. If other value is configured, it may cause unpredictable consequence.
<table>
<tr><th>EFALC</th><th>Flash Memory Capacity (KB)</th><th>Data flash size (KB)</th></tr>
<tr><td>0000</td><td>384/256</td><td>64</td></tr>
<tr><td>0001</td><td>128</td><td>32</td></tr>
<tr><td>1111</td><td colspan="2">Default value. The extended flash is not configured with a valid partition code.</td></tr>
</table>
</td></tr>
</table>

Note:

1. Bits 63:32 is the complement byte of Bits 31:0. When option bytes 1 reload, if the option complement byte and option byte does not match, the OB1ERR bit in FMC_OB1CS register is set.

2. When the size and partition is modified, the whole extend flash will be erased by hardware.

3. If the modification of size and partition is illegal, the operation will be ignored and an error occurs.

4. It is recommended to configure the option bytes 1 only once throughout the life cycle. The modification of option bytes 1 takes effect after a system reset, or setting the OBRLD bit in FMC_CTL1 register.

5. It is recommended to do a power reset after the modification of option bytes 1.

2.3.17. Erase / program protection

The FMC provides page erase / program protection functions to prevent inadvertent operations on the flash memory. The page erase or program will not be accepted by the FMC on protected pages. If the page erase or program command is sent to the FMC on a protected page, the WPERR bit in the FMC_STATx register will then be set by the FMC. If the WPERR bit is set and the ERRIE bit is set, then the flash operation error interrupt will be generated to get the attention of the CPU.

Page erase / program protection of bank 0

The page erase / program protection of bank 0 can be individually enabled by configuring the OB_BK0WP[31:0] bit field to 0 in the option bytes 0. If a page erase operation is executed on the Option Byte region, all the flash memory page erase / program protection functions will be disabled. When setting or resetting OB_BK0WP[31:0] in the option bytes 0, the software need to set OBRLD in FMC_CTL1 register or a system reset to reload the OB_BK0WP[31:0] bits. The ***Table 2-7. OB_BK0WP bit for pages protected*** shows which pages are protected by setting OB_BK0WP[31:0].

Table 2-7. OB_BK0WP bit for pages protected

OB_BK0WP bit	pages protected
OB_BK0WP[0]	BANK0_SIZE / 32
OB_BK0WP[1]	BANK0_SIZE / 32
OB_BK0WP[2]	BANK0_SIZE / 32
. . .	. . .
OB_BK0WP[30]	BANK0_SIZE / 32
OB_BK0WP[31]	BANK0_SIZE / 32

Note: BANK0_SIZE is the memory size of bank0.

Page erase / program protection of bank 1

The page erase / program protection of bank 1 can be individually enabled by configuring the OB_BK1WP[7:0] bit field to 0 in the option bytes 0. Each bit of OB_BK1WP[7:0] represents one eighth of bank 1. If a page erase operation is executed on the Option Byte region, all the flash memory page erase / program protection functions will be disabled. When setting or resetting OB_BK1WP[7:0] in the option bytes 0, the software need to set OBRLD in FMC_CTL1 register or a system reset to reload the OB_BK1WP[7:0] bits. The ***Table 2-8. OB_BK1WP bit for pages protected*** shows which pages are protected by setting OB_BK1WP[7:0].

Table 2-8. OB_BK1WP bit for pages protected

OB_BK1WP bit	pages protected
OB_BK1WP[0]	BANK1_SIZE / 8
OB_BK1WP[1]	BANK1_SIZE / 8
. . .	. . .
OB_BK1WP[6]	BANK1_SIZE / 8
OB_BK1WP[7]	BANK1_SIZE / 8

Note: 1. BANK1_SIZE is the memory size of bank1.

2. OTP write protection is controlled by OB_BK1WP[7].

Page erase / program protection of data flash

The page erase / program protection of data flash can be individually enabled by configuring the OB_DFWP[7:0] bit field to 0 in the option bytes 0. Each bit of OB_DFWP[7:0] represents one eighth of data flash. If a page erase operation is executed on the Option Byte region, all the flash memory page erase / program protection functions will be disabled. When setting or resetting OB_DFWP[7:0] in the option bytes 0, the software need to set OBRLD in FMC_CTL1 register or a system reset to reload the OB_DFWP[7:0] bits. The ***Table 2-9. OB_DFWP bit for pages protected*** shows which pages are protected by setting OB_DFWP[7:0].

Table 2-9. OB_DFWP bit for pages protected

OB_DFWP bit	pages protected
OB_DFWP[0]	DFLASH_SIZE / 8
OB_DFWP[1]	DFLASH_SIZE / 8
. . .	. . .
OB_DFWP[6]	DFLASH_SIZE / 8
OB_DFWP[7]	DFLASH_SIZE / 8

Note: DFLASH_SIZE is the memory size of data flash.

2.3.18. Security protection

The FMC provides a security protection function to prevent illegal code / data access on the flash memory. This function is useful for protecting the software / firmware from illegal users. There are 3 levels for protection:

No protection: when setting OB_SPC byte and its complement value to 0x5AA5, no protection performed after a system reset. The main flash and option bytes 0 block are accessible by all operations.

Low level protection: when setting OB_SPC byte and its complement value to any value except 0x5AA5 or 0x33CC, low level protection performed. The main flash, OTP, data flash can only be accessed by user code. In debug mode, or boot from boot loader mode, all operations to main flash, OTP, data flash is forbidden. If a read operation is executed to main flash in debug mode or boot from boot loader mode, a bus error will be generated. If a program / erase / check blank operation is executed to main flash, OTP, data flash in debug mode or boot from bootloader mode, the WPERR bit in FMC_STATx register will be set. At low level protection, option bytes 0 block are accessible by all operations. If program back to no protection level by setting OB_SPC byte and its complement value to 0x5AA5, a mass erase for main flash will be performed.

High level protection: when set OB_SPC byte and its complement value to 0x33CC, high level protection performed. When this level is programmed in debug mode or boot from bootloader mode is disabled. The main flash block is accessible by all operations from user code. The option byte cannot be erased, and the OB_SPC byte and its complement value cannot be

reprogrammed. So, if high level protection is programmed, it cannot move back to protection level low or no protection level.

2.3.19. Error description

PGSERR bit in FMC_CTLx register will be set if one of the conditions occurs in ***Table 2-10. PGSERR conditions***.

Table 2-10. PGSERR conditions

Mode	Condition	Operation
program / fast program	PG and FSTPG are cleared	Write data
fast program	1. not write by address order 2. not write from 0 or not write full 32 double-word 3. PRAMRDY is not set	Set START
program	CBCMD / FSTPG / OB0ER / OB0PG / MERDF / MER / PER are not cleared	Set PG
fast program	CBCMD / PG / OB0ER / OB0PG / MERDF / MER / PER are not cleared	Set FSTPG
option bytes 1 modify	CBCMD / FSTPG / OB0ER / OB0PG / MERDF / MER / PER / PG are not cleared	Set OB1START
option bytes 1 modify	not valid EPSIZE / EFALC	Set OB1START
option byte 0 erase	CBCMD / FSTPG / OB0PG / MERDF / MER / PER / PG are not cleared	Set OB0ER
option byte 0 program	CBCMD / FSTPG / OB0ER /MERDF / MER / PER / PG are not cleared	Set OB0PG
mass erase	CBCMD / OB0ER / OB0PG / MERDF / PER / PG are not cleared	Set MER
data flash mass erase	CBCMD / FSTPG / OB0ER / OB0PG / MER / PER / PG are not cleared No Data Flash	Set MERDF
page erase	CBCMD / FSTPG / OB0ER / OB0PG / MERDF / MER / PG are not cleared FMC_ADDRx is not valid address	Set PER
check blank	FMC_ADDRx and CBCMDLEN configure error: 1. exceed 1KB boundary	Set CBCMD

Mode	Condition	Operation
	2. not valid address	
check blank	FSTPG / OBER / OBPG / MERDF / MER / PER / PG are not cleared	Set CBCMD

PGAERR bit in FMC_CTLx register will be set if one of the conditions occurs in ***Table 2-11. PGAERR conditions***.

Table 2-11. PGAERR conditions

Mode	Condition	Operation
program	1. The DBUS program do not use 32-bit write. 2. The DBUS write is not alignment. The first DBUS write must double-word alignment and the second write belong to same double-word address.	Write data
fast program	1. The DBUS program do not use 32-bit write. 2. The DBUS write is not alignment. The first DBUS write must double-word alignment and the second write belong to same double-word address.	Set START

PGERR bit in FMC_CTLx register will be set if one of the conditions occurs in ***Table 2-12. PGERR conditions***.

Table 2-12. PGERR conditions

Mode	Condition	Operation
program	If the program address is not erased	Write data

WPERR bit in FMC_CTLx register will be set if one of the conditions occurs in ***Table 2-13. WPERR conditions***.

Table 2-13. WPERR conditions

Mode	Condition	Operation
program	1.The program address is write protected by option byte 2. read protection low and boot from bootloader or debug mode	Write data
erase	1.The erase address is write protected by option byte 2. read protection low and boot from bootloader or debug mode	set START
check blank	read protection low and boot from bootloader or debug mode	set START

2.4. Register definition

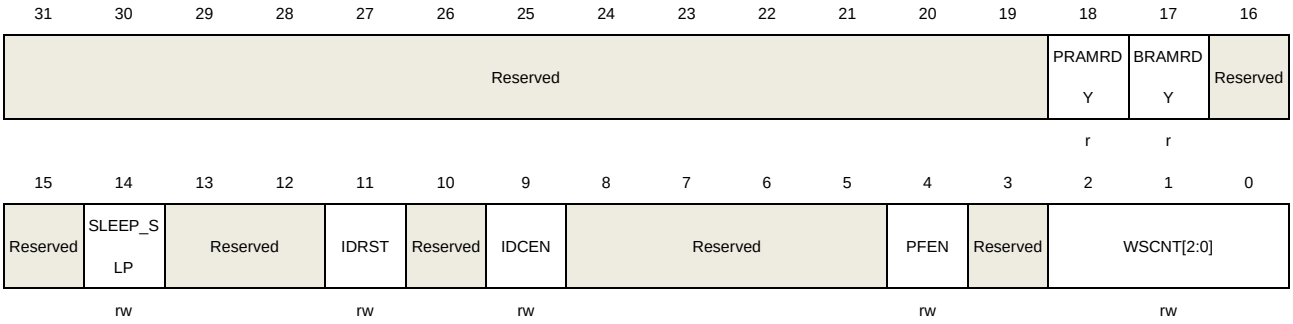
FMC base address: 0x4002 2000

2.4.1. Wait state register (FMC_WS)

Address offset: 0x00

Reset value: 0x0000 0210

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	PRAMRDY	Fast program SRAM ready flag. This bit is set by hardware. And cleared by configuring the SRAMCMD bits as Basic RAM. 0: Fast program SRAM is not ready. 1: Fast program SRAM is ready.
17	BRAMRDY	Basic SRAM ready flag. This bit is set by hardware. And cleared by configuring the SRAMCMD bits as fast program RAM. 0: Basic SRAM is not ready. 1: Basic SRAM is ready.
16:15	Reserved	Must be kept at reset value.
14	SLEEP_SLP	Flash goto sleep mode or power-down mode when MCU enters deepsleep mode. 0: Flash is in power-down mode. 1: Flash is in sleep mode.
13:12	Reserved	Must be kept at reset value.
11	IDRST	Cache reset 0: No reset 1: Reset the cache if cache is disabled
10	Reserved	Must be kept at reset value.
9	IDCEN	Cache enable 0: Cache disable

		1: Cache enable
8:5	Reserved	Must be kept at reset value.
4	PFEN	Pre-fetch enable 0: Pre-fetch disable 1: Pre-fetch enable
3	Reserved	Must be kept at reset value.
2:0	WSCNT[2:0]	Wait state counter These bits are configured by software. 000: 0 wait state added 001: 1 wait state added 010: 2 wait state added 011: 3 wait state added 100 ~111: reserved

2.4.2. ECC control and status register (FMC_ECCCS)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCDET	ECCCOR	Reserved		OB0ECCDET	OB1ECCDET	ECCDETIE	ECCCORIE	OTP_EC C	DF_ECC	SYS_EC C	BK1_EC C	OB0_EC C	Reserved		
rc_w1	rc_w1			rc_w1	rc_w1	rw	rw	r	r	r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ECCADDR[14:0]														
	r														

Bits	Fields	Descriptions
31	ECCDET	Two bit errors detect flag. This bit set when two bit errors is detected. And cleared by writing 1 to ECCDET bit in FMC_ECCCS register or FLASHECCIF bit in SYSCFG_STAT register. 0: Two ECC errors are not detected. 1: Two ECC errors are detected.
30	ECCCOR	One bit error detected and correct flag. This bit is cleared by writing 1. 0: No ECC error is detected and corrected. 1: An ECC error is detected and corrected.
29:28	Reserved	Must be kept at reset value.
27	OB0ECCDET	Option bytes 0 two bit errors detect flag. This bit is cleared by writing 1.

0: Two ECC errors of option bytes 0 are not detected.
1: Two ECC errors of option bytes 0 are detected.

26	OB1ECCDET	Option bytes 1 two bit errors detect flag. This bit is cleared by writing 1. 0: Two ECC errors of option bytes 1 are not detected. 1: Two ECC errors of option bytes 1 are detected.
25	ECCDETIE	Two bit errors detect interrupt enable. When EPECCDET, OB0ECCDET, or OB1ECCDET is set, and this bit is set, an interrupt will be generated. 0: Two bit errors detect interrupt disable. 1: Two bit errors detect interrupt enable.
24	ECCCORIE	One bit error correct interrupt enable. 0: One bit error correct interrupt disable. 1: One bit error correct interrupt enable.
23	OTP_ECC	If an ECC bit error is detected in OTP, this bit will be set. And the ECCADDR records the offset address of OTP. 0: No ECC error is detected in OTP. 1: An ECC bit error is detected in OTP.
22	DF_ECC	If an ECC bit error is detected in data flash, this bit will be set. And the ECCADDR records the offset address of data flash. 0: No ECC error is detected in data flash. 1: An ECC bit error is detected in data flash.
21	SYS_ECC	If an ECC bit error is detected in system memory, this bit will be set. And the ECCADDR records the offset address of system memory. 0: No ECC error is detected in system memory. 1: An ECC bit error is detected in system memory.
20	BK1_ECC	If an ECC bit error is detected in bank 1, this bit will be set. And the ECCADDR records the offset address of bank 1. 0: No ECC error is detected in bank 1. 1: An ECC bit error is detected in bank 1.
19	OB0_ECC	If an ECC bit error is detected in option bytes 0, this bit will be set. And the ECCADDR records the offset address of option bytes 0. 0: No ECC error is detected in option bytes 0. 1: An ECC bit error is detected in option bytes 0.
18:15	Reserved	Must be kept at reset value.
14:0	ECCADDR[14:0]	The offset address of double word where an ECC error is detected. Error address = base address + ECCADDR[14:0] * 8, the base address can be the start address of bank0, bank1, data flash, system area, option bytes 0, option bytes 1 and OTP. For details, refer to ***2.3.1***.

2.4.3. Unlock key register 0 (FMC_KEY0)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w															

Bits	Fields	Descriptions
31:0	KEY[31:0]	FMC_CTL0 unlock key These bits are only be written by software. Write KEY[31:0] with keys to unlock FMC_CTL0 register.

2.4.4. Status register 0 (FMC_STAT0)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTERR	Reserved								CBCMDERR	ENDF	WPERR	PGAERR	PGERR	PGSERR	BUSY
rc_w1									rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	RSTERR	If the voltage is below 3.0V or a system reset occurs during flash programming or erasing, an error will be generated and this bit will be set. When the error is occurred, the data in the current address unreliable, and it is necessary to erase and program again. If the voltage is lower than BOR / POR, the value of this bit will be reset after the BOR / POR reset, but retained after a system reset. **Note:** Programming/erasing is not recommended when the voltage is below 3.0V.
14:7	Reserved	Must be kept at reset value.
6	CBCMDERR	The checked area by the check blank command is all 0xFF or not. 0: The checked area is all 0xFF.

		1: The checked area is not all 0xFF.
5	ENDF	End of operation flag bit When the operation executed successfully, this bit is set by hardware. The software can clear it by writing 1.
4	WPERR	Erase / Program protection error flag bit When erase / program on protected pages, this bit is set by hardware. The software can clear it by writing 1.
3	PGAERR	Program alignment error flag bit This bit is set by hardware when DBUS write data is not alignment. The software can clear it by writing 1.
2	PGERR	Program error flag bit When program to the flash while it is not 0xFFFF, this bit is set by hardware. The software can clear it by writing 1.
1	PGSERR	Program sequence error flag bit.
0	BUSY	The flash busy flag. When the operation is in progress, this bit is set to 1. When the operation is end or an error is generated, this bit is cleared to 0.

2.4.5. Control register 0 (FMC_CTL0)

Address offset: 0x10

Reset value: 0x0000 0080

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CBCMDLEN[2:0]			Reserved												CBCMD
	rw														rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			ENDIE	Reserved	ERRIE	Reserved	FSTPG	LK	START	Reserved			MER	PER	PG
			rw		rw		rw	rs	rs				rw	rw	rw

Bits	Fields	Descriptions
31:29	CBCMDLEN[2:0]	CBCMD read length 2^(CBCMDLEN). The read length by check blank command. The read length is 2^ CBCMDLEN double words.
28:17	Reserved	Must be kept at reset value.
16	CBCMD	The command to check the selected area is blank or not.
15:13	Reserved	Must be kept at reset value.
12	ENDIE	End of operation interrupt enable bit

		This bit is set or cleared by software 0: no interrupt generated by hardware. 1: end of operation interrupt enable
11	Reserved	Must be kept at reset value.
10	ERRIE	Error interrupt enable bit This bit is set or clear by software0: no interrupt generated by hardware 1: error interrupt enable
9	Reserved	Must be kept at reset value.
8	FSTPG	Main flash fast program command bit This bit is set or clear by software 0: no effect 1: main flash fast program command
7	LK	FMC_CTL0 lock bit This bit is cleared by hardware when right sequence written to the FMC_KEY0 register. This bit can be set by software.
6	START	Send erase command to FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
5:3	Reserved	Must be kept at reset value.
2	MER	Main flash mass erase command bit This bit is set or cleared by software 0: no effect 1: main flash mass erase command
1	PER	Main flash page erase command bit This bit is set or clear by software 0: no effect 1: main flash page erase command
0	PG	Main flash program command bit This bit is set or clear by software 0: no effect 1: main flash program command

Note: This register should be reset after the corresponding flash operation completed.

2.4.6. Address register 0 (FMC_ADDR0)

Address offset: 0x14
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
w															

Bits	Fields	Descriptions
31:0	ADDR[31:0]	Flash erase / program command address bits These bits are configured by software. ADDR bits are the address of flash to be erased / programmed.

2.4.7. Option byte unlock key register (FMC_OBKEY)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBKEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBKEY[15:0]															
w															

Bits	Fields	Descriptions
31:0	OBKEY[31:0]	FMC_CTL1 option bytes operation unlock register These bits are only be written by software. Write OBKEY[31:0] with keys to unlock option bytes command in the FMC_CTL1 register.

2.4.8. Unlock key register 1 (FMC_KEY1)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w															

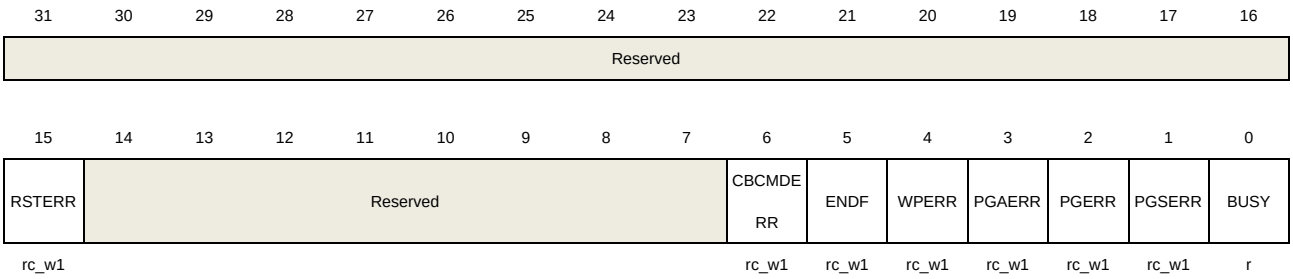
Bits	Fields	Descriptions
31:0	KEY[31:0]	FMC_CTL1 unlock key These bits are only be written by software Write KEY[31:0] with key to unlock FMC_CTL1 register.

2.4.9. Status register 1 (FMC_STAT1)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	RSTERR	If the voltage is below 3.0V or a system reset occurs during flash programming or erasing, an error will be generated and this bit will be set. When the error is occurred, the data in the current address unreliable, and it is necessary to erase and program again. If the voltage is lower than BOR / POR, the value of this bit will be reset after the BOR / POR reset, but retained after a system reset. **Note:** Programming/erasing is not recommended when the voltage is below 3.0V.
14:7	Reserved	Must be kept at reset value.
6	CBCMDERR	The checked page by the check blank command is all 0xFF or not. 0: The checked page is all 0xFF. 1: The checked page is not all 0xFF.
5	ENDF	End of operation flag bit When the operation executed successfully, this bit is set by hardware. The software can clear it by writing 1.
4	WPERR	Erase / Program protection error flag bit When erase / program on protected pages, this bit is set by hardware. The software can clear it by writing 1.
3	PGAERR	Program alignment error flag bit This bit is set by hardware when DBUS write data is not alignment. The software can clear it by writing 1.
2	PGERR	Program error flag bit

		When program to the flash while it is not 0xFFFF, this bit is set by hardware. The software can clear it by writing 1.
1	PGSERR	Program sequence error flag bit.
0	BUSY	The flash is busy bit When the operation is in progress, this bit is set to 1. When the operation is end or an error is generated, this bit is cleared to 0.

2.4.10. Control register 1 (FMC_CTL1)

Address offset: 0x50

Reset value: 0x0000 0080

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CBCMDLEN[2:0]			Reserved			SRAMCMD[1:0]		Reserved							CBCMD
	rw						rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		OBRLD	ENDIE	Reserved	ERRIE	OBWEN	FSTPG	LK	START	OB0ER	OB0PG	MERDF	MER	PER	PG
		rw	rw		rw	rw	rw	rs	rs	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:29	CBCMDLEN[2:0]	CBCMD read length 2^(CBCMDLEN). The read length by check blank command. The read length is 2^ CBCMDLEN double words.
27:26	Reserved	Must be kept at reset value.
25:24	SRAMCMD[1:0]	Shared RAM command. These bits are set by software and cleared by hardware when PRAMRDY or BRAMRDY is set. 00: no operation 01: set fast program RAM mode 10: set Basic RAM mode 11: Reserved. If this value is configured, it may cause unpredictable consequence.
23:17	Reserved	Must be kept at reset value.
16	CBCMD	The command to check the selected page is blank or not.
15:14	Reserved	Must be kept at reset value.
13	OBRLD	Option byte reload bit This bit is set by software. 0: No effect 1: Force option byte reload
12	ENDIE	End of operation interrupt enable bit This bit is set or cleared by software

		0: no interrupt generated by hardware. 1: end of operation interrupt enable
11	Reserved	Must be kept at reset value.
10	ERRIE	Error interrupt enable bit This bit is set or cleared by software 0: no interrupt generated by hardware. 1: error interrupt enable.
9	OBWEN	Option byte erase / program enable bit This bit is set by hardware when right sequence written to the FMC_OBKEY register. This bit can be cleared by software.
8	FSTPG	Main flash fast program command bit. This bit is set or clear by software. 0: no effect 1: main flash fast program command.
7	LK	FMC_CTL1 lock bit This bit is cleared by hardware when right sequence written to the FMC_KEY1 register. This bit can be set by software.
6	START	Send erase command to FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
5	OB0ER	Option bytes 0 erase command bit This bit is set or clear by software 0: no effect 1: option byte erase command
4	OB0PG	Option bytes 0 program command bit This bit is set or clear by software 0: no effect 1: option bytes 0 program command
3	MERDF	Data flash mass erase command bit This bit is set or cleared by software 0: no effect 1: Data flash mass erase command
2	MER	Main flash mass erase command bit This bit is set or cleared by software 0: no effect 1: main flash mass erase command
1	PER	Main flash page erase command bit This bit is set or clear by software 0: no effect

		1: main flash page erase command
0	PG	Main flash program command bit This bit is set or clear by software 0: no effect 1: main flash program command

Note: This register should be reset after the corresponding flash operation completed.

2.4.11. Address register 1 (FMC_ADDR1)

Address offset: 0x54

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:0	ADDR[31:0]	Flash erase / program command address bits These bits are configured by software. ADDR bits are the address of flash to be erased / programmed.

2.4.12. Option byte status register (FMC_OBSTAT)

Address offset: 0x5C

Reset value: 0xXXXX XX0X

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USER[7:0]								Reserved					PLEVEL[1:0]		OBERR
r													r		r

Bits	Fields	Descriptions
31:16	DATA[15:0]	Store OB_DATA[15:0] of option byte block after system reset
15:8	USER[7:0]	Store OB_USER byte of option byte block after system reset
7:3	Reserved	Must be kept at reset value

2:1	PLEVEL[1:0]	Security Protection level 00: No protection level 01: Protect level low 10: Reserved 11: Protect level high
0	OBERR	Option byte read error bit. This bit is set by hardware when the option byte and its complement byte do not match, and the option byte set 0xFF.

2.4.13. Erase / Program protection register 0 (FMC_WP0)

Address offset: 0x60

Reset value: 0xXXXX XXXX

This register has to be accessed by word(32-bit).



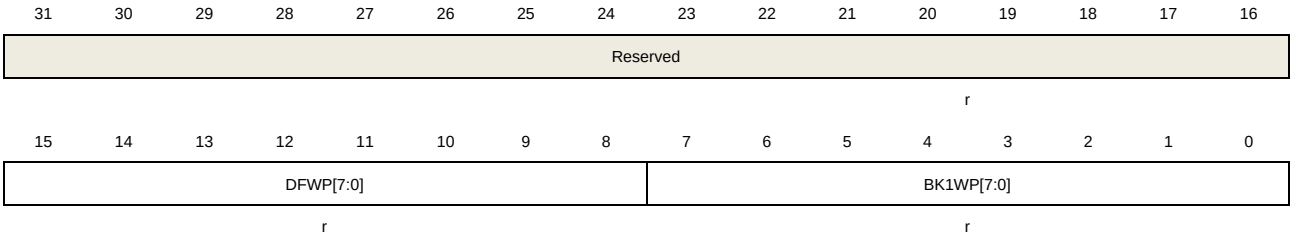
Bits	Fields	Descriptions
31:0	BK0WP[31:0]	Store OB_BK0WP[31:0] of option bytes 0 block after system reset.

2.4.14. Erase / Program protection register 1 (FMC_WP1)

Address offset: 0x64

Reset value: 0xXXXX XXXX

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	DFWP[7:0]	Store OB_DFWP[7:0] of option bytes 0 block after system reset.
7:0	BK1WP[7:0]	Store OB_BK1WP[7:0] of option bytes 0 block after system reset.

2.4.15. Option byte 1 control and status register (FMC_OB1CS)

Address offset: 0x68
Reset value: 0xXXXX XX0X

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LKVAL[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPLOAD	Reserved			EPSIZE[3:0]				EFALC[3:0]				Reserved	OB1LK	OB1START	OB1ERR
rw				rw				rw					r	rw	r

Bits	Fields	Descriptions
31:16	LKVAL[15:0]	Load LKVAL of option byte 1 after reset. These bits can be written by software when OB1LK is 0.
15	EPLOAD	Load EPLOAD of option byte 1 after reset. This bit can be written by software when OB1LK is 0.
14:12	Reserved	Must be kept at reset value.
11:8	EPSIZE[3:0]	Load EPSIZE of option byte 1 after reset. These bits can be written by software when OB1LK is 0.
7:4	EFALC[3:0]	Load EFALC of option byte 1 after reset. These bits can be written by software when OB1LK is 0.
3	Reserved	Must be kept at reset value.
2	OB1LK	When LKVAL is 0x33CC, the OB1LK bit will be set. If OB1LK is 1, the FMC_OB1CS register cannot be configured anymore.
1	OB1START	Send option byte 1 change command to FMC. It is set only by software and cleared when the BUSY bit is cleared.
0	OB1ERR	Option bytes 1 read error bit. This bit is set by hardware when the option bytes 1 and its complement byte do not match, and the option byte 1 set 0xFFFF FFFF.

2.4.16. Product ID register (FMC_PID)

Address offset: 0x100
Reset value: 0xXXXX XXXX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PID[31:16]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID[15:0]															

r

Bits	Field	Descriptions
31:0	PID[31:0]	Product reserved ID code register These bits are read only by software. These bits are unchanged constant after power on. These bits are one time program when the chip produced.

3. Power management unit (PMU)

3.1. Overview

The power consumption is regarded as one of the most important issues for the devices of GD32A50x series. Power management unit (PMU) provides three types of power saving modes, including Sleep, Deep-sleep and Standby mode. These modes reduce the power consumption and allow the application to achieve a best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. For GD32A50x devices, there are two power domains, including V_{DD} / V_{DDA} domain and 1.1V domain, as is shown in the ***Figure 3-1. Power supply overview***. The power of the V_{DD} domain is supplied directly by V_{DD}. An embedded LDO in the V_{DD} / V_{DDA} domain is used to supply the 1.1V domain power.

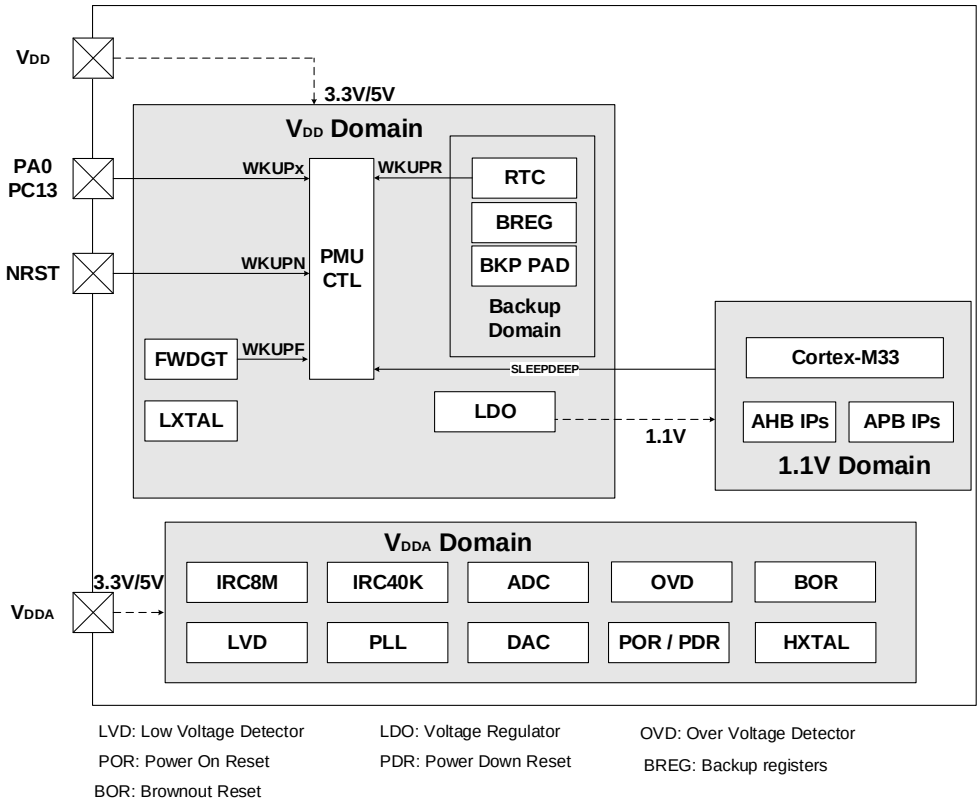
3.2. Characteristics

- Two power domains: V_{DD} / V_{DDA} domain and 1.1V domain.
- Three power saving modes: Sleep, Deep-sleep and Standby modes.
- Internal Voltage regulator (LDO) supplies around 1.1V voltage source for 1.1V domain.
- Low Voltage Detector (LVD) issue an interrupt or event when the power is lower than a programmed threshold.
- Over Voltage Detector (OVD) issue an interrupt or event when the power is higher than a programmed threshold.

3.3. Function overview

Figure 3-1. Power supply overview provides details on the internal configuration of the PMU and the relevant power domains.

Figure 3-1. Power supply overview



3.3.1. V_{DD} / V_{DDA} power domain

V_{DD} / V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes Backup domain, LXTAL (low speed crystal oscillator), LDO (voltage regulator), FWDGT (free watchdog timer), all pads except PC13 / PC14 / PC15, etc. V_{DDA} domain includes ADC / DAC (AD / DA converter), IRC8M (internal 8MHz RC oscillator), IRC40K (internal 40KHz RC oscillator), PLL (phase locking loop), HXTAL (high speed crystal oscillator), POR / PDR (power on / down reset), LVD (low voltage detector), OVD (over voltage detector), etc.

V_{DD} domain

The LDO, which is implemented to supply power for the 1.1V domain, is always enabled after the reset. It can be configured to operate in three different statuses, including the Sleep mode (full power on), the Deep-sleep mode (full power on or low power), and the Standby mode (power off).

Backup domain

The Backup domain is located in V_{DD} domain, the V_{DD} supplies power for RTC unit, LXTAL oscillator, BREG and three BKP PAD including PC13 to PC15.

The application software can trigger the Backup domain software reset by setting the BKPRST bit in the RCU_BDCTL register.

The clock source of the Real Time Clock (RTC) circuit can be derived from the Internal 40KHz

RC oscillator (IRC40K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL clock divided by 128. Before entering the power saving mode by executing the WFI / WFE instruction, the Cortex®-M33 can setup the RTC register with an expected wakeup time and enable the wakeup function to achieve the RTC timer wakeup event. After entering the power saving mode for a certain amount of time, the RTC will wake up the device when the time match event occurs. The details of the RTC configuration and operation will be described in the ***Real-time clock (RTC)***.

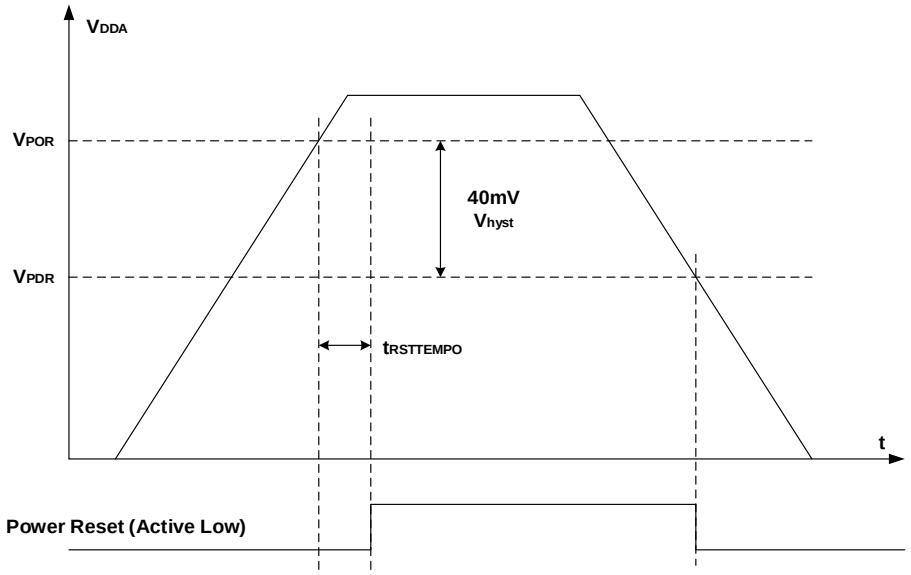
When the Backup domain is supplied, the following functions are available:

- PC13 can be used as GPIO or RTC function pin described in the ***RTC clock calibration***.
- PC14 and PC15 can be used as either GPIO or LXTAL crystal oscillator pins.

V_{DDA} domain

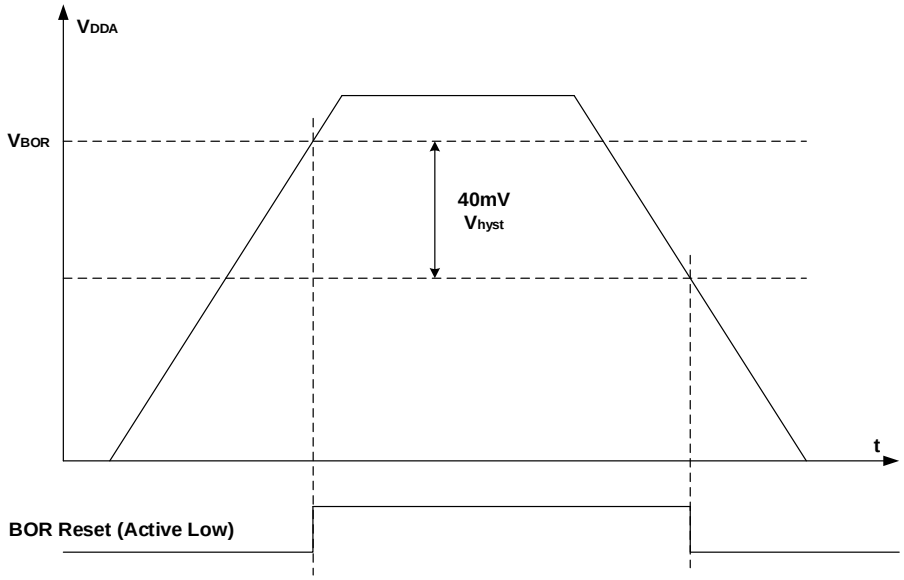
The POR / PDR circuit is implemented to detect V_{DDA} and generate the power reset signal which resets the whole chip when the supply voltage is lower than the specified threshold. ***Figure 3-2. Waveform of the POR / PDR*** shows the relationship between the supply voltage and the power reset signal. V_{POR} indicates the threshold of power on reset, while V_{PDR} means the threshold of power down reset. The hysteresis voltage (V_{hyst}) is around 40mV.

Figure 3-2. Waveform of the POR / PDR



The BOR circuit is used to detect V_{DDA} and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold which defined in the BOR_TH bits in option bytes. ***Figure 3-3. Waveform of the BOR*** shows the relationship between the supply voltage and the BOR reset signal. V_{BOR}, which defined in the BOR_TH bits in option bytes, indicates the threshold of BOR on reset. The hysteresis voltage (V_{hyst}) is 40mV.

Figure 3-3. Waveform of the BOR



The LVD is used to detect whether the V_{DDA} supply voltage is lower than a programmed threshold selected by the LVDT[2:0] bits in the Power control register (PMU_CTL). The LVD is enabled by setting the LVDEN bit in the PMU_CTL register. And LVDF bit, which is in the Power control and status register (PMU_CS), indicates if V_{DDA} is higher or lower than the LVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. ***Figure 3-4. Waveform of the LVD threshold*** shows the relationship between the LVD threshold and the LVD output (LVD interrupt signal depends on EXTI line 16 rising or falling edge configuration). The following figure also shows the relationship between the supply voltage and the LVD signal. The hysteresis voltage (V_{hyst}) is 100mV.

The OVD is used to detect whether the V_{DDA} supply voltage is over than a programmed threshold selected by the OVDT bits in the Power control register (PMU_CTL). The OVD is enabled by setting the OVDEN bit in the PMU_CTL register. And OVDF bit indicates if V_{DDA} is higher or lower than the OVD threshold. This event is internally connected to the EXTI line 24 and can generate an interrupt if it is enabled through the EXTI registers. ***Figure 3-5. Waveform of the OVD threshold*** shows the relationship between the OVD threshold and the OVD output (OVD interrupt signal depends on EXTI line 24 rising or falling edge configuration). The following figure also shows the relationship between the supply voltage and the OVD signal. The hysteresis voltage (V_{hyst}) is 25mV.

Figure 3-4. Waveform of the LVD threshold

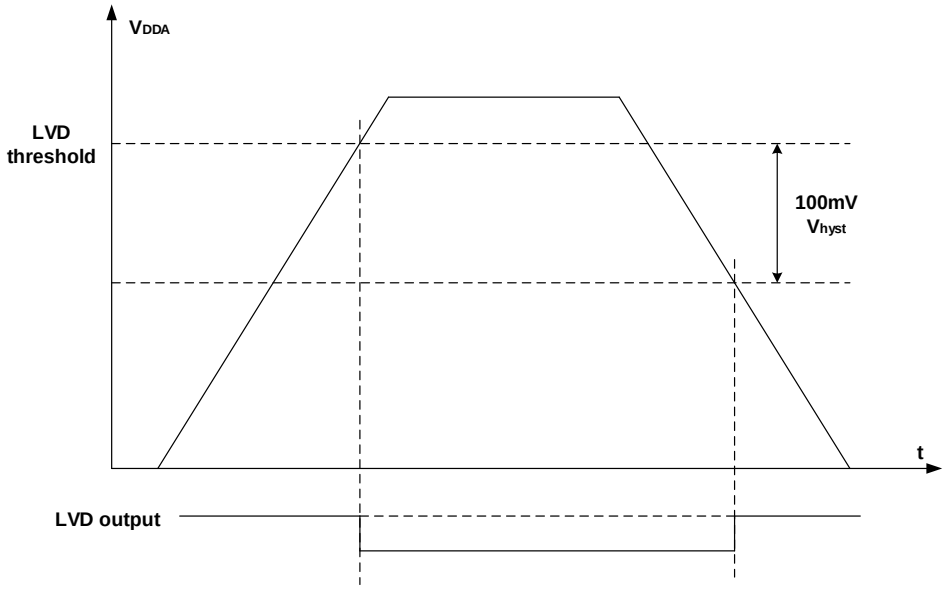
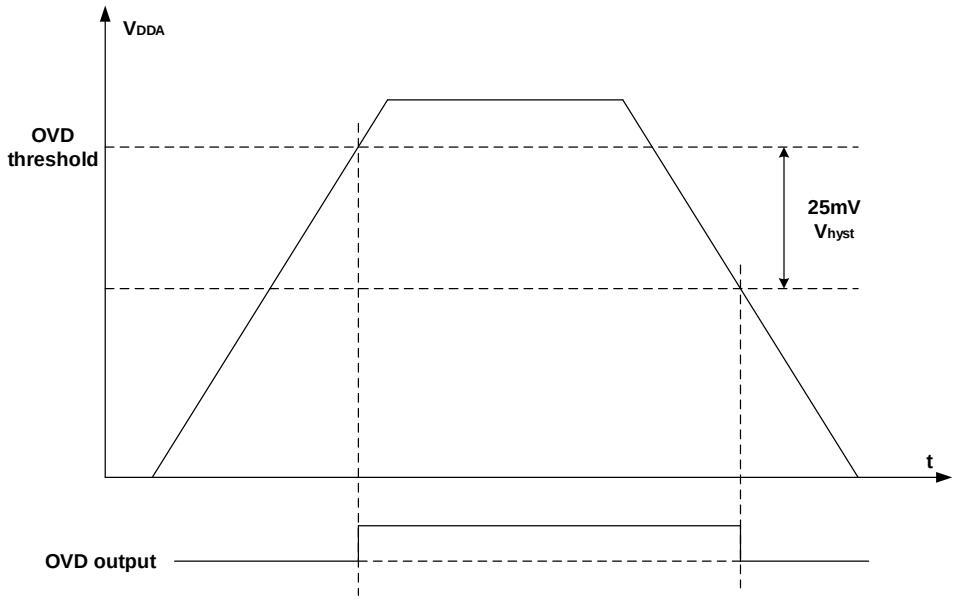


Figure 3-5. Waveform of the OVD threshold



Generally, digital circuits are powered by V_{DD}, while most of analog circuits are powered by V_{DDA}. To improve the conversion accuracy of ADC and DAC, the independent power supply V_{DDA} is implemented to achieve better performance of analog circuits. V_{DDA} can be externally connected to V_{DD} through the external filtering circuit to avoid noise on V_{DDA}, and V_{SSA} should be connected to V_{SS} through the specific circuit independently. Otherwise, if V_{DDA} is different from V_{DD}, V_{DDA} must always be higher, and the voltage difference should not exceed 0.3V.

To ensure a high accuracy on ADC and DAC, the ADC / DAC independent external reference voltage should be connected to V_{REFP} / V_{REFN} pins. According to the different packages, V_{REFP} pin can be connected to V_{DDA} pin, or external reference voltage which refers to ***Table 14-2. ADC input pins definition*** and ***Table 15-1. DAC I/O description***, V_{REFN} pin must be connected to V_{SSA} pin. The V_{REFP} pin is only available on 100-pin and 64-pin packages. For

other packages, the V_{REFP} pin is not available and it is internally connected to V_{DDA}. The V_{REFN} pin is internally connected to V_{SSA}.

3.3.2. 1.1V power domain

1.1V power domain supplies power for Cortex®-M33 logic, AHB / APB peripherals, the APB interfaces for the V_{DD} / V_{DDA} domain, etc. Once the 1.1V is powered up, the POR will generate a reset sequence on the 1.1V power domain. If it is needed to enter the specified power saving mode, the associated control bits must be configured. Then, once a WFI (Wait for Interrupt) or WFE (Wait for Event) instruction is executed, the device will enter the specified power saving mode which will be discussed in the following section.

3.3.3. Power saving modes

After a system reset or a power reset, the GD32A50x MCU operates at full function state and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1 and PCLK2), closing the clocks of the unused peripherals. Besides, three power saving modes are provided to achieve even lower power consumption. They are Sleep mode, Deep-sleep mode and Standby mode.

Sleep mode

The Sleep mode is corresponding to the SLEEPING mode of the Cortex®-M33. In Sleep mode, only clock of Cortex®-M33 is off. To enter the Sleep mode, it is only necessary to clear the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and execute a WFI or WFE instruction. If the Sleep mode is entered by executing a WFI instruction, any interrupt can wake up the system. If it is entered by executing a WFE instruction, any wakeup event can wake up the system (If SEVONPEND is 1, any interrupt can wake up the system, refer to Cortex-M33 Technical Reference Manual). The mode costs the lowest wakeup time as no time is wasted in interrupt entry or exit.

According to the SLEEPONEXIT bit in the Cortex®-M33 System Control Register, there are two options to select the entry mechanism of Sleep mode.

- Sleep-now: if the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode as soon as a WFI or WFE instruction is executed.
- Sleep-on-exit: if the SLEEPONEXIT bit is set, the MCU enters Sleep mode as soon as it exits from the ISR with the lowest priority.

Deep-sleep mode

The Deep-sleep mode is based on the SLEEPDEEP mode of the Cortex®-M33. In Deep-sleep mode, SRAM0 (0KB to 16KB) data is retained, all clocks in the 1.1V domain are off, and all of IRC8M, HXTAL and PLL are disabled. The registers' contents are preserved. According to the configuration of SRAMSW1 and SRAMSW2 bits in the PMU_CTL register, the data retention ability of SRAM1 (16KB ~ 32KB) and SRAM2 (32KB ~ 48KB) in deep sleep mode

can be set respectively. The LDO can operate in normal mode or in low power mode depending on the LDOLP bit in the PMU_CTL register. Before entering the Deep-sleep mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and clear the STBMOD bit in the PMU_CTL register. Then, the device enters the Deep-sleep mode after a WFI or WFE instruction is executed. If the Deep-sleep mode is entered by executing a WFI instruction, any interrupt from EXTI lines can wake up the system. If it is entered by executing a WFE instruction, any wakeup event from EXTI lines can wake up the system (If SEVONPEND is 1, any interrupt from EXTI lines can wake up the system, refer to Cortex®-M33 Technical Reference Manual). When exiting the Deep-sleep mode, the IRC8M is selected as the system clock. Notice that an additional wakeup delay will be incurred if the LDO operates in low power mode.

The low-driver mode in Deep-sleep mode can be entered by configuring the LDEN bit in the PMU_CTL0 register. The low-driver mode provides lower drive capability, and the low-power mode take lower power.

Normal-driver / Normal-power: The Deep-sleep mode is not in low-driver mode by configure LDEN to 0 in the PMU_CTL0 register, and not in low-power mode depending on the LDOLP bit reset in the PMU_CTL0 register.

Normal-driver / Low-power: The Deep-sleep mode is not in low-driver mode by configure LDEN to 0 in the PMU_CTL0 register. The low-power mode enters depending on the LDOLP bit set in the PMU_CTL0 register.

Low-driver / Normal-power: The low-driver mode in Deep-sleep mode enters by configure LDEN to 1 in the PMU_CTL0 register. And not in low-power mode depending on the LDOLP bit reset in the PMU_CTL0 register.

Low-driver / Low-power: The low-driver mode in Deep-sleep mode enters by configure LDEN to 1 in the PMU_CTL0 register. The low-power mode enters depending on the LDOLP bit set.

Note: In order to enter Deep-sleep mode smoothly, all EXTI line pending status (in the EXTI_PD register) and related peripheral flags must be reset, refer to ***Table 6-3. EXTI source***. If not, the program will skip the entry process of Deep-sleep mode to continue to execute the following procedure. Before enter Deep-sleep mode, need to set SLEEP_SLP bit in FMC_WS register.

Standby mode

The Standby mode is also based on the SLEEPDEEP mode of the Cortex®-M33. In Standby mode, the whole 1.1V domain is powered off, the LDO is shut down, and all of IRC8M, HXTAL and PLL are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, set the STBMOD bit in the PMU_CTL register, and clear WUF bit in the PMU_CS register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS register indicates whether the MCU has been in Standby mode. There are four wakeup sources for the Standby mode, including the external reset from NRST pin, the RTC alarm, the FWDGT reset and the rising edge on WKUP pin. The Standby mode achieves the

lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in 1.1V power domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset occurs and the Cortex®-M33 will execute instruction code from the address of 0x0000 0000.

Table 3-1. Power saving mode summary

Mode	Sleep	Deep-sleep	Standby
Description	Only CPU clock is off	1. All clocks in the 1.1V domain are off 2. Disable IRC8M, HXTAL and PLL	1. The 1.1V domain is powered off 2. Disable IRC8M, HXTAL and PLL
LDO Status	On (normal power mode, normal driver mode)	On (normal power mode or low power mode, normal driver mode or low driver mode)	Off
Configuration	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST = 1
Entry	WFI or WFE	WFI or WFE	WFI or WFE
Wakeup	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	1. NRST pin 2. WKUP pin 3. FWDGT reset 4. RTC alarm
Wakeup Latency	None	IRC8M wakeup time, LDO wakeup time added if LDO is in low power mode	Power on sequence

Note: In Standby mode, all I / Os are in high-impedance state except NRST pin, PC13 pin when configured for RTC function, PC14 and PC15 pins when used as LXTAL crystal oscillator pins, and WKUP pin if enabled.

3.4. Register definition

PMU base address: 0x4000 7000

3.4.1. Control register (PMU_CTL)

Address offset: 0x00

Reset value: 0x0000 8000 (reset after wakeup from Standby mode)

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										SRAMSW2	SRAMSW1	Reserved	LDEN	Reserved	
										rw	rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVDT	OVDEN	Reserved					BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
rw	rw						rw	rw			rw	rc_w1	rc_w1	rw	rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	SRAMSW2	SRAM2(32KB~48KB) power switch in deep-sleep mode 0: SRAM2 power on and data retention in deep-sleep mode 1: SRAM2 power off and data lost in deep-sleep mode
20	SRAMSW1	SRAM1(16KB~32KB) power switch in deep-sleep mode 0: SRAM1 power on and data retention in deep-sleep mode 1: SRAM1 power off and data lost in deep-sleep mode
19	Reserved	Must be kept at reset value.
18	LDEN	Low-driver mode enable in Deep-sleep mode 0: Low-driver mode disable in Deep-sleep mode 1: Low-driver mode enable in Deep-sleep mode
17:16	Reserved	Must be kept at reset value.
15	OVDT	Over Voltage Detector Threshold 0: 5V 1: 5.5V
14	OVDEN	Over Voltage Detector Enable 0: Disable Over Voltage Detector 1: Enable Over Voltage Detector
13:9	Reserved	Must be kept at reset value.
8	BKPWEN	Backup Domain Write Enable 0: Disable write access to the registers in Backup domain

		1: Enable write access to the registers in Backup domain After reset, any write access to the registers in Backup domain is ignored. This bit has to be set to enable write access to these registers.
7:5	LVDT[2:0]	Low Voltage Detector Threshold 000: 2.9V 001: 3.1V 010: 3.3V 011: 3.5V 100: 4.0V 101: 4.2V 110: 4.4V 111: 4.6V
4	LVDEN	Low Voltage Detector Enable 0: Disable Low Voltage Detector 1: Enable Low Voltage Detector **Note:** When LVD_LOCK bit is set to 1 in the SYSCFG_CFG2 register, LVDEN and LVDT[2:0] are read only.
3	STBRST	Standby Flag Reset 0: No effect 1: Reset the standby flag This bit is always read as 0.
2	WURST	Wakeup Flag Reset 0: No effect 1: Reset the wakeup flag This bit is always read as 0.
1	STBMOD	Standby Mode 0: Enter the Deep-sleep mode when the Cortex®-M33 enters SLEEPDEEP mode 1: Enter the Standby mode when the Cortex®-M33 enters SLEEPDEEP mode
0	LDOLP	LDO Low Power Mode 0: The LDO operates normally during the Deep-sleep mode 1: The LDO is in low power mode during the Deep-sleep mode **Note:** Some peripherals may work with the IRC8M clock in the Deep-sleep mode. In this case, the LDO automatically switches from the low power mode to the normal mode and remains in this mode until the peripheral stop working.

3.4.2. Control and status register (PMU_CS)

Address offset: 0x04

Reset value: 0x0000 0000 (will not reset after wakeup from Standby mode)

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						WUPEN1	WUPEN0	Reserved				OVDF	LVDF	STBF	WUF
						rw	rw					r	r	r	r

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	WUPEN1	WKUP Pin1 Enable (PC13) 0: Disable WKUP pin1 function 1: Enable WKUP pin1 function If WUPEN1 is set before entering the Standby mode, a rising edge on the WKUP pin1 will wake up the system from the Standby mode. As the WKUP pin1 is active high, the WKUP pin1 is internally configured to input pull down mode. And setting this bit will trigger a wakeup event when the input is already high.
8	WUPEN0	WKUP Pin0 Enable (PA0) 0: Disable WKUP pin0 function 1: Enable WKUP pin0 function If WUPEN0 is set before entering the Standby mode, a rising edge on the WKUP pin0 will wake up the system from the Standby mode. As the WKUP pin0 is active high, the WKUP pin0 is internally configured to input pull down mode. And setting this bit will trigger a wakeup event when the input is already high.
7:4	Reserved	Must be kept at reset value.
3	OVDF	Over Voltage Detector Status Flag 0: Over Voltage event has not occurred (V_{DDA} is lower than the specified OVD threshold) 1: Over Voltage event occurred (V_{DDA} is equal to or higher than the specified OVD threshold) **Note:** The OVD function is disabled in Standby mode.
2	LVDF	Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (V_{DDA} is higher than the specified LVD threshold) 1: Low Voltage event occurred (V_{DDA} is equal to or lower than the specified LVD threshold) **Note:** The LVD function is disabled in Standby mode.
1	STBF	Standby Flag 0: The device has not entered the Standby mode 1: The device has been in the Standby mode This bit is cleared only by a POR/PDR or by setting the STBRST bit in the PMU_CTL register.
0	WUF	Wakeup Flag

0: No wakeup event has been received

1: Wakeup event occurred from the WKUP pin or the RTC alarm event

This bit is reset by the system or cleared by setting the WURST bit in the PMU_CTL register.

4. Backup registers (BKP)

4.1. Overview

The Backup registers are located in the Backup domain that remains powered-on by V_{DD} power, there are ten 16-bit (20 bytes) registers for data protection of user application data, and the wake-up action from standby mode or system reset do not affect these registers.

In addition, the BKP registers can be used to implement the tamper detection and RTC calibration function.

After reset, any writing access to the registers in Backup domain is disabled, that is, the Backup registers and RTC cannot be written to access. In order to enable access to the Backup registers and RTC, the Power and Backup interface clocks should be enabled firstly by setting the PMUEN and BKPEN bits in the RCU_APB1EN register, and writing access to the registers in Backup domain should be enabled by setting the BKPWEN bit in the PMU_CTL register.

4.2. Characteristics

- 20 bytes Backup registers which can keep data under power saving mode. If tamper event is detected, Backup registers will be reset.
- The active level of Tamper source (PC13) can be configured.
- RTC Clock Calibration register provides RTC alarm and second output selection, and sets the calibration value.
- Tamper control and status register (BKP_TPCS) can control tamper detection with interrupt or event capability.

4.3. Function overview

4.3.1. RTC clock calibration

In order to improve the RTC clock accuracy, the MCU provides the RTC output for calibration function. The RTC clock, or a clock with the frequency is f_{RTCCLK}/64, can be output on the PC13. It is enabled by setting the COEN bit in the BKP_OCTL register.

The calibration value is set by RCCV[6:0] in the BKP_OCTL register, and the calibration function can slow down or speed up the RTC clock by steps of 1000000/2^20 ppm.

4.3.2. Tamper detection

In order to protect the important user data, the MCU provides the tamper detection function, and it can be independently enabled on TAMPER pin by setting corresponding TPEN bit in

the BKP_TPCTL register. To prevent the tamper event from losing, the edge detection is logically ANDed with the TPEN bit, used for tamper detection signal. So the tamper detection configuration should be set before enable TAMPER pin. When the tamper event is detected, the corresponding TEF bit in the BKP_TPCS register will be set. Tamper event can generate an interrupt if tamper interrupt is enabled. Any tamper event will reset all Backup data registers.

Note: When TPAL=0/1, if the TAMPER pin is already high / low before it is enabled (by setting TPEN bit), an extra tamper event is detected, while there was no rising / falling edge on the TAMPER pin after TPEN bit was set.

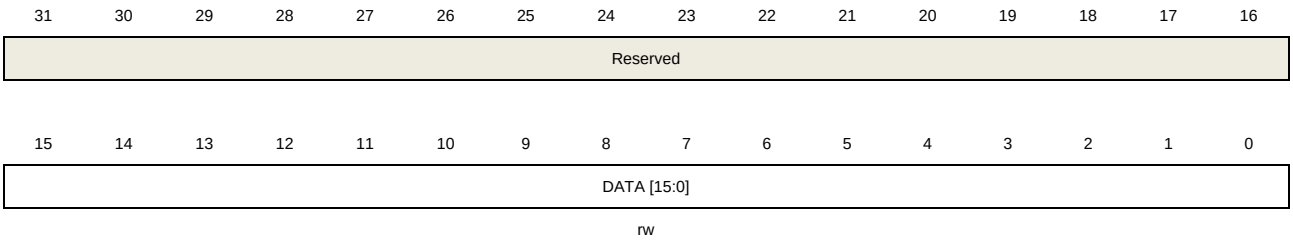
4.4. Register definition

BKP base address: 0x4000 6C00

4.4.1. Backup data register x (BKP_DATAx) (x= 0..9)

Address offset: 0x04 to 0x28
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

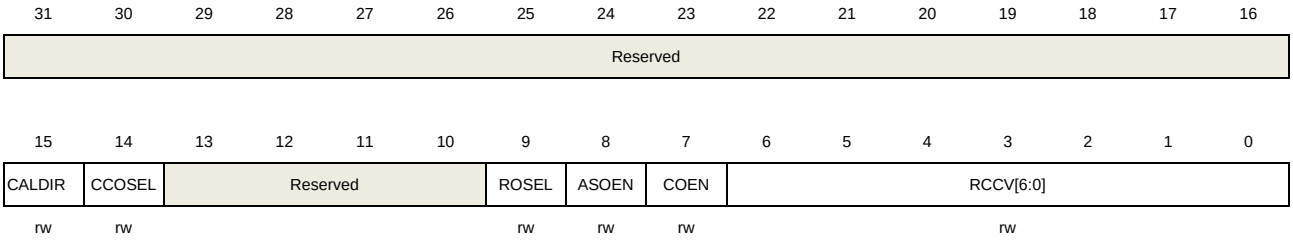


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DATA[15:0]	Backup data These bits are used for general purpose data storage. The contents of the BKP_DATAx register will remain even if wake up action from Standby mode or system reset.

4.4.2. RTC signal output control register (BKP_OCTL)

Address offset: 0x2C
Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CALDIR	RTC clock calibration direction 0: Slowed down 1: Speed up This bit is reset only by a Backup domain reset.

14	CCOSEL	RTC clock output selection 0: RTC clock div 64 1: RTC clock This bit is reset only by a POR.
13:10	Reserved	Must be kept at reset value.
9	ROSEL	RTC output selection 0: RTC alarm pulse is selected as the RTC output 1: RTC second pulse is selected as the RTC output This bit is reset only by a Backup domain reset.
8	ASOEN	RTC alarm or second signal output enable 0: Disable RTC alarm or second output 1: Enable RTC alarm or second output When enable, the TAMPER pin will output the RTC output. This bit is reset only by a Backup domain reset.
7	COEN	RTC clock calibration output enable 0: Disable RTC clock calibration output 1: Enable RTC clock Calibration output When enable, the TAMPER pin will output the RTC clock or RTC clock divided by 64. ASOEN has the priority over COEN. When ASOEN is set, the TAMPER pin will output the RTC alarm or second signal whether COEN is set or not. This bit is reset only by a POR.
6:0	RCCV[6:0]	RTC clock calibration value The value indicates how many clock pulses are ignored or added every 2^20 RTC clock pulses. This bit is reset only by a Backup domain reset.

4.4.3. Tamper pin control register (BKP_TPCTL)

Address offset: 0x30

Reset value: 0x0000 8000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCSEL	Reserved													TPAL	TPEN
rw														rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15	PCSEL	OSC32_IN pin select 0: PC13 is OSC32_IN pin 1: PC14 is OSC32_IN pin
14:2	Reserved	Must be kept at reset value.
1	TPAL	TAMPER pin active level 0: The TAMPER pin is active high 1: The TAMPER pin is active low
0	TPEN	TAMPER detection enable 0: The TAMPER pin is free for GPIO functions 1: The TAMPER pin is dedicated for the Backup Reset function. The active level on the TAMPER pin resets all data of the BKP_DATAx register.

4.4.4. Tamper control and status register (BKP_TPCS)

Address offset: 0x34

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TIF	TEF	Reserved					TPIE	TIR	TER
						r	r						rw	w	w

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	TIF	Tamper interrupt flag 0: No tamper interrupt occurred 1: A tamper interrupt occurred This bit is reset by writing 1 to the TIR bit or the TPIE bit being 0.
8	TEF	Tamper event flag 0: No tamper event occurred 1: A tamper event occurred This bit is reset by writing 1 to the TER bit.
7:3	Reserved	Must be kept at reset value
2	TPIE	Tamper interrupt enable 0: Disable the tamper interrupt 1: Enable the tamper interrupt This bit is reset only by a system reset and wake-up from Standby mode.

1	TIR	Tamper interrupt reset 0: No effect 1: Reset the TIF bit This bit is always read as 0.
0	TER	Tamper event reset 0: No effect 1: Reset the TEF bit This bit is always read as 0.

5. Reset and clock unit (RCU)

5.1. Reset control unit (RCTL)

5.1.1. Overview

GD32A50x reset control includes the control of three kinds of reset: power reset, system reset and backup domain reset. The power on reset, known as a cold reset, resets the full system except the backup domain during a power up. A system reset resets the processor core and peripheral IP components with the exception of the SW-DP controller and the backup domain. A backup domain reset resets the backup domain. The resets can be triggered by an external signal, internal events and the reset generators. More information about these resets will be described in the following sections.

5.1.2. Function overview

Power Reset

The power reset is generated by either an external reset as power on and power down reset (POR/PDR reset), or by the internal reset generator when exiting standby mode. The power reset sets all registers to their reset values except the backup domain. The power reset which active signal is low will be de-asserted when the internal LDO voltage regulator is ready to provide 1.1V power for GD32A50x series. The reset service routine vector is fixed at address 0x0000_0004 in the memory map.

System Reset

A system reset is generated by the following events:

- A power reset (POWER_RSTn)
- A external pin reset (NRST)
- A window watchdog timer reset (WWDGT_RSTn)
- A free watchdog timer reset (FWDGT_RSTn)
- The SYSRESETREQ bit in Cortex®-M33 application interrupt and reset control register is set (SW_RSTn)
- Option byte loader reset (OBL_RSTn)
- Reset generated when entering Standby mode when resetting nRST_STDBY bit in user option bytes (OB_STDBY_RSTn)
- Reset generated when entering Deep-sleep mode when resetting nRST_DPSLP bit in user option bytes (OB_DPSLP_RSTn)
- Low voltage detect reset (LVD_RSTn)
- Loss-of-HXTAL reset (LOH_RSTn)
- Loss-of-PLL reset (LOP_RSTn)
- CPU lockup reset (LOCKUP_RSTn)

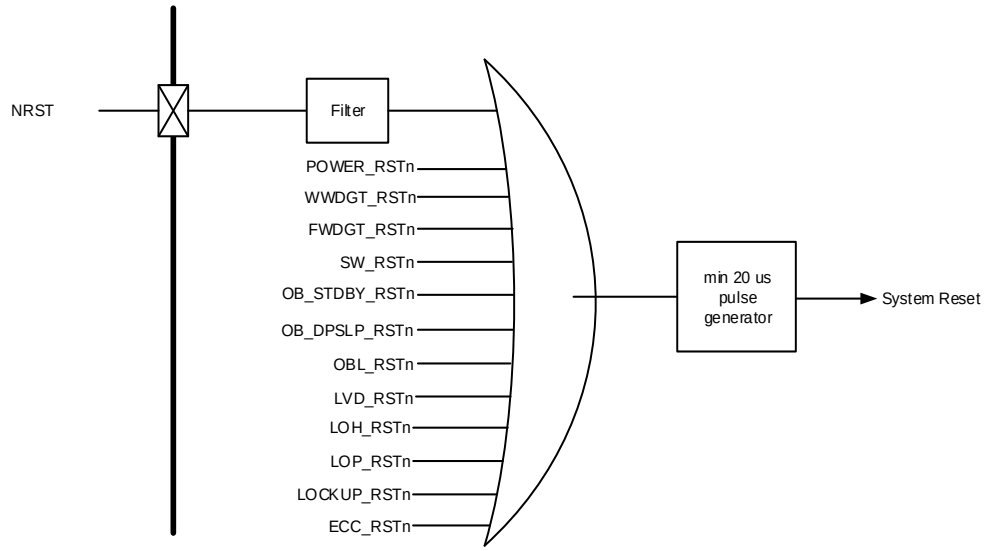
- FLASH or SRAM 2-bit ECC error reset (ECC_RSTn)

LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn / ECC_RSTn should be enable by software.

A system reset resets the processor core and peripheral IP components except for the SW-DP controller and the backup domain.

A system reset pulse generator guarantees low level pulse duration of 20 μs for each reset source (external or internal reset).

Figure 5-1. The system reset circuit



Note: It is conditional for LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn / ECC_RSTn to generate a reset. LVDRSTEN / LOHRSTEN / LOPRSTEN / LOCKUPRSTEN / ECCRSTEN bit in Reset source /clock register (RCU_RSTSCK) must be set.

Backup domain reset

A backup domain reset is generated by setting the BKPRST bit in the backup domain control register or backup domain power on reset (V_{DD} power on).

5.2. Clock control unit (CCTL)

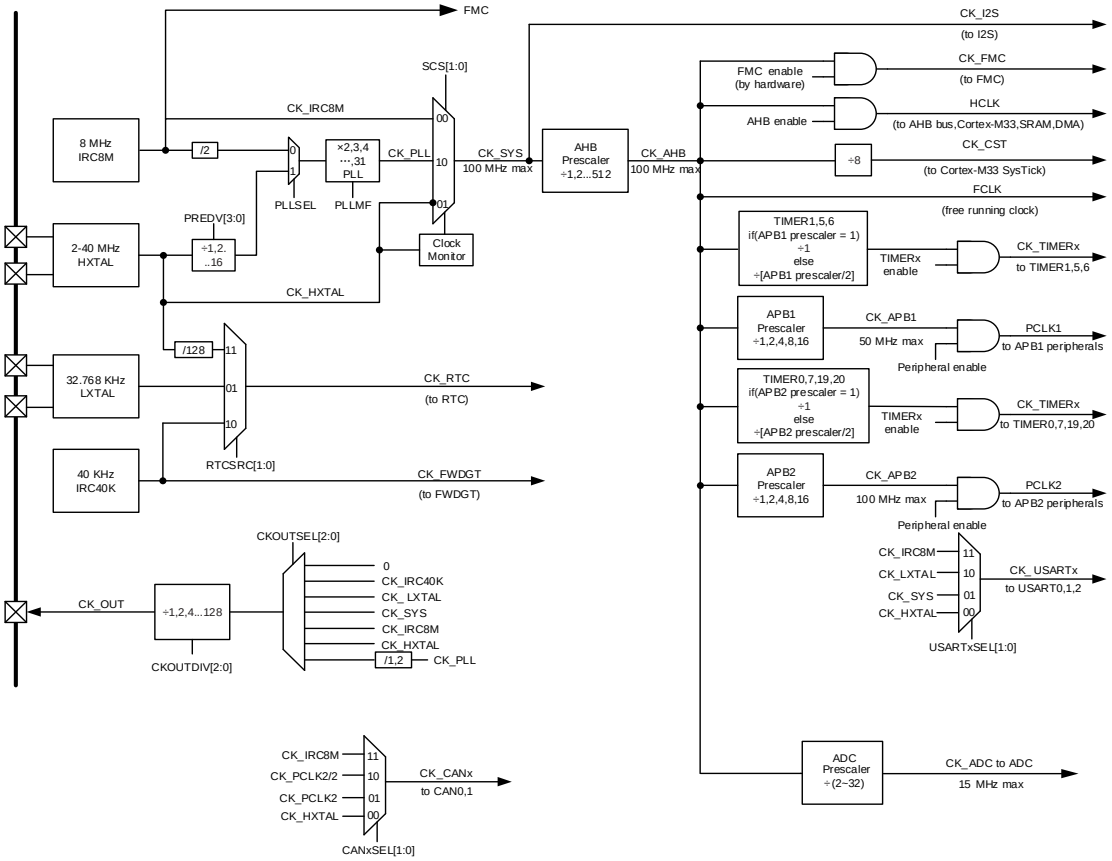
5.2.1. Overview

The clock control unit provides a range of frequencies and clock functions. These include an Internal 8 MHz RC oscillator (IRC8M), a High speed crystal oscillator (HXTAL), an Internal 40KHz RC oscillator (IRC40K), a Low speed crystal oscillator (LXTAL), a Phase Lock Loop (PLL), a HXTAL clock monitor, PLL Clock Monitor (PLLM), LXTAL Clock Monitor (LCKM), clock prescalers, clock multiplexers and clock gating circuitry.

The clocks of the AHB, APB and Cortex®-M33 are derived from the system clock (CK_SYS) which can source from the IRC8M, HXTAL or PLL. The maximum operating frequency of the

system clock (CK_SYS) can be up to 100 MHz.

Figure 5-2. Clock tree



The frequency of AHB, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB, APB2 and APB1 domains is 100 MHz/100 MHz/50 MHz. The Cortex System Timer (SysTick) external clock is clocked with the AHB clock (HCLK) divided by 8. The SysTick can work either with this clock or with the AHB clock (HCLK), configurable in the SysTick control and status register.

The ADC are clocked by the clock of AHB divided by 2~32 selected by ADCPSC bits in configuration register 2 (RCU_CFG2). The USART0/1/2 is clocked by IRC8M clock or LXTAL clock or system clock or HXTAL clock, which selected by USART0/1/2SEL bits in configuration register 2 (RCU_CFG2). The CAN0/1 is clocked by IRC8M clock or PCLK2/2 clock or PCLK2 clock or HXTAL clock, which selected by CAN0/1SEL bits in configuration register 2 (RCU_CFG2).

The RTC is clocked by LXTAL clock or IRC40K clock or HXTAL clock divided by 128 which select by RTCSRC bits in backup domain control register (RCU_BDCTL).

The FWDGT is clocked by IRC40K clock, which is forced on when FWDGT started.

If the APB prescaler is 1, the timer clock frequencies are set to AHB frequency divide by 1. Otherwise, they are set to the AHB frequency divide by half of APB prescaler.

5.2.2. Characteristics

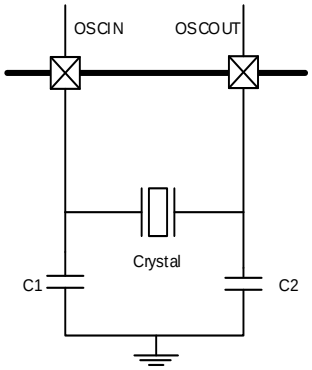
- 2 to 40 MHz High speed crystal oscillator (HXTAL)
- Internal 8 MHz RC oscillator (IRC8M)
- 32.768 KHz Low speed crystal oscillator (LXTAL)
- Internal 40 KHz RC oscillator (IRC40K)
- PLL clock source can be HXTAL or IRC8M
- HXTAL clock monitor

5.2.3. Function overview

High Speed Crystal Oscillator (HXTAL)

The high speed crystal oscillator (HXTAL), which has a frequency from 2 to 40 MHz, produces a highly accurate clock source for use as the system clock. A crystal with a specific frequency must be connected and located close to the two HXTAL pins. The external resistor and capacitor components connected to the crystal are necessary for proper oscillation.

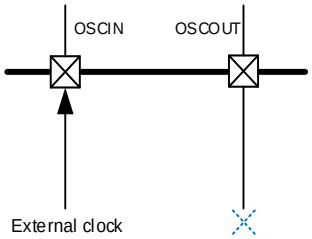
Figure 5-3. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register, RCU_CTL. The HXTALSTB flag in control register, RCU_CTL indicates if the high-speed external crystal oscillator is stable. When the HXTAL is powered up, it will not be released for use until this HXTALSTB bit is set by the hardware. This specific delay period is known as the oscillator “Start-up time”. As the HXTAL becomes stable, an interrupt will be generated if the related interrupt enable bit HXTALSTBIE in the interrupt register RCU_INT is set. At this point the HXTAL clock can be used directly as the system clock source or the PLL input clock.

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the Control Register RCU_CTL. During bypass mode, the signal is connected to OSCIN, and OSCOUT remains in the suspended state, as shown in ***Figure 5-4. HXTAL clock source in bypass mode***.The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 5-4. HXTAL clock source in bypass mode



Select the HXTAL frequency scale by using the HXTALSCAL bit in the control register, RCU_CTL. If HXTAL frequency is higher than 8MHz, HXTALSCAL bit must be set.

Internal 8 MHz RC Oscillator (IRC8M)

The Internal 8 MHz RC oscillator, IRC8M, has a fixed frequency of 8 MHz and is the default clock source selection for the CPU when the device is powered up. The IRC8M oscillator provides a lower cost type clock source as no external components are required. The IRC8M RC oscillator can be switched on or off using the IRC8MEN bit in the control register, RCU_CTL. The IRC8MSTB flag in the control register, RCU_CTL is used to indicate if the internal RC oscillator is stable. The start-up time of the IRC8M oscillator is shorter than the HXTAL crystal oscillator. An interrupt can be generated if the related interrupt enable bit, IRC8MSTBIE, in the interrupt register, RCU_INT, is set when the IRC8M becomes stable. The IRC8M clock can also be used as the PLL input clock.

The frequency accuracy of the IRC8M can be calibrated by the manufacturer, but its operating frequency is still less accurate than HXTAL. The application requirements, environment and cost will determine which oscillator type is selected.

If the HXTAL or PLL is the system clock source, to minimize the time required for the system to recover from the Deep-sleep Mode, the hardware forces the IRC8M clock to be the system clock when the system initially wakes-up.

Phase Locked Loop (PLL)

The internal Phase Locked Loop, PLL, can provide 16~100 MHz clock output which is 2 ~31 multiples of a fundamental reference frequency of 2 ~ 40 MHz.

The PLL can be switched on or off by using the PLLEN bit in the control register, RCU_CTL. The PLLSTB flag in the control register, RCU_CTL will indicate if the PLL clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLSTBIE, in the interrupt register, RCU_INT, is set as the PLL becomes stable.

Low Speed Crystal Oscillator (LXTAL)

The low speed crystal or ceramic resonator oscillator, which has a frequency of 32,768 Hz, produces a low power but highly accurate clock source for the real time clock circuit. The LXTAL oscillator can be switched on or off using the LXTALEN bit in the backup domain control register (RCU_BDCTL). The LXTALSTB flag in the backup domain control register (RCU_BDCTL) will indicate if the LXTAL clock is stable. An interrupt can be generated if the

related interrupt enable bit, LXTALSTBIE, in the interrupt register RCU_INT is set when the LXTAL becomes stable.

Select external clock bypass mode by setting the LXTALBPS and LXTALEN bits in the backup domain control register (RCU_BDCTL). The CK_LXTAL is equal to the external clock which drives the OSC32IN pin.

Note: In this series, only bypass mode (LXTALBPS is set to 1) can be used, and the 32.768KHz clock is provided externally.

Internal 40 KHz RC Oscillator (IRC40K)

The Internal 40 KHz RC Oscillator has a frequency of about 40 kHz and is a low power clock source for the real time clock circuit or the free watchdog timer. The IRC40K offers a low cost clock source as no external components are required. The IRC40K RC oscillator can be switched on or off by using the IRC40KEN bit in the reset source/clock register, RCU_RSTSCK. The IRC40KSTB flag in the reset source/clock register RCU_RSTSCK will indicate if the IRC40K clock is stable. An interrupt can be generated if the related interrupt enable bit IRC40KSTBIE in the Interrupt register RCU_INT is set when the IRC40K becomes stable.

System Clock (CK_SYS) Selection

After the system reset, the default CK_SYS source will be IRC8M and can be switched to HXTAL or PLL by changing the system clock switch bits, SCS, in the configuration register 0, RCU_CFG0. When the SCS value is changed, the CK_SYS will continue to operate using the original clock source until the target clock source is stable. When a clock source is used directly by the CK_SYS or the PLL, it is not possible to stop it.

PLL Clock Monitor (PLLM)

The PLL clock monitor function is enabled by the PLL clock monitor enable bit, PLLMEN, in the control register, RCU_CTL. This function should be enabled after the PLL start-up delay and disabled when the PLL is stopped or slow down. Once the PLL failure is detected, Loss-of-PLL reset or an interrupt will generate. Reset function is decided by LOPRSTEN in the Reset source /clock register, RCU_RSTSCK. Interrupt function is decided by PLLMIE in the interrupt register RCU_INT.

The PLL clock is monitored by IRC8M.

If LOPRSTEN is 0, the PLL lost flag, PLLMIF, in the interrupt register, RCU_INT, will be set and the PLL failure interrupt will be generated. The PLL will be automatically disabled. If the PLL is selected as the clock source of CK_SYS, the PLL failure will force the CK_SYS source to IRC8M and the PLL will be disabled automatically.

If LOPRSTEN is 1, Loss-of-PLL reset will generate.

HXTAL Clock Monitor (CKM)

The HXTAL clock monitor function is enabled by the HXTAL clock monitor enable bit, CKMEN, in the control register, RCU_CTL. This function should be enabled after the HXTAL start-up

delay and disabled when the HXTAL is stopped. Once the HXTAL failure is detected, Loss-of-HXTAL reset will generate decided by LOHRSTEN in the reset source /clock register, RCU_RSTSCK. Or an NMI Interrupt may generate decided by the CKMNMIIE bit in SYSCFG_CFG3 register.

If LOHRSTEN is 0, the HXTAL Clock Stuck Flag, CKMIF, in the interrupt register, RCU_INT, will be set and the HXTAL failure event will be generated. The HXTAL will be automatically disabled. This failure interrupt is connected to the Non-Maskable interrupt, NMI, of the Cortex®-M33. If the HXTAL is selected as the clock source of CK_SYS or PLL, the HXTAL failure will force the CK_SYS source to IRC8M and the PLL will be disabled automatically.

If LOHRSTEN is 1, Loss-of-HXTAL reset will generate.

LXTAL Clock Monitor (LCKM)

A clock monitor on LXTAL can be activated by software writing the LCKMEN, in the control register, RCU_CTL. LCKMEN can not be enabled before LXTAL and IRC40K are enabled and ready.

A 4-bits plus one counter will work at IRC40K domain when LCKMEN enable. If the LXTAL clock has stuck at 0/1 error or slow down about 20KHz, the counter will overflow. The LXTAL clock failure will been found.

Clock Output Capability

The clock output capability is ranging from 32 kHz to 100 MHz. There are several clock signals can be selected via the CK_OUT clock source selection bits, CKOUTSEL, in the configuration register 0 (RCU_CFG0). The corresponding GPIO pin should be configured in the properly alternate function I/O (AFIO) mode to output the selected clock signal.

Table 5-1. Clock source select

Clock Source Selection bits	Clock Source
000	No Clock
001	Reserved
010	CK_IRC40K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M
110	CK_HXTAL
111	CK_PLL or CK_PLL/2

The CK_OUT frequency can be reduced by a configurable binary divider, controlled by the CKOUTDIV[2:0] bits, in the configuration register 0(RCU_CFG0).

Deep-sleep mode clock control

When the MCU is in Deep-sleep mode, the USART0/1/2 can wake up the MCU, when their clock is provided by LXTAL clock and LXTAL clock is enable.

If the USART0/1/2 clock is selected IRC8M clock in Deep-sleep mode, they have capable of

open IRC8M clock or close IRC8M clock, which used to the USART0/1/2 to wake up the Deep-sleep mode.

Voltage control

The core domain voltage in Deep-sleep mode can be controlled by DSLPVS[1:0] bits in the Deep-sleep mode voltage register (RCU_DSV).

Table 5-2. Core domain voltage selected in Deep-sleep mode

DSLPVS[1:0]	Deep-sleep mode voltage(V)
00	0.8
01	0.9
10	1.0
11	1.1

The RCU_DSV register are protected by voltage key register (RCU_VKEY). Only after write 0x1A2B3C4D to the RCU_VKEY register, the RCU_DSV register can be write.

5.3. Register definition

RCU base address: 0x4002 1000

5.3.1. Control register (RCU_CTL)

Address offset: 0x00

Reset value: 0x0000 XX83 where X is undefined.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLLSTB	PLLEN	Reserved	HXTALSCAL	LCKMEN	PLLMEN	CKMEN	HXTALBPS	HXTALSTB	HXTALEN
						r	rw		rw	rw	rw	rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC8MCALIB[7:0]								IRC8MADJ[4:0]					Reserved	IRC8MSTB	IRC8MEN
r								rw						r	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	PLLSTB	PLL clock stabilization flag Set by hardware to indicate if the PLL output clock is stable and ready for use. 0: PLL is not stable 1: PLL is stable
24	PLLEN	PLL enable Set and reset by software. This bit cannot be reset if the PLL clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL is switched off 1: PLL is switched on
23	Reserved	Must be kept at reset value.
22	HXTALSCAL	HXTAL frequency scale select The HXTALSCAL bit can be written only if the HXTALEN is 0. When the HXTAL frequency is equal to 8MHz, this bit must be set to 0. This bit needs to configured only when HXTALBPS bit is 0. 0: HXTAL scale is 2 ~ 8MHz 1: HXTAL scale is 8 ~ 40MHz
21	LCKMEN	LXTAL clock monitor enable 0: Disable the external 32.768k LXTAL clock monitor 1: Enable the external 32.768k LXTAL clock monitor This bit cannot be set to 1, if the LXTAL clock or IRC40K clock is disabled. LCKMEN enable the hardware detects that the LXTAL clock is stuck at a low/high

		state or slow down to about 20KHz.
20	PLLMEN	PLL clock monitor enable 0: Disable the PLL clock monitor 1: Enable the PLL clock monitor PLLMEN enable the hardware detects that the PLL clock is stuck at a low/high state. This bit cannot be set to 1, if the PLL is disabled. It reset by hardware when entering Deep-sleep or Standby mode or PLL unlock are detected.
19	CKMEN	HXTAL clock monitor enable 0: Disable the external 2 ~ 40 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the external 2 ~ 40 MHz crystal oscillator (HXTAL) clock monitor When the hardware detects that the HXTAL clock is stuck at a low or high state, the internal hardware will switch the system clock to be the internal high speed IRC8M RC clock. The way to recover the original system clock is by either an external reset, power on reset or clearing CKMIF by software. **Note:** When the HXTAL clock monitor is enabled, the hardware will automatically enable the IRC8M internal RC oscillator regardless of the control bit, IRC8MEN, state.
18	HXTALBPS	External crystal oscillator (HXTAL) clock bypass mode enable The HXTALBPS bit can be written only if the HXTALEN is 0. 0: Disable the HXTAL bypass mode 1: Enable the HXTAL bypass mode in which the HXTAL output clock is equal to the input clock.
17	HXTALSTB	External crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
16	HXTALEN	External high speed oscillator enable Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock or the PLL input clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: External 2 ~ 40 MHz crystal oscillator disabled 1: External 2 ~ 40 MHz crystal oscillator enabled
15:8	IRC8MCALIB[7:0]	Internal 8M RC oscillator calibration value register These bits are load automatically at power on.
7:3	IRC8MADJ[4:0]	Internal 8M RC oscillator clock trim adjust value These bits are set by software. The trimming value is there bits (IRC8MADJ) added to the IRC8MCALIB[7:0] bits. The trimming value should trim the IRC8M to 8 MHz ± 1%.
2	Reserved	Must be kept at reset value.

1	IRC8MSTB	IRC8M high speed internal oscillator stabilization flag Set by hardware to indicate if the IRC8M oscillator is stable and ready for use. 0: IRC8M oscillator is not stable 1: IRC8M oscillator is stable
0	IRC8MEN	Internal high speed oscillator enable Set and reset by software. This bit cannot be reset if the IRC8M clock is used as the system clock. Set by hardware when leaving Deep-sleep or Standby mode or the HXTAL clock is stuck at a low or high state when HXTALCKM is set. 0: Internal 8 MHz RC oscillator disabled 1: Internal 8 MHz RC oscillator enabled

5.3.2. Configuration register 0 (RCU_CFG0)

Address offset: 0x04

Reset value: 0x0002 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV	CKOUTDIV[2:0]			PLLMF[4]	CKOUTSEL[2:0]			Reserved	PLLMF[3:0]					DPLL	PLLSEL
rw	rw			rw	rw				rw					rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]		
	rw			rw			rw				r		rw		

Bits	Fields	Descriptions
31	PLLDV	The CK_PLL divide by 1 or 2 for CK_OUT 0: CK_PLL divide by 2 for CK_OUT 1: CK_PLL divide by 1 for CK_OUT
30:28	CKOUTDIV[2:0]	The CK_OUT divider which the CK_OUT frequency can be reduced, see bits 26:24 of RCU_CFG0 for CK_OUT. 000: The CK_OUT is divided by 1 001: The CK_OUT is divided by 2 010: The CK_OUT is divided by 4 011: The CK_OUT is divided by 8 100: The CK_OUT is divided by 16 101: The CK_OUT is divided by 32 110: The CK_OUT is divided by 64 111: The CK_OUT is divided by 128
27	PLLMF[4]	Bit 4 of PLLMF register see bits 21:18 of RCU_CFG0.
26:24	CKOUTSEL[2:0]	CK_OUT clock source selection Set and reset by software.

		000: No clock selected 001: Reserved 010: Internal 40K RC oscillator clock selected 011: External low speed oscillator clock selected 100: System clock selected 101: Internal 8MHz RC oscillator clock selected 110: External high speed oscillator clock selected 111: (CK_PLL / 2) or CK_PLL selected depend on PLLDV
23:22	Reserved	Must be kept at reset value
21:18	PLLMF[3:0]	PLL multiply factor These bits and bit 27 of RCU_CFG0 are written by software to define the PLL multiplication factor. 00000: (PLL source clock x 2) 00001: (PLL source clock x 3) 00010: (PLL source clock x 4) 00011: (PLL source clock x 5) 00100: (PLL source clock x 6) 00101: (PLL source clock x 7) 00110: (PLL source clock x 8) 00111: (PLL source clock x 9) 01000: (PLL source clock x 10) 01001: (PLL source clock x 11) 01010: (PLL source clock x 12) 01011: (PLL source clock x 13) 01100: (PLL source clock x 14) 01101: (PLL source clock x 15) 01110: (PLL source clock x 16) 01111: (PLL source clock x 16) 10000: (PLL source clock x 17) 10001: (PLL source clock x 18) 10010: (PLL source clock x 19) 10011: (PLL source clock x 20) 10100: (PLL source clock x 21) 10101: (PLL source clock x 22) 10110: (PLL source clock x 23) 10111: (PLL source clock x 24) 11000: (PLL source clock x 25) 11001: (PLL source clock x 26) 11010: (PLL source clock x 27) 11011: (PLL source clock x 28) 11100: (PLL source clock x 29) 11101: (PLL source clock x 30) 11110: (PLL source clock x 31)

		11111: (PLL source clock x 31) **Note:** The PLL output frequency must not exceed 100 MHz.
17	DPLL	Double PLL clock 0: Double PLL clock 1: PLL clock
16	PLLSEL	PLL clock source selection Set and reset by software to control the PLL clock source. 0: (IRC8M / 2) clock selected as source clock of PLL 1: HXTAL selected as source clock of PLL
15:14	Reserved	Must be kept at reset value
13:11	APB2PSC[2:0]	APB2 prescaler selection Set and reset by software to control the APB2 clock division ratio. 0xx: CK_AHB selected 100: (CK_AHB / 2) selected 101: (CK_AHB / 4) selected 110: (CK_AHB / 8) selected 111: (CK_AHB / 16) selected
10:8	APB1PSC[2:0]	APB1 prescaler selection Set and reset by software to control the APB1 clock division ratio. 0xx: CK_AHB selected 100: (CK_AHB / 2) selected 101: (CK_AHB / 4) selected 110: (CK_AHB / 8) selected 111: (CK_AHB / 16) selected
7:4	AHBPSC[3:0]	AHB prescaler selection Set and reset by software to control the AHB clock division ratio 0xxx: CK_SYS selected 1000: (CK_SYS / 2) selected 1001: (CK_SYS / 4) selected 1010: (CK_SYS / 8) selected 1011: (CK_SYS / 16) selected 1100: (CK_SYS / 64) selected 1101: (CK_SYS / 128) selected 1110: (CK_SYS / 256) selected 1111: (CK_SYS / 512) selected
3:2	SCSS[1:0]	System clock switch status Set and reset by hardware to indicate the clock source of system clock. 00: Select CK_IRC8M as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_PLL as the CK_SYS source

		11: Reserved
1:0	SCS[1:0]	System clock switch Set by software to select the CK_SYS source. Because the change of CK_SYS has inherent latency, software should read SCSS to confirm whether the switching is complete or not. The switch will be forced to IRC8M when leaving Deep-sleep and Standby mode or by HXTAL clock monitor when the HXTAL failure is detected and the HXTAL is selected as the clock source of CK_SYS or PLL. 00: Select CK_IRC8M as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_PLL as the CK_SYS source 11: Reserved

5.3.3. Interrupt register (RCU_INT)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CKMIC	PLLMIC	LCKMIC	PLL STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLLMIE	LCKMIE	PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	CKMIF	PLLMIF	LCKMIF	PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23	CKMIC	HXTAL clock stuck interrupt clear Write 1 by software to reset the CKMIF flag. 0: Not reset CKMIF flag 1: Reset CKMIF flag
22	PLLMIC	PLL clock monitor interrupt clear Write 1 by software to reset the PLLMIF flag. 0: Not reset PLLMIF flag 1: Reset PLLMIF flag
21	LCKMIC	LXTAL clock monitor interrupt clear Write 1 by software to reset the LCKMIF flag. 0: Not reset LCKMIF flag 1: Reset LCKMIF flag
20	PLLSTBIC	PLL stabilization interrupt clear

		Write 1 by software to reset the PLLSTBIF flag. 0: Not reset PLLSTBIF flag 1: Reset PLLSTBIF flag
19	HXTALSTBIC	HXTAL stabilization interrupt clear Write 1 by software to reset the HXTALSTBIF flag. 0: Not reset HXTALSTBIF flag 1: Reset HXTALSTBIF flag
18	IRC8MSTBIC	IRC8M stabilization interrupt clear Write 1 by software to reset the IRC8MSTBIF flag. 0: Not reset IRC8MSTBIF flag 1: Reset IRC8MSTBIF flag
17	LXTALSTBIC	LXTAL stabilization interrupt clear Write 1 by software to reset the LXTALSTBIF flag. 0: Not reset LXTALSTBIF flag 1: Reset LXTALSTBIF flag
16	IRC40KSTBIC	IRC40K stabilization interrupt clear Write 1 by software to reset the IRC40KSTBIF flag. 0: Not reset IRC40KSTBIF flag 1: Reset IRC40KSTBIF flag
15	Reserved	Must be kept at reset value
14	PLLMIE	PLL clock monitor interrupt enable Set and reset by software to enable/disable the PLL clock monitor interrupt. 0: Disable the PLL clock monitor interrupt 1: Enable the PLL clock monitor interrupt
13	LCKMIE	LXTAL clock monitor interrupt enable Set and reset by software to enable/disable the LXTAL clock monitor interrupt. 0: Disable the LXTAL clock monitor interrupt 1: Enable the LXTAL clock monitor interrupt
12	PLLSTBIE	PLL stabilization interrupt enable Set and reset by software to enable/disable the PLL stabilization interrupt. 0: Disable the PLL stabilization interrupt 1: Enable the PLL stabilization interrupt
11	HXTALSTBIE	HXTAL stabilization interrupt enable Set and reset by software to enable/disable the HXTAL stabilization interrupt 0: Disable the HXTAL stabilization interrupt 1: Enable the HXTAL stabilization interrupt
10	IRC8MSTBIE	IRC8M stabilization interrupt enable Set and reset by software to enable/disable the IRC8M stabilization interrupt 0: Disable the IRC8M stabilization interrupt

		1: Enable the IRC8M stabilization interrupt
9	LXTALSTBIE	LXTAL stabilization interrupt enable LXTAL stabilization interrupt enable/disable control 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt
8	IRC40KSTBIE	IRC40K stabilization interrupt enable IRC40K stabilization interrupt enable/disable control 0: Disable the IRC40K stabilization interrupt 1: Enable the IRC40K stabilization interrupt
7	CKMIF	HXTAL clock stuck interrupt flag Set by hardware when the HXTAL clock is stuck. Reset by software when setting the CKMIC bit. Reset by software when setting the CKMNMIIE. 0: Clock operating normally 1: HXTAL clock stuck
6	PLLMIF	PLL clock monitor interrupt flag Set by hardware when PLL clock is stuck. Reset by software when setting the PLLMIC bit. 0: PLL clock operating normally 1: PLL clock stuck
5	LCKMIF	LXTAL clock monitor interrupt flag Set by hardware when LXTAL clock is stuck. Reset by software when setting the LCKMIC bit. 0: LXTAL clock operating normally 1: LXTAL clock stuck
4	PLLSTBIF	PLL stabilization interrupt flag Set by hardware when the PLL is stable and the PLLSTBIE bit is set. Reset by software when setting the PLLSTBIC bit. 0: No PLL stabilization interrupt generated 1: PLL stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the external 2 ~ 40 MHz crystal oscillator clock is stable and the HXTALSTBIE bit is set. Reset by software when setting the HXTALSTBIC bit. 0: No HXTAL stabilization interrupt generated 1: HXTAL stabilization interrupt generated
2	IRC8MSTBIF	IRC8M stabilization interrupt flag Set by hardware when the internal 8 MHz RC oscillator clock is stable and the IRC8MSTBIE bit is set. Reset by software when setting the IRC8MSTBIC bit.

		0: No IRC8M stabilization interrupt generated 1: IRC8M stabilization interrupt generated
1	LXTALSTBIF	LXTAL stabilization interrupt flag Set by hardware when the external 32,768 Hz crystal oscillator clock is stable and the LXTALSTBIE bit is set. Reset by software when setting the LXTALSTBIC bit. 0: No LXTAL stabilization interrupt generated 1: LXTAL stabilization interrupt generated
0	IRC40KSTBIF	IRC40K stabilization interrupt flag Set by hardware when the internal 40kHz RC oscillator clock is stable and the IRC40KSTBIE bit is set. Reset by software when setting the IRC40KSTBIC bit. 0: No IRC40K stabilization clock ready interrupt generated 1: IRC40K stabilization interrupt generated

5.3.4. APB2 reset register (RCU_APB2RST)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN1RST	CAN0RST	Reserved								TIMER20 RST	TIMER19 RST	Reserved			
rw	rw									rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART0 RST	TIMER7 RST	SPI0 RST	TIMER0 RST	ADC1 RST	ADC0 RST	Reserved							CMPRST	CFGRST
	rw	rw	rw	rw	rw	rw								rw	rw

Bits	Fields	Descriptions
31	CAN1RST	CAN1 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN1
30	CAN0RST	CAN0 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN0
29:22	Reserved	Must be kept at reset value
21	TIMER20RST	TIMER20 reset This bit is set and reset by software.

		0: No reset 1: Reset the TIMER20
20	TIMER19RST	TIMER19 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER19
19:15	Reserved	Must be kept at reset value
14	USART0RST	USART0 reset This bit is set and reset by software. 0: No reset 1: Reset the USART0
13	TIMER7RST	TIMER7 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER7
12	SPI0RST	SPI0 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI0
11	TIMER0RST	TIMER0 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER0
10	ADC1RST	ADC1 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC1
9	ADC0RST	ADC0 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC0
8:2	Reserved	Must be kept at reset value
1	CMPRST	Comparator reset This bit is set and reset by software. 0: No reset 1: Reset comparator
0	CFGRST	System configuration reset This bit is set and reset by software. 0: No reset

1: Reset system configuration

5.3.5. APB1 reset register (RCU_APB1RST)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved.		DAC RST	PMU RST	BKP RST	Reserved				I2C1 RST	I2C0 RST	Reserved		USART2 RST	USART1 RST	Reserved
		rw	rw	rw	rw				rw	rw			rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI1 RST	Reserved		WWDGT RST	Reserved					TIMER6 RST	TIMER5 RST	Reserved			TIMER1 RST
	rw			rw						rw	rw				rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset DAC
28	PMURST	Power control reset This bit is set and reset by software. 0: No reset 1: Reset power control unit
27	BKPRST	Back-up control reset This bit is set and reset by software. 0: No reset 1: Reset Back-up control unit
26:23	Reserved	Must be kept at reset value
22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset I2C1
21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset I2C0

20:19	Reserved	Must be kept at reset value
18	USART2RST	USART2 reset This bit is set and reset by software. 0: No reset 1: Reset USART2
17	USART1RST	USART1 reset This bit is set and reset by software. 0: No reset 1: Reset USART1
16:15	Reserved	Must be kept at reset value
14	SPI1RST	SPI1 reset This bit is set and reset by software. 0: No reset 1: Reset SPI1
13:12	Reserved	Must be kept at reset value
11	WWDGTRST	Window watchdog timer reset This bit is set and reset by software. 0: No reset 1: Reset window watchdog timer
10:6	Reserved	Must be kept at reset value
5	TIMER6RST	TIMER6 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER6 timer
4	TIMER5RST	TIMER5 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER5 timer
3:1	Reserved	Must be kept at reset value
0	TIMER1RST	TIMER1 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER1 timer

5.3.6. AHB enable register (RCU_AHBEN)

Address offset: 0x14

Reset value: 0x0000 0014

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PFEN	PEEN	PDEN	PCEN	PBEN	PAEN	Reserved
									rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MFCOMEN	Reserved							CRCEN	Reserved	FMCSPEN	DMAMUXEN	SRAMSPEN	DMA1EN	DMA0EN
	rw								rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	PFEN	GPIO port F clock enable This bit is set and reset by software. 0: Disabled GPIO port F clock 1: Enabled GPIO port F clock
21	PEEN	GPIO port E clock enable This bit is set and reset by software. 0: Disabled GPIO port E clock 1: Enabled GPIO port E clock
20	PDEN	GPIO port D clock enable This bit is set and reset by software. 0: Disabled GPIO port D clock 1: Enabled GPIO port D clock
19	PCEN	GPIO port C clock enable This bit is set and reset by software. 0: Disabled GPIO port C clock 1: Enabled GPIO port C clock
18	PBEN	GPIO port B clock enable This bit is set and reset by software. 0: Disabled GPIO port B clock 1: Enabled GPIO port B clock
17	PAEN	GPIO port A clock enable This bit is set and reset by software. 0: Disabled GPIO port A clock 1: Enabled GPIO port A clock
16:15	Reserved	Must be kept at reset value
14	MFCOMEN	MFCOM port A clock enable This bit is set and reset by software. 0: Disabled MFCOM port A clock 1: Enabled MFCOM port A clock

13:7	Reserved	Must be kept at reset value
6	CRCEN	CRC clock enable This bit is set and reset by software. 0: Disabled CRC clock 1: Enabled CRC clock
5	Reserved	Must be kept at reset value
4	FMCSPEN	FMC clock enable This bit is set and reset by software to enable/disable FMC clock during Sleep mode. 0: Disabled FMC clock during Sleep mode 1: Enabled FMC clock during Sleep mode
3	DMAMUXEN	DMAMUX clock enable This bit is set and reset by software. 0: Disabled DMAMUX clock 1: Enabled DMAMUX clock
2	SRAMSPEN	SRAM interface clock enable This bit is set and reset by software to enable/disable SRAM interface clock during Sleep mode. 0: Disabled SRAM interface clock during sleep mode. 1: Enabled SRAM interface clock during sleep mode
1	DMA1EN	DMA1 clock enable This bit is set and reset by software. 0: Disabled DMA1 clock 1: Enabled DMA1 clock
0	DMA0EN	DMA0 clock enable This bit is set and reset by software. 0: Disabled DMA0 clock 1: Enabled DMA0 clock

5.3.7. APB2 enable register (RCU_APB2EN)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CAN1EN	CAN0EN	TRIGSEL EN	Reserved							TIMER20 EN	TIMER19E N	Reserved			
rw	rw	rw								rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	USART0EN	TIMER7EN	SPI0EN	TIMER0EN	ADC1EN	ADC0EN	Reserved	CMPEN	CFGEN
	rw	rw	rw	rw	rw	rw		rw	rw

Bits	Fields	Descriptions
31	CAN1EN	CAN1 clock enable This bit is set and reset by software. 0: Disabled CAN1 clock 1: Enabled CAN1 clock
30	CAN0EN	CAN0 clock enable This bit is set and reset by software. 0: Disabled CAN0 clock 1: Enabled CAN0 clock
29	TRIGSELEN	TRIGSEL clock enable This bit is set and reset by software. 0: Disabled TRIGSEL clock 1: Enabled TRIGSEL clock
28:22	Reserved	Must be kept at reset value
21	TIMER20EN	TIMER20 timer clock enable This bit is set and reset by software. 0: Disabled TIMER20 timer clock 1: Enabled TIMER20 timer clock
20	TIMER19EN	TIMER19 timer clock enable This bit is set and reset by software. 0: Disabled TIMER19 timer clock 1: Enabled TIMER19 timer clock
19:15	Reserved	Must be kept at reset value
14	USART0EN	USART0 clock enable This bit is set and reset by software. 0: Disabled USART0 clock 1: Enabled USART0 clock
13	TIMER7EN	TIMER7 timer clock enable This bit is set and reset by software. 0: Disabled TIMER7 timer clock 1: Enabled TIMER7 timer clock
12	SPI0EN	SPI0 clock enable This bit is set and reset by software. 0: Disabled SPI0 clock 1: Enabled SPI0 clock

11	TIMER0EN	TIMER0 timer clock enable This bit is set and reset by software. 0: Disabled TIMER0 timer clock 1: Enabled TIMER0 timer clock
10	ADC1EN	ADC1 interface clock enable This bit is set and reset by software. 0: Disabled ADC1 interface clock 1: Enabled ADC1 interface clock
9	ADC0EN	ADC interface clock enable This bit is set and reset by software. 0: Disabled ADC0 interface clock 1: Enabled ADC0 interface clock
8:2	Reserved	Must be kept at reset value
1	CMPEN	Comparator clock enable This bit is set and reset by software. 0: Disabled comparator clock 1: Enabled comparator clock
0	CFGEN	System configuration clock enable This bit is set and reset by software. 0: Disabled system configuration clock 1: Enabled system configuration clock

5.3.8. APB1 enable register (RCU_APB1EN)

Address offset:0x1C

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DACEN	PMUEN	BKPEN	Reserved				I2C1EN	I2C0EN	Reserved		USART2EN	USART1EN	Reserved
		rw	rw	rw					rw	rw			rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI1EN	Reserved		WWDGTEN	Reserved					TIMER6EN	TIMER5EN	Reserved			TIMER1EN
	rw			rw						rw	rw				rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29	DACEN	DAC clock enable This bit is set and reset by software.

		0: Disabled DAC clock 1: Enabled DAC clock
28	PMUEN	Power interface clock enable This bit is set and reset by software. 0: Disabled Power interface clock 1: Enabled Power interface clock
27	BKPEN	Back-up interface clock enable This bit is set and reset by software. 0: Disabled Back-up interface clock 1: Enabled Back-up interface clock
26:23	Reserved	Must be kept at reset value
22	I2C1EN	I2C1 clock enable This bit is set and reset by software. 0: Disabled I2C1 clock 1: Enabled I2C1 clock
21	I2C0EN	I2C0 clock enable This bit is set and reset by software. 0: Disabled I2C0 clock 1: Enabled I2C0 clock
20:19	Reserved	Must be kept at reset value
18	USART2EN	USART2 clock enable This bit is set and reset by software. 0: Disabled USART2 clock 1: Enabled USART2 clock
17	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
16:15	Reserved	Must be kept at reset value
14	SPI1EN	SPI1 clock enable This bit is set and reset by software. 0: Disabled SPI1 clock 1: Enabled SPI1 clock
13:12	Reserved	Must be kept at reset value
11	WWDGTEN	Window watchdog timer clock enable This bit is set and reset by software. 0: Disabled window watchdog timer clock 1: Enabled window watchdog timer clock

10:6	Reserved	Must be kept at reset value
5	TIMER6EN	TIMER6 timer clock enable This bit is set and reset by software. 0: Disabled TIMER6 timer clock 1: Enabled TIMER6 timer clock
4	TIMER5EN	TIMER5 timer clock enable This bit is set and reset by software. 0: Disabled TIMER5 timer clock 1: Enabled TIMER5 timer clock
3:1	Reserved	Must be kept at reset value
0	TIMER1EN	TIMER1 timer clock enable This bit is set and reset by software. 0: Disabled TIMER1 timer clock 1: Enabled TIMER1 timer clock

5.3.9. Backup domain control register (RCU_BDCTL)

Address offset: 0x20

Reset value: 0x0000 0018, reset by Backup domain Reset.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

Note: The LXTALEN, LXTALBPS, RTCSRC and RTCEN bits of the backup domain control register (BDCTL) are only reset after a backup domain reset. These bits can be modified only when the BKPWEN bit in the power control register (PMU_CTL) has to be set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															BKPRST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSRC[1:0]		Reserved			LXTALDRI[1:0]		LXTALBPS	LXTALSTB	LXTALEN
rw						rw					rw		rw	r	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	BKPRST	Backup domain reset This bit is set and reset by software. 0: No reset 1: Resets backup domain
15	RTCEN	RTC clock enable This bit is set and reset by software. 0: Disabled RTC clock

		1: Enabled RTC clock
14:10	Reserved	Must be kept at reset value
9:8	RTCSRC[1:0]	RTC clock entry selection Set and reset by software to control the RTC clock source. 00: No clock selected 01: CK_LXTAL selected as RTC source clock 10: CK_IRC40K selected as RTC source clock 11: (CK_HXTAL / 128) selected as RTC source clock
7:5	Reserved	Must be kept at reset value
4:3	LXTALDRI[1:0]	LXTAL drive capability Set and reset by software. Backup domain reset reset this value. 00: Lower driving capability 01: Medium low driving capability 10: Medium high driving capability 11: Higher driving capability (reset value) **Note:** The LXTALDRI is not in bypass mode.
2	LXTALBPS	LXTAL bypass mode enable Set and reset by software. 0: Disable the LXTAL Bypass mode 1: Enable the LXTAL Bypass mode **Note:** In this series, the LXTALBPS bit must be configured to 1.
1	LXTALSTB	External low-speed oscillator stabilization Set by hardware to indicate if the LXTAL output clock is stable and ready for use. 0: LXTAL is not stable 1: LXTAL is stable
0	LXTALEN	LXTAL enable Set and reset by software. 0: Disable LXTAL 1: Enable LXTAL

5.3.10. Reset source /clock register (RCU_RSTSCK)

Address offset: 0x24

Reset value: 0x0C80 0000, reset flags reset by power reset only, other reset by system reset.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	OBL RSTF	RSTFC	V11 RSTF	LOP RSTF	LOH RSTF	ECC RSTF	LVD RSTF	LOCKUP RSTF	BOR RSTF	Reserved
r	r	r	r	r	r	r	rw	r	r	r	r	r	r	r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LOPRSTEN	LOHRSTEN	ECCRSTEN	LVDRSTEN	LOCKUPRSTEN	Reserved								IRC40KSTB	IRC40KEN
	rw	rw	rw	rw	rw									r	rw

Bits	Fields	Descriptions
31	LPRSTF	Low-power reset flag Set by hardware when Deep-sleep /standby reset generated. Reset by writing 1 to the RSTFC bit. 0: No Low-power management reset generated 1: Low-power management reset generated
30	WWDGTRSTF	Window watchdog timer reset flag Set by hardware when a window watchdog timer reset generated. Reset by writing 1 to the RSTFC bit. 0: No window watchdog reset generated 1: Window watchdog reset generated
29	FWDGTRSTF	Free Watchdog timer reset flag Set by hardware when a Free Watchdog timer generated. Reset by writing 1 to the RSTFC bit. 0: No Free Watchdog timer reset generated 1: Free Watchdog timer reset generated
28	SWRSTF	Software reset flag Set by hardware when a software reset generated. Reset by writing 1 to the RSTFC bit. 0: No software reset generated 1: Software reset generated
27	PORRSTF	Power reset flag Set by hardware when a power reset generated. Reset by writing 1 to the RSTFC bit. 0: No power reset generated 1: Power reset generated
26	EPRSTF	External PIN reset flag Set by hardware when an External PIN generated. Reset by writing 1 to the RSTFC bit. 0: No external PIN reset generated 1: External PIN reset generated
25	OBLRSTF	Option byte loader reset flag Set by hardware when an option byte loader generated. Reset by writing 1 to the RSTFC bit. 0: No option byte loader reset generated 1: Option byte loader reset generated

24	RSTFC	Reset flag clear This bit is set by software to clear all reset flags. 0: Not clear reset flags 1: Clear reset flags
23	V11RSTF	1.1V domain power reset flag Set by hardware when a 1.1V domain Power reset generated. Reset by writing 1 to the RSTFC bit. 0: No 1.1V domain power reset generated 1: 1.1V domain power reset generated
22	LOPRSTF	Lost of PLL error reset flag Set by hardware when a Lost of PLL Error reset generated. Reset by writing 1 to the RSTFC bit. 0: No lost of PLL error reset generated 1: Lost of PLL error reset generated
21	LOHRSTF	Lost of HXTAL error reset flag Set by hardware when a Lost of HXTAL Error reset generated. Reset by writing 1 to the RSTFC bit. 0: No lost of HXTAL error reset generated 1: Lost of HXTAL error reset generated
20	ECCRSTF	2 bits ECC error reset flag Set by hardware when a two bit ECC Error reset generated. Reset by writing 1 to the RSTFC bit. 0: No 2 bits ECC error reset generated 1: 2 bits ECC error reset generated
19	LVDRSTF	Low voltage detect error reset flag Set by hardware when a low voltage detect error reset generated. **Note**: The low voltage reset function needs to enable LVDEN in the PMU and enable LVDRSTEN to take effect. Reset by writing 1 to the RSTFC bit. 0: No low voltage detect error reset generated 1: Low voltage detect error reset generated
18	LOCKUPRSTF	CPU Lock-Up error reset flag Set by hardware when a CPU Lock-Up error reset generated. Reset by writing 1 to the RSTFC bit. 0: No CPU Lock-Up error reset generated 1: CPU Lock-Up error reset generated
17	BORRSTF	BOR reset flag Set by hardware when a BOR reset generated. Reset by writing 1 to the RSTFC bit. 0: No BOR reset generated

		1: BOR reset generated
16:15	Reserved	Must be kept at reset value
14	LOPRSTEN	Lost of PLL reset enable 0: PLL monitor generates interrupt when error detected, if PLLMIE is 1 1: PLL monitor generates reset when error detected
13	LOHRSTEN	Lost of HXTAL reset enable 0: No reset generated 1: HXTAL Monitor generates reset when error detected
12	ECCRSTEN	ECC 2 bits error reset enable 0: No reset generated 1: ECC generates reset when ECC 2 bits error detected
11	LVDRSTEN	Low voltage detection reset enable 0: No reset generated 1: Low voltage detection generates reset when V_{DDA} is lower than pre-setting
10	LOCKUPRSTEN	CPU Lock-Up reset enable 0: No reset generated 1: CPU Lock-Up generates reset.
9:2	Reserved	Must be kept at reset value
1	IRC40KSTB	IRC40K stabilization Set by hardware to indicate if the IRC40K output clock is stable and ready for use. 0: IRC40K is not stable 1: IRC40K is stable
0	IRC40KEN	IRC40K enable Set and reset by software. 0: Disable IRC40K 1: Enable IRC40K

5.3.11. AHB reset register (RCU_AHBRST)

Address offset: 0x28

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PFRST	PERST	PDRST	PCRST	PBRST	PARST	Reserved
									rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MFCOMRST	Reserved							CRCRST	Reserved		DMAMUXRST	Reserved	DMA1RST	DMA0RST
	rw								rw			rw		rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	PFRST	GPIO port F reset This bit is set and reset by software. 0: No reset GPIO port F 1: Reset GPIO port F
21	PERST	GPIO port E reset This bit is set and reset by software. 0: No reset GPIO port E 1: Reset GPIO port E
20	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset GPIO port D 1: Reset GPIO port D
19	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset GPIO port C 1: Reset GPIO port C
18	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset GPIO port B 1: Reset GPIO port B
17	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset GPIO port A 1: Reset GPIO port A
16:15	Reserved	Must be kept at reset value
14	MFCOMRST	MFCOM reset This bit is set and reset by software. 0: No reset MFCOM module 1: Reset MFCOM module
13:7	Reserved	Must be kept at reset value
6	CRCRST	CRC reset This bit is set and reset by software. 0: No reset CRC module 1: Reset CRC module
5:4	Reserved	Must be kept at reset value

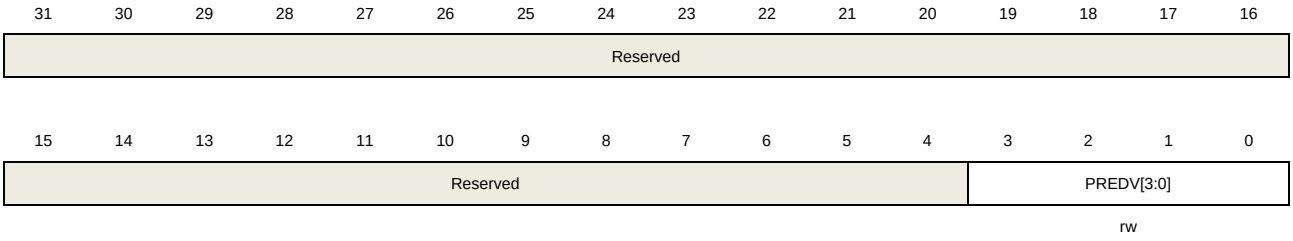
3	DMAMUXRST	DMAMUX reset This bit is set and reset by software. 0: No reset DMAMUX module 1: Reset DMAMUX module
2	Reserved	Must be kept at reset value
1	DMA1RST	DMA1 reset This bit is set and reset by software. 0: No reset DMA1 module 1: Reset DMA1 module
0	DMA0RST	DMA0 reset This bit is set and reset by software. 0: No reset DMA0 module 1: Reset DMA0 module

5.3.12. Configuration register 1 (RCU_CFG1)

Address offset: 0x2C

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)



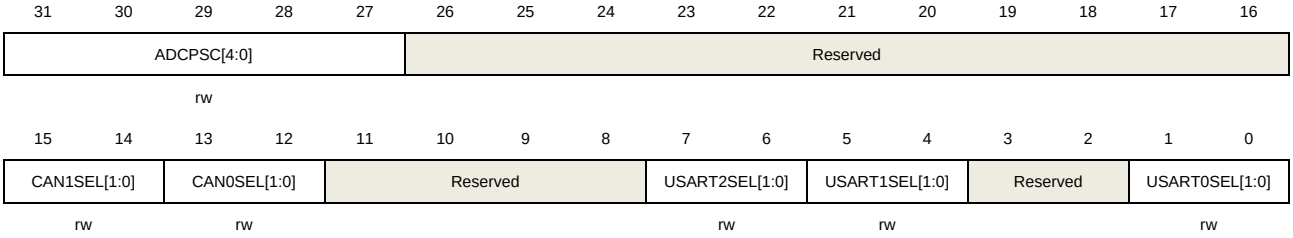
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	PREDV[3:0]	CK_HXTAL divider previous PLL This bit is set and reset by software. These bits can be written when PLL is disabled. The CK_HXTAL is divided by (PREDV + 1). 0000: Input to PLL not divided 0001: Input to PLL divided by 2 0010: Input to PLL divided by 3 0011: Input to PLL divided by 4 0100: Input to PLL divided by 5 0101: Input to PLL divided by 6 0110: Input to PLL divided by 7 0111: Input to PLL divided by 8 1000: Input to PLL divided by 9 1001: Input to PLL divided by 10

1010: Input to PLL divided by 11
1011: Input to PLL divided by 12
1100: Input to PLL divided by 13
1101: Input to PLL divided by 14
1110: Input to PLL divided by 15
1111: Input to PLL divided by 16

5.3.13. Configuration register 2 (RCU_CFG2)

Address offset: 0x30
Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)



Bits	Fields	Descriptions
31:27	ADCPSC[4:0]	ADC clock prescaler selection These bits are set and cleared by software. **Note:** These bits cannot be set to “11111”. CK_ADC = CK_AHB / (ADCPSC + 2)
29:16	Reserved	Must be kept at reset value
15:14	CAN1SEL[1:0]	CK_CAN1 clock source selection This bit is set and reset by software. 00: CK_CAN1 select CK_HXTAL 01: CK_CAN1 select CK_PCLK2 10: CK_CAN1 select CK_PCLK2/2 11: CK_CAN1 select CK_IRC8M
13:12	CAN0SEL[1:0]	CK_CAN0 clock source selection This bit is set and reset by software. 00: CK_CAN0 select CK_HXTAL 01: CK_CAN0 select CK_PCLK2 10: CK_CAN0 select CK_PCLK2/2 11: CK_CAN0 select CK_IRC8M
11:7	Reserved	Must be kept at reset value
7:6	USART2SEL[1:0]	CK_USART2 clock source selection This bit is set and reset by software.

		00: CK_USART2 select CK_HXTAL 01: CK_USART2 select CK_SYS 10: CK_USART2 select CK_LXTAL 11: CK_USART2 select CK_IRC8M
5:4	USART1SEL[1:0]	CK_USART1 clock source selection This bit is set and reset by software. 00: CK_USART1 select CK_HXTAL 01: CK_USART1 select CK_SYS 10: CK_USART1 select CK_LXTAL 11: CK_USART1 select CK_IRC8M
3:2	Reserved	Must be kept at reset value
1:0	USART0SEL[1:0]	CK_USART0 clock source selection This bit is set and reset by software. 00: CK_USART0 select CK_HXTAL 01: CK_USART0 select CK_SYS 10: CK_USART0 select CK_LXTAL 11: CK_USART0 select CK_IRC8M

5.3.14. Voltage key register (RCU_VKEY)

Address offset: 0x100

Reset value: 0x0000 0000.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

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

Bits	Fields	Descriptions
31:0	KEY[31:0]	The key of RCU_DSV register These bits are written only by software and read as 0. Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_DSV register can be written.

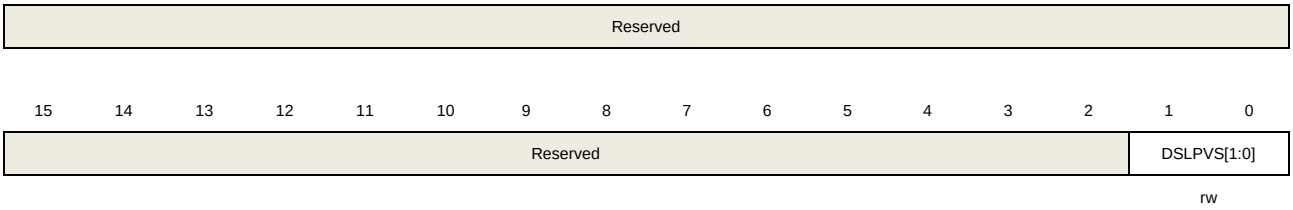
5.3.15. Deep-sleep mode voltage register (RCU_DSV)

Offset: 0x134

Reset value: 0x0000 0003

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1:0	DSLPVS[1:0]	Deep-sleep mode voltage select These bits is set and reset by software 00 : The core voltage is 0.8V in Deep-sleep mode 01 : The core voltage is 0.9V in Deep-sleep mode 10 : The core voltage is 1.0V in Deep-sleep mode 11 : The core voltage is 1.1V in Deep-sleep mode

6. Interrupt / event controller (EXTI)

6.1. Overview

Cortex®-M33 integrates the Nested Vectored Interrupt Controller (NVIC) for efficient exception and interrupts processing. NVIC facilitates low-latency exception and interrupt handling and power management controls. It's tightly coupled to the processer core. You can read the Technical Reference Manual of Cortex®-M33 for more details about NVIC.

EXTI (interrupt / event controller) contains up to 25 independent edge detectors and generates interrupt requests or events to the processer. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

6.2. Characteristics

- Cortex®-M33 system exception.
- Up to 71 maskable peripheral interrupts.
- 4 bits interrupt priority configuration—16 priority levels.
- Efficient interrupt processing.
- Support exception pre-emption and tail-chaining.
- Wake up system from power saving mode.
- Up to 25 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

6.3. Interrupts function overview

The Arm Cortex®-M33 processor and the Nested Vectored Interrupt Controller (NVIC) prioritize and handle all exceptions in Handler Mode. The processor state is automatically stored to the stack on an exception and automatically restored from the stack at the end of the Interrupt Service Routine (ISR).

The vector is fetched in parallel to the state saving, enabling efficient interrupt entry. The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoration. ***Table 6-1. NVIC exception types in Cortex®-M33*** and ***Table 6-2. Interrupt vector table*** list all exception types.

Table 6-1. NVIC exception types in Cortex®-M33

Exception type	Vector number	Priority (a)	Vector address	Description
-	0	-	0x0000_0000	Reserved
Reset	1	-3	0x0000_0004	Reset
NMI	2	-2	0x0000_0008	Non maskable interrupt.
HardFault	3	-1	0x0000_000C	All class of fault
MemManage	4	Programmable	0x0000_0010	Memory management
BusFault	5	Programmable	0x0000_0014	Prefetch fault, memory access fault
UsageFault	6	Programmable	0x0000_0018	Undefined instruction or illegal state
-	7-10	-	0x0000_001C - 0x0000_002B	Reserved
SVCall	11	Programmable	0x0000_002C	System service call via SWI instruction
Debug Monitor	12	Programmable	0x0000_0030	Debug Monitor
-	13	-	0x0000_0034	Reserved
PendSV	14	Programmable	0x0000_0038	Pendable request for system service
SysTick	15	Programmable	0x0000_003C	System tick timer

Table 6-2. Interrupt vector table

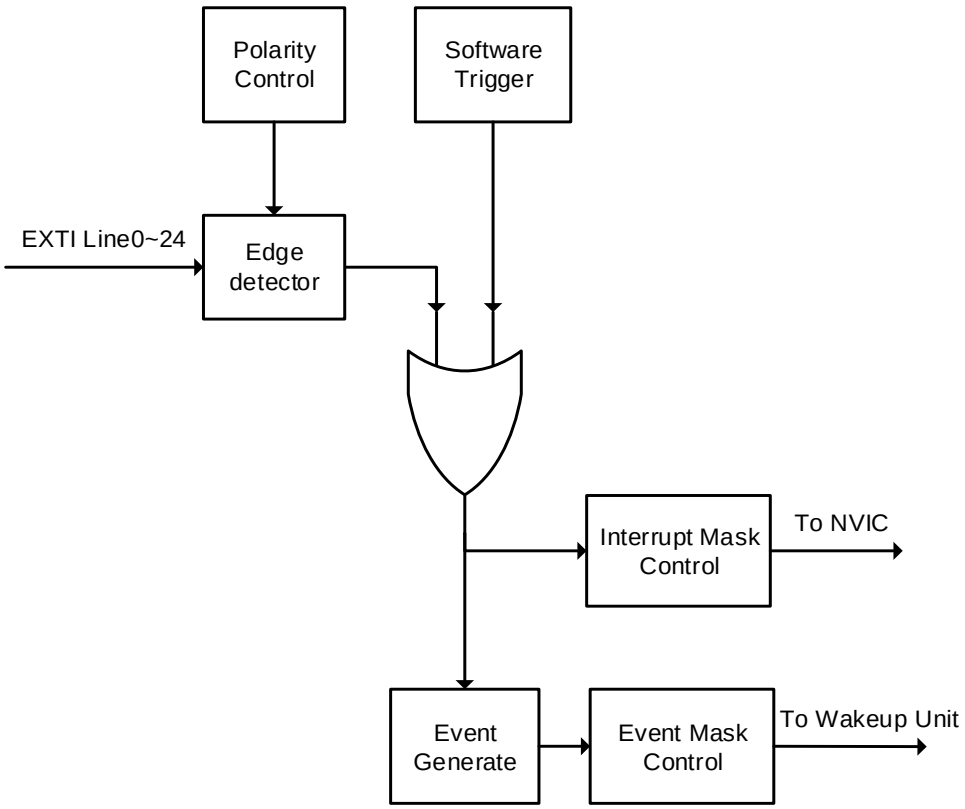
Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 0	16	Window watchdog timer interrupt	0x0000_0040
IRQ 1	17	LVD(PVD) through EXTI Line detection interrupt	0x0000_0044
IRQ 2	18	Reserved	0x0000_0048
IRQ 3	19	RTC global interrupt	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU global interrupt	0x0000_0054
IRQ 6	22	EXTI Line0 interrupt	0x0000_0058
IRQ 7	23	EXTI Line1 interrupt	0x0000_005C
IRQ 8	24	EXTI Line2 interrupt	0x0000_0060
IRQ 9	25	EXTI Line3 interrupt	0x0000_0064
IRQ 10	26	EXTI Line4 interrupt	0x0000_0068
IRQ 11	27	DMA0 Channel0 global interrupt	0x0000_006C
IRQ 12	28	DMA0 Channel1 global interrupt	0x0000_0070
IRQ 13	29	DMA0 Channel2 global interrupt	0x0000_0074
IRQ 14	30	DMA0 Channel3 global interrupt	0x0000_0078
IRQ 15	31	DMA0 Channel4 global interrupt	0x0000_007C
IRQ 16	32	DMA0 Channel5 global interrupt	0x0000_0080
IRQ 17	33	DMA0 Channel6 global interrupt	0x0000_0084

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 18	34	ADC0 and ADC1 interrupt	0x0000_0088
IRQ 19	35	CAN0 Interrupt for message buffer	0x0000_008C
IRQ 20	36	CAN0 Interrupt for Bus off / Bus off done	0x0000_0090
IRQ 21	37	CAN0 Interrupt for Error	0x0000_0094
IRQ 22	38	CAN0 Interrupt for Error in fast transmission	0x0000_0098
IRQ 23	39	CAN0 Interrupt for Transmit warning	0x0000_009C
IRQ 24	40	CAN0 Interrupt for Receive warning	0x0000_00A0
IRQ 25	41	CAN0 wakeup through EXTI Line detection interrupt	0x0000_00A4
IRQ 26	42	TIMER0 Break, update, trigger and commutation interrupt	0x0000_00A8
IRQ 27	43	TIMER0 Capture Compare interrupt	0x0000_00AC
IRQ 28	44	TIMER1 global interrupt	0x0000_00B0
IRQ 29	45	TIMER19 Break, update, trigger and commutation interrupt	0x0000_00B4
IRQ 30	46	TIMER19 Capture Compare interrupt	0x0000_00B8
IRQ 31	47	I2C0 event interrupt	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1 global interrupt	0x0000_00D0
IRQ 37	53	USART0 global interrupt	0x0000_00D4
IRQ 38	54	USART1 global interrupt	0x0000_00D8
IRQ 39	55	USART2 global interrupt	0x0000_00DC
IRQ 40	56	EXTI Line10-15 interrupt	0x0000_00E0
IRQ 41	57	EXTI Line5-9 interrupt	0x0000_00E4
IRQ 42	58	BKP Tamper	0x0000_00E8
IRQ 43	59	TIMER20 Break, update, trigger and commutation interrupt	0x0000_00EC
IRQ 44	60	TIMER20 Capture Compare interrupt	0x0000_00F0
IRQ 45	61	TIMER7 Break, update, trigger and commutation interrupt	0x0000_00F4
IRQ 46	62	TIMER7 Capture Compare interrupt	0x0000_00F8
IRQ 47	63	DMA MUX interrupt	0x0000_00FC
IRQ 48	64	SYSCFG SRAM ECC single err interrupt	0x0000_0100
IRQ 49	65	CMP through EXTI Line detection interrupt	0x0000_0104
IRQ 49	66	Reserved	0x0000_0108
IRQ 51	67	Over voltage detector through EXTI Line detection interrupt	0x0000_010C
IRQ52	68	Reserved	0x0000_0110

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ53	69	Reserved	0x0000_0114
IRQ54	70	TIMER5 interrupt, DAC global interrupt	0x0000_0118
IRQ55	71	TIMER6 global interrupt	0x0000_011C
IRQ56	72	DMA1 Channel 0 global interrupt	0x0000_0120
IRQ57	73	DMA1 Channel 1 global interrupt	0x0000_0124
IRQ58	74	DMA1 Channel 2 global interrupt	0x0000_0128
IRQ59	75	DMA1 Channel 3 global interrupt	0x0000_012C
IRQ60	76	DMA1 Channel 4 global interrupt	0x0000_0130
IRQ61	77	Reserved	0x0000_0134
IRQ62	78	CAN1 wakeup through EXTI Line detection interrupt	0x0000_0138
IRQ63	79	CAN1 Interrupt for message buffer	0x0000_013C
IRQ64	80	CAN1 Interrupt for Bus off / Bus off done	0x0000_0140
IRQ65	81	CAN1 Interrupt for Error	0x0000_0144
IRQ66	82	CAN1 Interrupt for Error in fast transmission	0x0000_014C
IRQ67	83	CAN1 Interrupt for Transmit warning	0x0000_0148
IRQ68	84	CAN1 Interrupt for Receive warning	0x0000_0150
IRQ69	85	FPU global interrupt	0x0000_0154
IRQ70	86	MFCOM interrupt	0x0000_0158

6.4. External interrupt and event (EXTI) block diagram

Figure 6-1. Block diagram of EXTI



6.5. External Interrupt and Event function overview

The EXTI contains up to 25 independent edge detectors and generates interrupts request or event to the processer. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

The EXTI trigger source includes 16 external lines from GPIO pins and 9 lines from internal modules which refers to ***Table 6-3. EXTI source*** for detail. All GPIO pins can be selected as an EXTI trigger source by configuring SYSCFG_EXTISSx registers in GPIO module (please refer to ***System configuration registers*** section for detail).

EXTI can provide not only interrupts but also event signals to the processor. The Cortex®-M33 processor fully implements the Wait For Interrupt (WFI), Wait For Event (WFE) and the Send Event (SEV) instructions. The Wake-up Interrupt Controller (WIC) enables the processor and NVIC to be put into a very low-power sleep mode leaving the WIC to identify and prioritize interrupts and event. EXTI can be used to wake up processor and the whole system when some expected event occurs, such as a special GPIO pin toggling or RTC alarm.

Hardware trigger

Hardware trigger may be used to detect the voltage change of external or internal signals. The software should follow these steps to use this function:

1. Configure EXTI sources in SYSCFG module based on application requirement.
2. Configure EXTI_RTEN and EXTI_FTEN to enable the rising or falling detection on related pins. (Software may set both RTENx and FTENx for a pin at the same time to detect both rising and falling changes on this pin).
3. Enable interrupts or events by setting related EXTI_INTEN or EXTI_EVEN bits.
4. EXTI starts to detect changes on the configured pins. The related interrupt or event will be triggered when desired change is detected on these pins. If the interrupt is triggered, the related PDx is set; if the event is triggered, the related PDx is not set. The software should response to the interrupts or events and clear these PDx bits.

Software trigger

Software may also trigger EXTI interrupts or events following these steps:

1. Enable interrupts or events by setting related EXTI_INTEN or EXTI_EVEN bits.

2. Set SWIEVx bits in EXTI_SWIEV register, the related interrupt or event will be triggered immediately. If the interrupt is triggered, the related PDx is set; if the event is triggered, the related PDx is not set. Software should response to these interrupts, and clear related PDx bits.

Table 6-3. EXTI source

EXTI Line Number	Source
0	PA0 / PB0 / PC0 / PD0 / PE0 / PF0
1	PA1 / PB1 / PC1 / PD1 / PE1 / PF1
2	PA2 / PB2 / PC2 / PD2 / PE2 / PF2
3	PA3 / PB3 / PC3 / PD3 / PE3 / PF3
4	PA4 / PB4 / PC4 / PD4 / PE4 / PF4
5	PA5 / PB5 / PC5 / PD5 / PE5 / PF5
6	PA6 / PB6 / PC6 / PD6 / PE6 / PF6
7	PA7 / PB7 / PC7 / PD7 / PE7 / PF7
8	PA8 / PB8 / PC8 / PD8 / PE8
9	PA9 / PB9 / PC9 / PD9 / PE9
10	PA10 / PB10 / PC10 / PD10 / PE10
11	PA11 / PB11 / PC11 / PD11 / PE11
12	PA12 / PB12 / PC12 / PD12 / PE12
13	PA13 / PB13 / PC13 / PD13 / PE13
14	PA14 / PB14 / PC14 / PD14 / PE14
15	PA15 / PB15 / PC15 / PD15 / PE15
16	LVD

EXTI Line Number	Source
17	RTC Alarm
18	CAN0
19	CAN1
20	CMP output
21	USART0 Wakeup
22	USART1 Wakeup
23	USART2 Wakeup
24	Over voltage

6.6. Register definition

EXTI base address: 0x4001 0400

6.6.1. Interrupt enable register (EXTI_INTEN)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	INTENx	Interrupt enable bit (x = 0…24) 0: Interrupt from linex is disabled. 1: Interrupt from linex is enabled.

6.6.2. Event enable register (EXTI_EVEN)

Address offset: 0x04
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	EVENx	Event enable bit (x = 0…24) 0: Event from linex is disabled. 1: Event from linex is enabled.

6.6.3. Rising edge trigger enable register (EXTI_RTEN)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	RTENx	Rising edge trigger enable (x = 0…24) 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt / event request

6.6.4. Falling edge trigger enable register (EXTI_FTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	FTENx	Falling edge trigger enable (x = 0…24) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request

6.6.5. Software interrupt event register (EXTI_SWIEV)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	SWIEVx	Interrupt / Event software trigger (x = 0…24) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request

6.6.6. Pending register (EXTI_PD)

Address offset: 0x14

Reset value: 0xXXXX XXXX where X is undefined.

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							PD24	PD23	PD22	PD21	PD20	PD19	PD18	PD17	PD16
							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	PDx	Interrupt pending status (x = 0…24) 0: EXTI Linex is not triggered 1: EXTI Linex is triggered This bit is cleared to 0 by writing 1 to it.

7. Trigger selection controller (TRIGSEL)

7.1. Overview

The trigger selection controller (TRIGSEL) allows software to select the trigger input signal for various peripherals. TRIGSEL provides a flexible mechanism for a peripheral to select different trigger inputs.

With TRIGSEL, there are up to 4 trigger selection outputs could be selected for each peripheral. And every output could select from different trigger input signal.

7.2. Characteristics

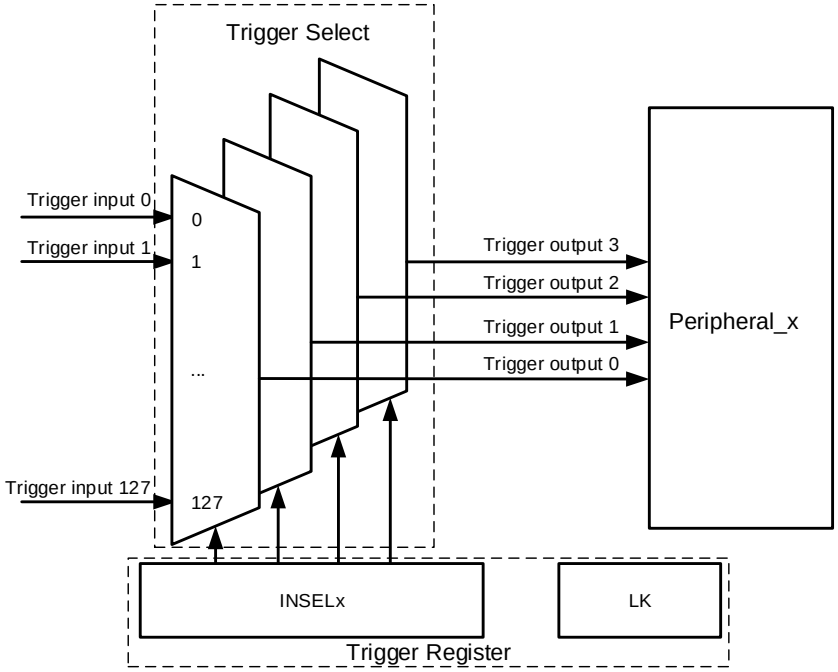
- Supports different optional trigger inputs.
- Each peripheral has its own register to select trigger input signal.
- TRIGSEL register can be configured up to 4 outputs to peripheral.
- Trigger input source could be external input signal or output of peripheral.
- Trigger selection output could be for external output or peripheral.

7.3. Function overview

With TRIGSEL, peripherals that support trigger source selection have dedicated registers to select the trigger input source. Each register can be configured with 4 outputs, which are connected to the trigger input of the peripheral. Each output can select different trigger input sources.

The ***Figure 7-1. TRIGSEL main composition example*** shows the main composition of TRIGSEL.

Figure 7-1. TRIGSEL main composition example



7.4. Internal connect

The TRIGSEL allows software to select the trigger input for peripherals. The ***Table 7-1. Trigger input bit fields selection*** gives the trigger input register selection.

Table 7-1. Trigger input bit fields selection

fields	bits value	trigger input selection
INSELx	0x00	0
	0x01	1
	0x02	TRIGSEL_IN0
	0x03	TRIGSEL_IN1
	0x04	TRIGSEL_IN2
	0x05	TRIGSEL_IN3
	0x06	TRIGSEL_IN4
	0x07	TRIGSEL_IN5
	0x08	TRIGSEL_IN6
	0x09	TRIGSEL_IN7
	0x0a	TRIGSEL_IN8
	0x0b	TRIGSEL_IN9
	0x0c	TRIGSEL_IN10
	0x0d	TRIGSEL_IN11
	0x0e	CMP_OUT
	0x0f	reserved
	0x10	LXTAL_TRG

fields	bits value	trigger input selection
	0x11	TIMER1_CH0
	0x12	TIMER1_CH1
	0x13	TIMER1_CH2
	0x14	TIMER1_CH3
	0x15	TIMER1_TRGO
	0x16	TIMER0_CH0
	0x17	TIMER0_CH1
	0x18	TIMER0_CH2
	0x19	TIMER0_CH3
	0x1a	TIMER0_MCH0
	0x1b	TIMER0_MCH1
	0x1c	TIMER0_MCH2
	0x1d	TIMER0_MCH3
	0x1e	TIMER0_TRGO
	0x1f	TIMER7_CH0
	0x20	TIMER7_CH1
	0x21	TIMER7_CH2
	0x22	TIMER7_CH3
	0x23	TIMER7_MCH0
	0x24	TIMER7_MCH1
	0x25	TIMER7_MCH2
	0x26	TIMER7_MCH3
	0x27	TIMER7_TRGO
	0x28	TIMER19_CH0
	0x29	TIMER19_CH1
	0x2a	TIMER19_CH2
	0x2b	TIMER19_CH3
	0x2c	TIMER19_MCH0
	0x2d	TIMER19_MCH1
	0x2e	TIMER19_MCH2
	0x2f	TIMER19_MCH3
	0x30	TIMER19_TRGO
	0x31	TIMER20_CH0
	0x32	TIMER20_CH1
	0x33	TIMER20_CH2
	0x34	TIMER20_CH3
	0x35	TIMER20_MCH0
	0x36	TIMER20_MCH1
	0x37	TIMER20_MCH2
	0x38	TIMER20_MCH3
	0x39	TIMER20_TRGO

fields	bits value	trigger input selection
	0x3a	TIMER5_TRGO
	0x3b	TIMER6_TRGO
	0x3c	MFCOM_TRIG0
	0x3d	MFCOM_TRIG1
	0x3e	MFCOM_TRIG2
	0x3f	MFCOM_TRIG3
	0x40	RTC_Alarm
	0x41	RTC_Second
	0x42	TRIGSEL_IN12
	0x43	TRIGSEL_IN13
	0x44~0x7f	reserved

Table 7-2. TRIGSEL input and output mapping shows the connection relationship between TRIGSEL input and output. Through the INSELx[6:0] bits of TRIGSEL register, an input trigger source can be selected for the output of TRIGSEL. Each TRIGSEL register can configure with up to 4 outputs, which are connected to the corresponding peripherals.

Table 7-2. TRIGSEL input and output mapping

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
1'b0	INSELx[6:0]	TRIGSEL_EXOUT0	output0	TRIGSEL_OUT0
1'b1			output1	TRIGSEL_OUT1
TRIGSEL_IN0			output2	TRIGSEL_OUT2
TRIGSEL_IN1			output3	TRIGSEL_OUT3
TRIGSEL_IN2		TRIGSEL_EXOUT1	output0	TRIGSEL_OUT4
TRIGSEL_IN3			output1	TRIGSEL_OUT5
TRIGSEL_IN4			output2	TRIGSEL_OUT6
TRIGSEL_IN5			output3	TRIGSEL_OUT7
TRIGSEL_IN6		TRIGSEL_ADC0	output0	ADC0_RTTRG
TRIGSEL_IN7				
TRIGSEL_IN8				
TRIGSEL_IN9				
TRIGSEL_IN10		TRIGSEL_ADC1	output0	ADC1_RTTRG
TRIGSEL_IN11				
CMP_OUT				
reserved				
LXTAL_TRG		TRIGSEL_DAC	output0	DAC_EXTRG
TIMER1_CH0				
TIMER1_CH1				
TIMER1_CH2				
TIMER1_CH3		TRIGSEL_TIMER0BRKIN	output0	TIMER0_BRKIN0
TIMER1_TRGO			output1	TIMER0_BRKIN1
TIMER0_CH0			output2	TIMER0_BRKIN2

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER0_CH1			output3	TIMER0_BRKIN3
TIMER0_CH2		TRIGSEL_TIMER7BRKIN	output0	TIMER7_BRKIN0
TIMER0_CH3			output1	TIMER7_BRKIN1
TIMER0_MCH0			output2	TIMER7_BRKIN2
TIMER0_MCH1			output3	TIMER7_BRKIN3
TIMER0_MCH2		TRIGSEL_TIMER19BRKIN	output0	TIMER19_BRKIN0
TIMER0_MCH3			output1	TIMER19_BRKIN1
TIMER0_TRGO			output2	TIMER19_BRKIN2
TIMER7_CH0			output3	TIMER19_BRKIN3
TIMER7_CH1		TRIGSEL_TIMER20BRKIN	output0	TIMER20_BRKIN0
TIMER7_CH2			output1	TIMER20_BRKIN1
TIMER7_CH3			output2	TIMER20_BRKIN2
TIMER7_MCH0			output3	TIMER20_BRKIN3
TIMER7_MCH1		TRIGSEL_MFCOM	output0	MFCOM_TRG_TIMER0
TIMER7_MCH2			output1	MFCOM_TRG_TIMER1
TIMER7_MCH3			output2	MFCOM_TRG_TIMER2
TIMER7_TRGO			output3	MFCOM_TRG_TIMER3
TIMER19_CH0		TRIGSEL_CAN0	output0	CAN0_EX_TIME_TICK
TIMER19_CH1				
TIMER19_CH2				
TIMER19_CH3				
TIMER19_MCH0		TRIGSEL_CAN1	output0	CAN1_EX_TIME_TICK
TIMER19_MCH1				
TIMER19_MCH2				
TIMER19_MCH3				
TIMER19_TRGO		TRIGSEL_TIMER0IN	output0	TIMER0_ITI0
TIMER20_CH0			output1	TIMER0_ITI1
TIMER20_CH1			output2	TIMER0_ITI2
TIMER20_CH2			output3	TIMER0_ITI3
TIMER20_CH3		TRIGSEL_TIMER7IN	output0	TIMER7_ITI0
TIMER20_MCH0			output1	TIMER7_ITI1
TIMER20_MCH1			output2	TIMER7_ITI2
TIMER20_MCH2			output3	TIMER7_ITI3
TIMER20_MCH3		TRIGSEL_TIMER19IN	output0	TIMER19_ITI0
TIMER20_TRGO			output1	TIMER19_ITI1
TIMER5_TRGO			output2	TIMER19_ITI2
TIMER6_TRGO			output3	TIMER19_ITI3
MFCOM_TRIG0		TRIGSEL_TIMER20IN	output0	TIMER20_ITI0
MFCOM_TRIG1			output1	TIMER20_ITI1
MFCOM_TRIG2			output2	TIMER20_ITI2
MFCOM_TRIG3			output3	TIMER20_ITI3

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
RTC_Alarm		TRIGSEL_TIMER1IN	output0	TIMER1_ITI0
RTC_Second			output1	TIMER1_ITI1
TRIGSEL_IN12			output2	TIMER1_ITI2
TRIGSEL_IN13			output3	TIMER1_ITI3

Note: All output can select all input as trigger source except TIMERx_ITIx and TIMERx_BRKINx. TIMERx_ITIx cannot select CMP_OUT and LXTAL_TRG, other timers CHx/MCHx signals and their own signals as trigger. TIMERx_BRKINx cannot select their own signals as trigger. When violate value is selected for these outputs, the output will be selected as 0.

When trigger input selection INSELx[6:0] bits is configured as 0, TRIGSEL trigger input is selected as low level; when configured as 1, TRIGSEL trigger input is selected as high level.

7.5. Register definition

TRIGSEL base address: 0x4001 8400

7.5.1. Trigger selection for EXTOUT0 register (TRIGSEL_EXTOUT0)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT0 register. 0: TRIGSEL_EXTOUT0 register write is enabled. 1: TRIGSEL_EXTOUT0 register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output3 is used as the source of external output3 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output2 is used as the source of external output2 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output1 can be as the source of external output1 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0

is used as the source of external output0 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.2. Trigger selection for EXTOUT1 register (TRIGSEL_EXTOUT1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT1 register. 0: TRIGSEL_EXTOUT1 register write is enabled. 1: TRIGSEL_EXTOUT1 register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output3 is used as the source of external output7 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output2 is used as the source of external output6 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output1 is used as the source of external output5 signal. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of external output4 signal. For the detailed configuration,

please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.3. Trigger selection for ADC0 register (TRIGSEL_ADC0)

Address offset: 0x08

Reset value: 0x0000 1E16

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INSEL0[6:0]						
									rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC0 register. 0: TRIGSEL_ADC0 register write is enabled. 1: TRIGSEL_ADC0 register write is disabled.
30:7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output1. The output is used as the source of ADC0_RTTRG(ADC0 routine channel group) trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.4. Trigger selection for ADC1 register (TRIGSEL_ADC1)

Address offset: 0xC

Reset value: 0x0000 1E16

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INSEL0[6:0]						
									rw						

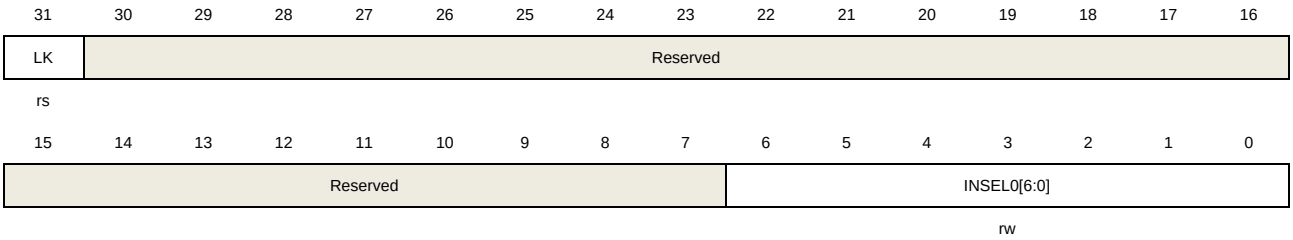
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock.

		This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC1 register. 0: TRIGSEL_ADC1 register write is enabled. 1: TRIGSEL_ADC1 register write is disabled.
30:7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output1. The output is used as the source of ADC1_RTTRG(ADC1 routine channel group) trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.5. Trigger selection for DAC register (TRIGSEL_DAC)

Address offset: 0x10
Reset value: 0x0000 0015

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC register. 0: TRIGSEL_DAC register write is enabled. 1: TRIGSEL_DAC register write is disabled.
30:7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of DAC_EXTRIG (DAC external trigger) input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.6. Trigger selection for TIMER0_ITI register (TRIGSEL_TIMER0IN)

Address offset: 0x14
Reset value: 0x2727 2727

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0IN register. 0: TRIGSEL_TIMER0IN register write is enabled. 1: TRIGSEL_TIMER0IN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER0_ITI3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER0_ITI2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER0_ITI1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER0_ITI0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.7. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0BRKIN register. 0: TRIGSEL_TIMER0BRKIN register write is enabled. 1: TRIGSEL_TIMER0BRKIN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER0_BRKIN3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER0_BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER0_BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER0_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.8. Trigger selection for TIMER7_ITI register (TRIGSEL_TIMER7IN)

Address offset: 0x1C

Reset value: 0x3030 3030

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						

rs				rw								rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7IN register. 0: TRIGSEL_TIMER7IN register write is enabled. 1: TRIGSEL_TIMER7IN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER7_ITI3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER7_ITI2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER7_ITI1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER7_ITI0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.9. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7BRKIN register. 0: TRIGSEL_TIMER7BRKIN register write is enabled. 1: TRIGSEL_TIMER7BRKIN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER7_BRKIN3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER7_BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER7_BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER7_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.10. Trigger selection for TIMER19_ITI register (TRIGSEL_TIMER19IN)

Address offset: 0x24

Reset value: 0x3939 3939

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	INSEL1[6:0]	Reserved	INSEL0[6:0]
	rw		rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER19IN register. 0: TRIGSEL_TIMER19IN register write is enabled. 1: TRIGSEL_TIMER19IN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER19_ITI3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER19_ITI2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER19_ITI1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER19_ITI0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.11. Trigger selection for TIMER19_BRKIN register (TRIGSEL_TIMER19BRKIN)

Address offset: 0x28

Reset value: 0x3939 3939

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	INSEL1[6:0]	Reserved	INSEL0[6:0]
	rw		rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER19BRKIN register. 0: TRIGSEL_TIMER19BRKIN register write is enabled. 1: TRIGSEL_TIMER19BRKIN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER19_BRKIN3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER19_BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER19_BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER19_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.12. Trigger selection for TIMER20_ITI register (TRIGSEL_TIMER20IN)

Address offset: 0x2C

Reset value: 0x1515 1515

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						

rw rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER20IN register. 0: TRIGSEL_TIMER20IN register write is enabled. 1: TRIGSEL_TIMER20IN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER20_ITI3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER20_ITI2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER20_ITI1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER20_ITI0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.13. Trigger selection for TIMER20_BRKIN register (TRIGSEL_TIMER20BRKIN)

Address offset: 0x30
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						

rw rw

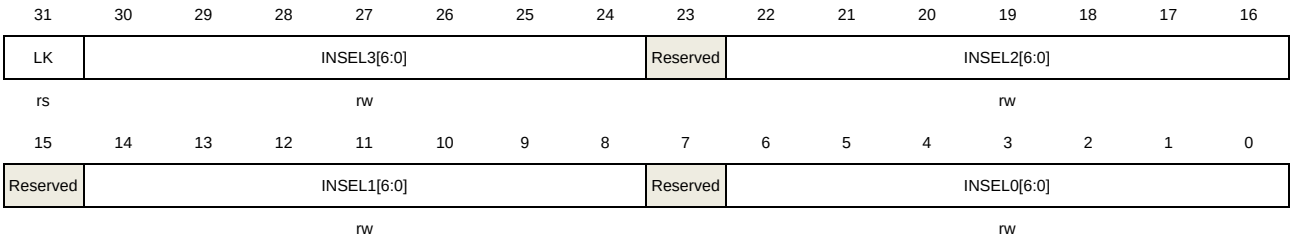
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER20BRKIN register. 0: TRIGSEL_TIMER20BRKIN register write is enabled. 1: TRIGSEL_TIMER20BRKIN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER20_BRKIN3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER20_BRKIN2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER20_BRKIN1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER20_BRKIN0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.14. Trigger selection for TIMER1_ITI register (TRIGSEL_TIMER1IN)

Address offset: 0x34

Reset value: 0x1E1E 1E1E

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER1IN register. 0: TRIGSEL_TIMER1IN register write is enabled. 1: TRIGSEL_TIMER1IN register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of TIMER1_ITI3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of TIMER1_ITI2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of TIMER1_ITI1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of TIMER1_ITI0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.15. Trigger selection for MFCOM register (TRIGSEL_MFCOM)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	INSEL3[6:0]							Reserved	INSEL2[6:0]						
rs	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INSEL1[6:0]							Reserved	INSEL0[6:0]						
	rw								rw						

Bits	Fields	Descriptions

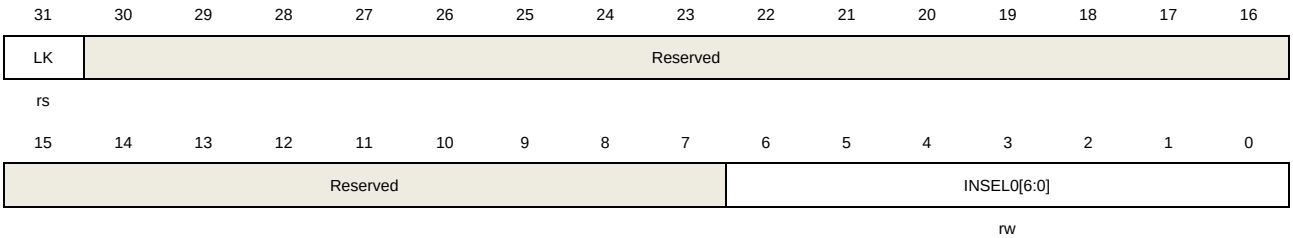
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_MFCOM register. 0: TRIGSEL_MFCOM register write is enabled. 1: TRIGSEL_MFCOM register write is disabled.
30:24	INSEL3[6:0]	Trigger input source selection for output3 These bits are used to select trigger input signal connected to output3. The output is used as the source of MFCOM_TRG_TIMER3 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
23	Reserved	Must be kept at reset value.
22:16	INSEL2[6:0]	Trigger input source selection for output2 These bits are used to select trigger input signal connected to output2. The output is used as the source of MFCOM_TRG_TIMER2 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
15	Reserved	Must be kept at reset value.
14:8	INSEL1[6:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output is used as the source of MFCOM_TRG_TIMER1 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.
7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output1. The output is used as the source of MFCOM_TRG_TIMER0 trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.16. Trigger selection for CAN0 register (TRIGSEL_CAN0)

Address offset: 0x3C

Reset value: 0x0000 003A

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	LK	TRIGSEL register lock.

		This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_CAN0 register. 0: TRIGSEL_CAN0 register write is enabled. 1: TRIGSEL_CAN0 register write is disabled.
30:7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of CAN0_EX_TIME_TICK trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

7.5.17. Trigger selection for CAN1 register (TRIGSEL_CAN1)

Address offset: 0x40

Reset value: 0x0000 003A

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved														
rs															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INSEL0[6:0]						
									rw						

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_CAN1 register. 0: TRIGSEL_CAN1 register write is enabled. 1: TRIGSEL_CAN1 register write is disabled.
30:7	Reserved	Must be kept at reset value.
6:0	INSEL0[6:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output is used as the source of CAN1_EX_TIME_TICK trigger input. For the detailed configuration, please refer to ***Table 7-1. Trigger input bit fields selection***.

8. General-purpose and alternate-function I/Os (GPIO and AFIO)

8.1. Overview

There are up to 88 general purpose I/O pins (GPIO), named PA0~PA15, PB0~PB15, PC0~PC15, PD0~PD15，PE0~PE15，PF0~ PF7 for the device to implement logic input/output functions. Each GPIO port has related control and configuration registers to satisfy the requirements of specific applications. The external interrupts on the GPIO pins of the device have related control and configuration registers in the Interrupt/Event Controller Unit (EXTI).

The GPIO ports are pin-shared with other alternative functions (AFs) to obtain maximum flexibility on the package pins. The GPIO pins can be used as alternative functional pins by configuring the corresponding registers regardless of the AF input or output pins.

Each of the GPIO pins can be configured by software as output (push-pull or open-drain), input, peripheral alternate function or analog mode. Each GPIO pin can be configured as pull-up, pull-down or no pull-up/pull-down. All GPIOs are high-current capable except for analog mode.

8.2. Characteristics

- Input/output direction control
- Schmitt trigger input function enable control
- Each pin weak pull-up/pull-down function
- Output push-pull/open drain enable control
- Output set/reset control
- External interrupt with programmable trigger edge – using EXTI configuration registers.
- Analog input/output configuration
- Alternate function input/output configuration
- Port configuration lock
- Single cycle toggle output capability

8.3. Function overview

Each of the general-purpose I/O ports can be configured as GPIO inputs, GPIO outputs, AF function or analog mode by GPIO 32-bit configuration registers (GPIOx_CTL). When select AF function, the pad input or output is decided by selected AF function output enable. When the port is output (GPIO output or AFIO output), it can be configured as push-pull or open drain mode by GPIO output mode registers (GPIOx_OMODE). And the port max speed can be configured by GPIO output speed registers (GPIOx_OSPD). Each port can be configured as floating (no pull-up and pull-down), pull-up or pull-down function by GPIO pull-up/pull-down

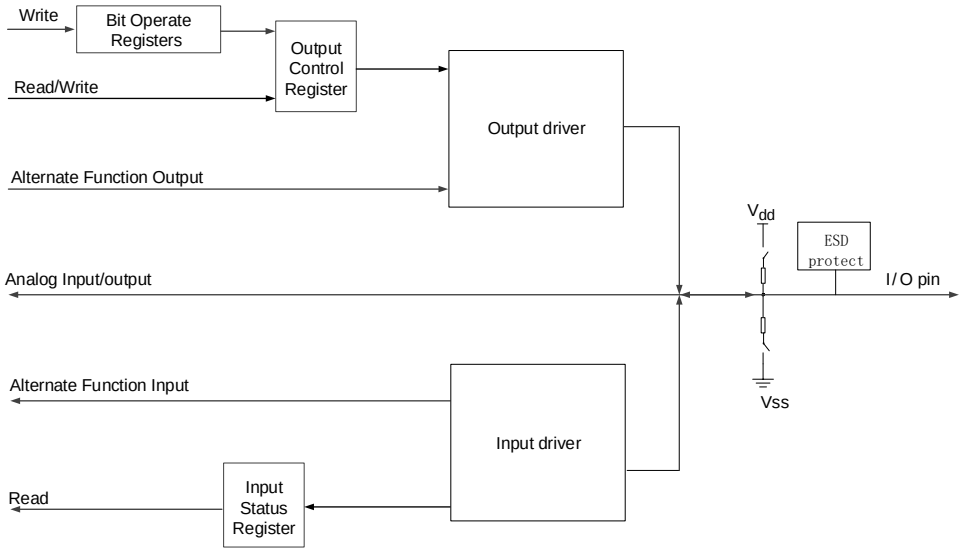
registers (GPIOx_PUD).

Table 8-1. GPIO configuration table

PAD TYPE			CTLy	OMy	PUDy
GPIO INPUT	X	Floating	00	X	00
		pull-up			01
		pull-down			10
GPIO OUTPUT	push-pull	Floating	01	0	00
		pull-up			01
		pull-down			10
	open-drain	Floating		1	00
		pull-up			01
		pull-down			10
AFIO INPUT	X	Floating	10	X	00
		pull-up			01
		pull-down			10
AFIO OUTPUT	push-pull	Floating	10	0	00
		pull-up			01
		pull-down			10
	open-drain	Floating		1	00
		pull-up			01
		pull-down			10
ANALOG	X	X	11	X	XX

*Figure 8-1. Basic structure of a general-pupose I/O*shows the basic structure of an I/O Port bit.

Figure 8-1. Basic structure of a general-pupose I/O



8.3.1. GPIO pin configuration

During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input floating mode that input disabled without Pull-Up(PU)/Pull-Down(PD) resistors. But the JTAG/Serial-Wired Debug pins are in input PU/PD mode after reset:

PB7: JTDI in PU mode
PB8: JTCK / SWCLK in PD mode
PB9: JTMS / SWDIO in PU mode
PB3: NJTRST in PU mode
PB4: JTDO in output mode

The GPIO pins can be configured as inputs or outputs. When the GPIO pins are configured as input pins, all GPIO pins have an internal weak pull-up and weak pull-down which can be chosen. And the data on the external pins can be captured at every AHB clock cycle to the port input status register (GPIOx_ISTAT).

When the GPIO pins are configured as output pins, user can configure the speed of the ports. And chooses the output driver mode: Push-Pull or Open-Drain mode. The value of the port output control register (GPIOx_OCTL) is output on the I/O pin.

There is no need to read-then-write when programming the GPIOx_OCTL at bit level, the user can modify only one or several bits in a single atomic AHB write access by programming '1' to the bit operate register (GPIOx_BOP, or for clearing only GPIOx_BC, or for toggle only GPIOx_TG). The other bits will not be affected.

8.3.2. External interrupt/event lines

All ports have external interrupt capability. To use external interrupt lines, the port must be configured as input mode.

8.3.3. Alternate functions (AF)

When the port is configured as AFIO (set CTLy bits to "0b10", which is in GPIOx_CTL registers), the port is used as peripheral alternate functions. Each port has sixteen alternate functions can be configured by GPIO alternate functions selected registers (GPIOx_AFSELy (y = 0,1)). The detail alternate function assignments for each port are in the device datasheet.

8.3.4. Additional functions

Some pins have additional functions, which have priority over the configuration in the standard GPIO registers. When for ADC , DAC, CMP or additional functions, the port must be configured as analog mode. When for RTC, WKUPx and oscillators additional functions, the port type is set automatically by related RTC, PMU and RCU registers. These ports can be used as normal GPIO when the additional functions disabled.

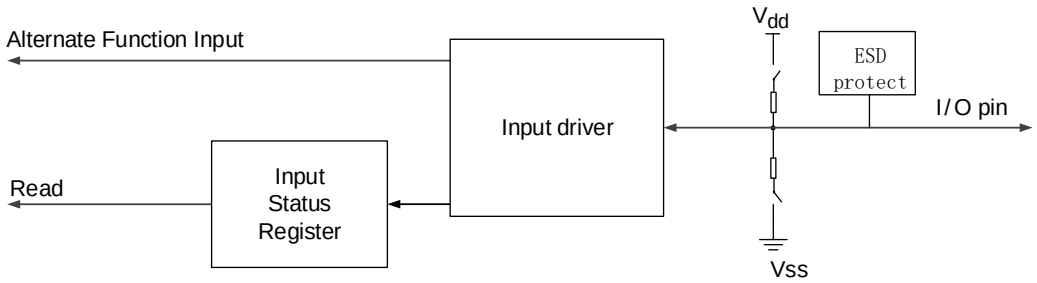
8.3.5. Input configuration

When GPIO pin is configured as input:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- Every AHB clock cycle the data present on the I/O pin is got to the port input status register.
- The output buffer is disabled.

Figure 8-2. Basic structure of Input configuration shows the input configuration.

Figure 8-2. Basic structure of Input configuration



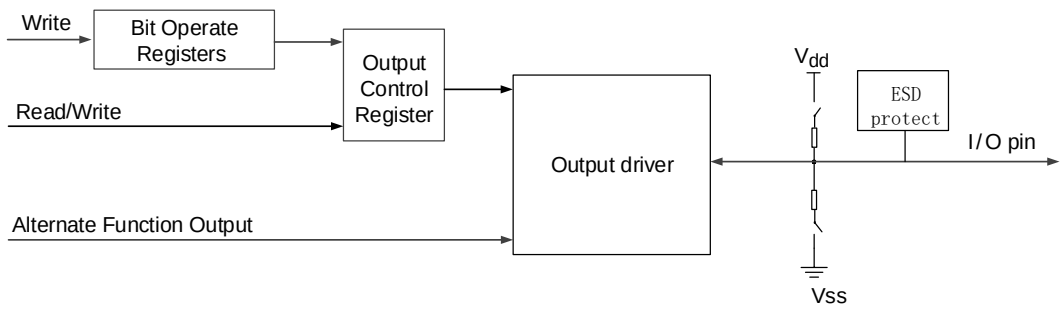
8.3.6. Output configuration

When GPIO pin is configured as output:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- The output buffer is enabled.
- Open Drain Mode: The pad output low level when a “0” in the output control register; while the pad leaves Hi-Z when a “1” in the output control register.
- Push-Pull Mode: The pad output low level when a “0” in the output control register; while the pad output high level when a “1” in the output control register.
- A read access to the port output control register gets the last written value.
- A read access to the port input status register gets the I/O state.

Figure 8-3. Basic structure of Output configuration shows the output configuration.

Figure 8-3. Basic structure of Output configuration



8.3.7. Analog configuration

When GPIO pin is used as analog configuration:

- The weak pull-up and pull-down resistors are disabled.
- The output buffer is disabled.
- The schmitt trigger input is disabled.
- The port input status register of this I/O port bit is “0”.

Figure 8-4. Basic structure of Analog configuration shows the analog configuration.

Figure 8-4. Basic structure of Analog configuration



8.3.8. Alternate function (AF) configuration

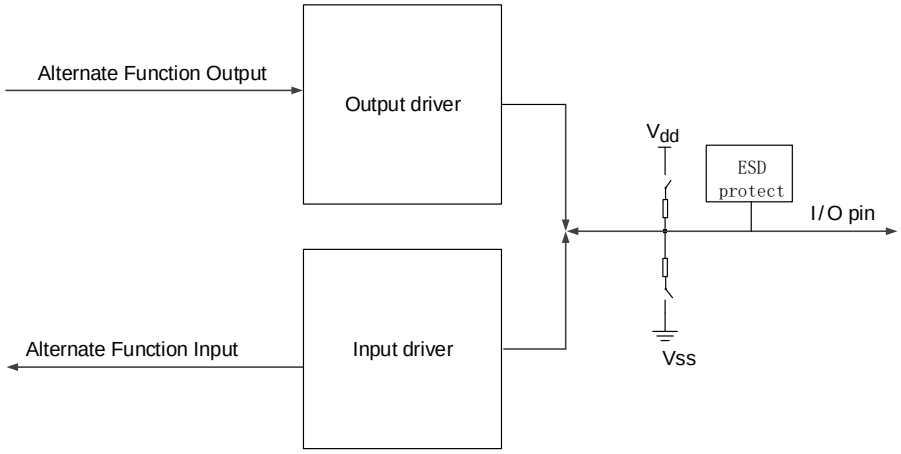
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

When be configured as alternate function:

- The output buffer is enabled in open-drain or push-pull configuration.
- The output buffer is driven by the peripheral.
- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- The I/O pin data is stored into the port input status register every AHB clock.
- A read access to the port input status register gets the I/O state.
- A read access to the port output control register gets the last written value.

Figure 8-5. Basic structure of Alternate function configuration shows the alternate function configuration.

Figure 8-5. Basic structure of Alternate function configuration



8.3.9. GPIO locking function

The locking mechanism allows the IO configuration to be protected.

The protected registers are GPIOx_CTL, GPIOx_OMODE, GPIOx_OSPD, GPIOx_PUD and GPIOx_AFSELy (y=0, 1). It allows the I/O configuration to be frozen by the 32-bit locking register (GPIOx_LOCK). When the special LOCK sequence has occurred on LKK bit in GPIOx_LOCK register and the LKy bit is set in GPIOx_LOCK register, the corresponding port is locked and the corresponding port configuration cannot be modified until the next reset. It recommended to be used in the configuration of driving a power module.

8.3.10. GPIO single cycle toggle function

GPIO could toggle the I/O output level in single AHB cycle by writing 1 to the corresponding bit of GPIOx_TG register. The output signal frequency could up to the half of the AHB clock.

8.4. Register definition

GPIOA base address: 0x4800 0000

GPIOB base address: 0x4800 0400

GPIOC base address: 0x4800 0800

GPIOD base address: 0x4800 0C00

GPIOE base address: 0x4800 1000

GPIOF base address: 0x4800 1400

8.4.1. Port control register (GPIOx_CTL, x=A..F)

Address offset: 0x00

Reset value: 0x000A 8280 for port B; 0x0000 0000 for others.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]	CTL12[1:0]	CTL11[1:0]	CTL10[1:0]	CTL9[1:0]	CTL8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]	CTL4[1:0]	CTL3[1:0]	CTL2[1:0]	CTL1[1:0]	CTL0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	CTL15[1:0]	Pin 15 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
29:28	CTL14[1:0]	Pin 14 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
27:26	CTL13[1:0]	Pin 13 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
25:24	CTL12[1:0]	Pin 12 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
23:22	CTL11[1:0]	Pin 11 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
21:20	CTL10[1:0]	Pin 10 configuration bits

		These bits are set and cleared by software. Refer to CTL0[1:0] description
19:18	CTL9[1:0]	Pin 9 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
17:16	CTL8[1:0]	Pin 8 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
15:14	CTL7[1:0]	Pin 7 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
13:12	CTL6[1:0]	Pin 6 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
11:10	CTL5[1:0]	Pin 5 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
9:8	CTL4[1:0]	Pin 4 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
7:6	CTL3[1:0]	Pin 3 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
5:4	CTL2[1:0]	Pin 2 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
3:2	CTL1[1:0]	Pin 1 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
1:0	CTL0[1:0]	Pin 0 configuration bits These bits are set and cleared by software. 00: Input mode (reset value) 01: GPIO output mode 10: Alternate function mode 11: Analog mode

8.4.2. Port output mode register (GPIOx_OMODE, x=A..F)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	OM15	Pin 15 output mode bit These bits are set and cleared by software. Refer to OM0 description
14	OM14	Pin 14 output mode bit These bits are set and cleared by software. Refer to OM0 description
13	OM13	Pin 13 output mode bit These bits are set and cleared by software. Refer to OM0 description
12	OM12	Pin 12 output mode bit These bits are set and cleared by software. Refer to OM0 description
11	OM11	Pin 11 output mode bit These bits are set and cleared by software. Refer to OM0 description
10	OM10	Pin 10 output mode bit These bits are set and cleared by software. Refer to OM0 description
9	OM9	Pin 9 output mode bit These bits are set and cleared by software. Refer to OM0 description
8	OM8	Pin 8 output mode bit These bits are set and cleared by software. Refer to OM0 description
7	OM7	Pin 7 output mode bit These bits are set and cleared by software. Refer to OM0 description
6	OM6	Pin 6 output mode bit

		These bits are set and cleared by software. Refer to OM0 description
5	OM5	Pin 5 output mode bit These bits are set and cleared by software. Refer to OM0 description
4	OM4	Pin 4 output mode bit These bits are set and cleared by software. Refer to OM0 description
3	OM3	Pin 3 output mode bit These bits are set and cleared by software. Refer to OM0 description
2	OM2	Pin 2 output mode bit These bits are set and cleared by software. Refer to OM0 description
1	OM1	Pin 1 output mode bit These bits are set and cleared by software. Refer to OM0 description
0	OM0	Pin 0 output mode bit These bits are set and cleared by software. 0: Output push-pull mode (reset value) 1: Output open-drain mode

8.4.3. Port output speed register (GPIOx_OSPD, x=A..F)

Address offset: 0x08

Reset value: 0x000C 0000 for port B; 0x0000 0000 for others.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	**13 12**	**11 10**	**9 8**	**7 6**	**5 4**	**3 2**	**1 0**
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	OSPD15[1:0]	Pin 15 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
29:28	OSPD14[1:0]	Pin 14 output max speed bits These bits are set and cleared by software.

		Refer to OSPD0[1:0] description
27:26	OSPD13[1:0]	Pin 13 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
25:24	OSPD12[1:0]	Pin 12 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
23:22	OSPD11[1:0]	Pin 11 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
21:20	OSPD10[1:0]	Pin 10 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
19:18	OSPD9[1:0]	Pin 9 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
17:16	OSPD8[1:0]	Pin 8 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
15:14	OSPD7[1:0]	Pin 7 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
13:12	OSPD6[1:0]	Pin 6 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
11:10	OSPD5[1:0]	Pin 5 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
9:8	OSPD4[1:0]	Pin 4 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
7:6	OSPD3[1:0]	Pin 3 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
5:4	OSPD2[1:0]	Pin 2 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
3:2	OSPD1[1:0]	Pin 1 output max speed bits

		These bits are set and cleared by software. Refer to OSPD0[1:0] description
1:0	OSPD0[1:0]	Pin 0 output max speed bits These bits are set and cleared by software. x0: Output max speed 2M (reset value) 01: Output max speed 10M 11: Output max speed 50M

8.4.4. Port pull-up/down register (GPIOx_PUD, x=A..F)

Address offset: 0x0C

Reset value: 0x0006 4040 for port B; 0x0000 0000 for others.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PUD7[1:0]	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	PUD15[1:0]	Pin 15 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
29:28	PUD14[1:0]	Pin 14 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
27:26	PUD13[1:0]	Pin 13 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
25:24	PUD12[1:0]	Pin 12 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
23:22	PUD11[1:0]	Pin 11 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
21:20	PUD10[1:0]	Pin 10 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
19:18	PUD9[1:0]	Pin 9 pull-up or pull-down bits

		These bits are set and cleared by software. Refer to PUD0[1:0] description
17:16	PUD8[1:0]	Pin 8 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
15:14	PUD7[1:0]	Pin 7 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
13:12	PUD6[1:0]	Pin 6 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
11:10	PUD5[1:0]	Pin 5 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
9:8	PUD4[1:0]	Pin 4 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
7:6	PUD3[1:0]	Pin 3 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
5:4	PUD2[1:0]	Pin 2 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
3:2	PUD1[1:0]	Pin 1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
1:0	PUD0[1:0]	Pin 0 pull-up or pull-down bits These bits are set and cleared by software. 00: Floating mode, no pull-up and pull-down (reset value) 01: With pull-up mode 10: With pull-down mode 11: Reserved

8.4.5. Port input status register (GPIOx_ISTAT, x=A..F)

Address offset: 0x10

Reset value: 0x0000 XXXX

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	ISTATy	Port input status (y=0..15) 0: Input signal low 1: Input signal high

8.4.6. Port output control register (GPIOx_OCTL, x=A..F)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	OCTLy	Port output control (y=0..15) These bits are set and cleared by software. 0: Pin output low 1: Pin output high

8.4.7. Port bit operate register (GPIOx_BOP, x=A..F)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	Cry	Port clear bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Clear the corresponding OCTLy bit
15:0	BOPy	Port set bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Set the corresponding OCTLy bit

8.4.8. Port configuration lock register (GPIOx_LOCK, x=A..F)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	LKK	Lock key It can only be set by using the lock key writing sequence. And is always readable. 0: GPIOx_LOCK register and the port configuration are not locked 1: GPIOx_LOCK register is locked until an MCU reset LOCK key writing sequence: Write 1→Write 0→Write 1→ Read 0→ Read 1 **Note:** The value of LKy(y=0..15) must be held during the LOCK Key writing sequence.
15:0	LKy	Port lock bit y(y=0..15) These bits are set and cleared by software. 0: Port configuration not locked 1: Port configuration locked

8.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A..F)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	SEL7[3:0]	Pin 7 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
27:24	SEL6[3:0]	Pin 6 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
23:20	SEL5[3:0]	Pin 5 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
19:16	SEL4[3:0]	Pin 4 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
15:12	SEL3[3:0]	Pin 3 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
11:8	SEL2[3:0]	Pin 2 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
7:4	SEL1[3:0]	Pin 1 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
3:0	SEL0[3:0]	Pin 0 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected

0111: AF7 selected

1000: AF8 selected

1001: AF9 selected

1010 ~ 1111: Reserved

8.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A..F)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	SEL15[3:0]	Pin 15 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
27:24	SEL14[3:0]	Pin 14 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
23:20	SEL13[3:0]	Pin 13 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
19:16	SEL12[3:0]	Pin 12 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
15:12	SEL11[3:0]	Pin 11 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
11:8	SEL10[3:0]	Pin 10 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
7:4	SEL9[3:0]	Pin 9 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description

3:0	SEL8[3:0]	Pin 8 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected 0111: AF7 selected 1000: AF8 selected 1001: AF9 selected 1010 ~ 1111: Reserved

8.4.11. Bit clear register (GPIOx_BC, x=A..F)

Address offset: 0x28

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CRy	Port clear bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Clear the corresponding OCTLy bit

8.4.12. Port bit toggle register (GPIOx_TG, x=A..F)

Address offset: 0x2C

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	TGy	Port toggle bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Toggle the corresponding OCTLy bit

9. Multi-function communication Interface (MFCOM)

9.1. Overview

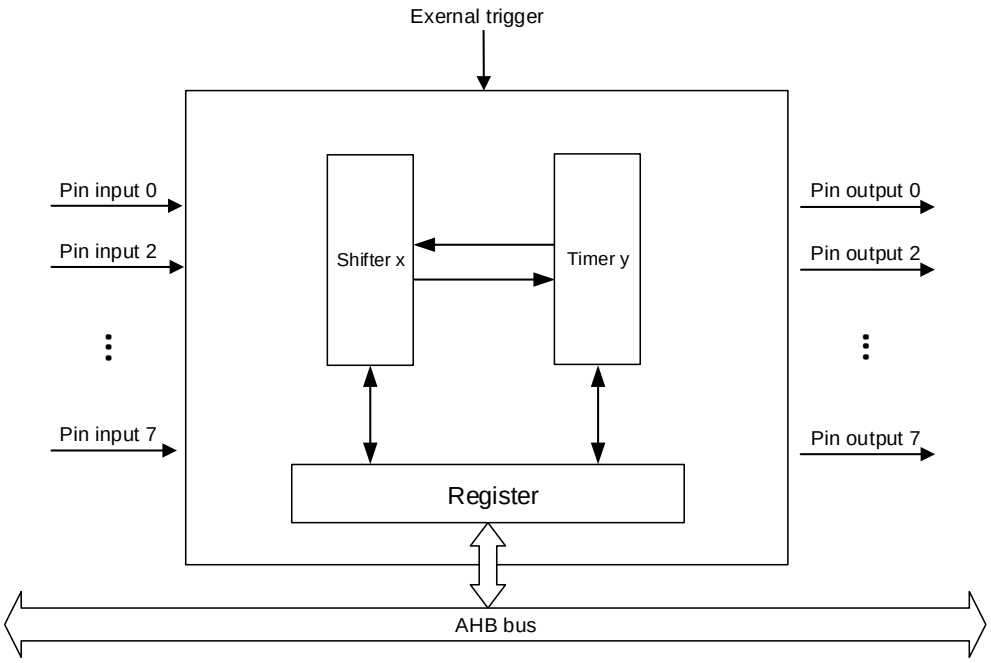
The MFCOM is a highly configurable module provide emulation of a variety of serial communication protocols and flexible timers.

9.2. Characteristics

- Continuous data transfer configuration
- Shift register with dual cache regions supports continuous data transfer.
- Automatic start/stop control
- DMA, Interrupt or polled transmit/receive control
- Supports various internal or external triggers reset, enable and disable modes of highly flexible 16-bit timer
- Baud rate programming from the HCLK reprogramming
- Combine different configurations of pin/shifter/timer to support PWM waveform

9.3. Block diagram

Figure 9-1. MFCOM block diagram



Access the MFCOM register through the AHB bus clock, MFCOM has 4 timers, 4 shifters and 8 pins. Can select the shifter's pin input and output through SPSEL[2:0] bit. By configuring INSRC, can select the input source of the shifte (the output of the shifter or the input of the

pin are optional).

Events such as data loads, stores can be signaled by using DMA/polling/interrupt methods. The trigger source (external trigger, pin or shifter status flag) is selected by configuring the timer, the timer generates the shift clock for the transmitted data, and the shifter can select different pins to input/output data.

9.4. Function overview

9.4.1. Clocking and resets

MFCOM clock

The MFCOMEN bit in RCU must be enabled before accessing any MFCOM registers. The clock of MFCOM is configured by RCU. The AHB clock is selected.

MFCOM reset

MFCOM allows software reset through set the SWRSTEN bit in the MFCOM_CTL. Software reset all registers to default state, except for the MFCOM_CTL. System reset will reset all logic and registers of the MFCOM.

MFCOM modes

MFCOM module supported normal mode and debug mode. If the MFCOM_HOLD bit in register DBG_CTL0 of DBG module is set, the MFCOM continues to work even the Cortex®-M33 core halted. While the MFOCM_HOLD bit is cleared, the MFCOM stops in debug mode.

9.4.2. Shifter

Shifter is used to cache and transfer MFCOM data. The TMSEL of MFCOM_SCTLx register is configured so that the timing of shift, load, and store events is controlled by the timer assigned to the shifter. When in transmit mode, if the stop bit is enabled, the shifter inserts the stop bit immediately when the shifter is initially configured. In receive mode, configure SSTART, TMSTART, or SSTOP, TMSTOP in MFCOM_SCFG and MFCOM_TCFG register check the start/stop bit can be enabled before/after the shift data.

Figure 9-2. Shifter microarchitecture

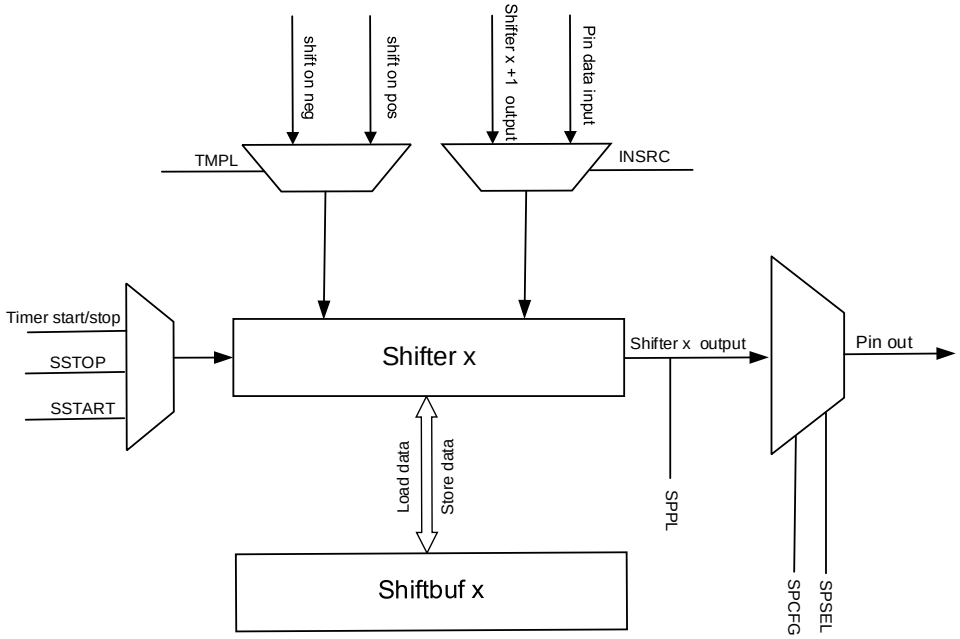


Table 9-1. Mode of shifter

Mode	Shifter working mechanism	SSTAT、Interrupt DMA request setting conditions	SERR、Interrupt setting conditions
Transmit	the shifter will load data from the shifter buffer and shift data out when a load event is signalled by the assigned timer.	data has been loaded from the shifter buffer into the shifter or when the Shifter is initially configured into transmit mode.	An attempt to load data from an empty shifter buffer occurs (buffer underrun).
Receive	the shifter will shift data in and store data into the shifter buffer when a store event is signalled by the assigned timer.	data has been stored into the shifter buffer from the shifter.	An attempt to store data into a full shifter buffer occurs (buffer overrun) or when a mismatch occurs on a start/stop bit check.

9.4.3. Timer

The shift registers loading, shifting and storing are controlled by the MFCOM timers, the counters load the contents of the compare register and decrement down to zero on MFCOM_CLK. Timers can be configured to enable or disable triggers generating a clock or PWM waveform, start condition and stop condition.

Timers operates independently, but the timer output can be used to trigger any other timer,

and also can configure a timer to enable or disable other timers. Timer output, pin input, shifter status flag or external trigger input can be configured independently by each timer. Before setting the configuration of the timer, the timer configuration register (MFCOM_TMCFG) should be configured. Trigger configuration is a pin independent configuration that can be configured for input/output data, or output enable.

8-bit baud counter configuration

The 16-bit counter is divided into two 8-bit counters, lower 8-bits are used to configure the baud rate of the shift clock and when the lower 8-bits decrement to zero, the timer output is toggled and the lower 8-bits reload from the compare register. The upper 8-bits are used to configure the number of shift clock edges in the transfer when the lower 8-bits equal zero and decrement.

In 8-bit baud counter configuration timer reset events only reset the lower 8-bit counter, not the upper 8-bit counter, and can decrement if timer reset is configured to update the state of timer output, which is switched due to the timer reset event. When the upper 8 bits are equal to zero and decrement the timer compare event occurs, and the timer status flag is set.

Note: when field TMMOD[1:0] configured as 0b01,the field TMDEC[1:0] can only be set as 0b00,0b01.

16-bit counter configuration

The number of shift clock edges in the transmission or the baud rate of the shift clock can be configured by a 16-bit counter. When the 16-bit counter equal to zero and decrement, the timer output switches and the counter is reloaded from the comparison register. When the 16-bit counter equal to zero and decrement, the timer comparison event occurs and the timer status flag is set.

8-bit high pwm configuration

The 16-bit counter is divided into two 8-bit counters. Low 8 bits are used to configure the timer outputs high cycle and the high 8-bits are used to configure the timer output low cycle. When the lower 8 bits are decrement to zero, the timer output is cleared and the lower 8 bits are reloaded from the comparison register. The high 8-bits decrement when output is low. And when decrement to zero, the timer output is set, and the high 8-bits will reloaded from the compare register. When the upper 8-bits equal to zero and decrement and field TMDEC[1:0] is set as 0b00 or 0b01, a timer comparison event occurs and timer status flag is set. If the bit field TMMOD[1:0] is configured in PWM configuration, the bit field TMSTART/TMSTOP[1:0]/TMOUT[1:0] are not supported.

Timer enable and start bit

When the TMMOD[1:0] in MFCOM_TMCTL register is the required configuration, and the condition is detected by the timer configuration (TMEN[2:0]) then the following events occur.

1. The timer counter will load the current value of the comparison register and begin to

decrement according to the TMDEC[1:0] configuration.

2. Based on the TMOUT[1:0] configuration, the timer output is updated to its initial state. The shifter controlled by this timer will not see this as the rising edge on the timer shift clock.
3. As configured by SSTART[1:0], load the shift register from the shift buffer and output the first bit, or the timer-controlled transfer shifter will output the starting bit value.

When the timer start bit is enabled, the timer will reload the compare register on the first rising edge of the shifter clock. On the shift clock if there is no falling edge before the first rising edge (TMOUT=1), shifter configured to shift on falling edge and will not load correctly on the first shift.

Timer decrement and reset

According to the TMMOD[1:0] and TMDEC[1:0] fields, the timer generates the timer output and the timer shift clock. The shifter clock is either equal to the timer output or equal to the decrement clock. When TMDEC[1:0] is configured to decrement from a pin or trigger, the timer will decrement on both rising and falling edges.

In the TMRST[2:0] field the timer is configured to reset then the timer counter will load the current value of the compare register. The timer shift clock and timer output can be configured to update on timer reset by TMMOD[1:0]. This can result in a timer shift clock edge if the timer output toggles as a result of the timer reset. In 8-bit baud counter configuration, high 8-bit counter decrement only occurs when the low 8-bits equal to zero and decrement.

A timer comparison event is triggered when the timer counter decrements to 0. The trigger of the timer comparison event will cause the timer counter to load the contents of the comparison register, the timer output to the toggle, the send shift register of any configuration to load, and the receive shift register of any configuration to store. Depending on the timer configuration, can set the timer status flag.

Timer disable and stop bit

The timer is configured to add a stop bit to each compare, and the following additional events will occur. When the timer is configured to insert a stop bit on each compare transmit shifters must be configured to load on the first shift.

1. Configure SSTOP[1:0] can controlled Transmit shifters will output their stop bit.
2. Receive shifters controlled by this timer will store the data of the shift register in their shift buffer, as configured by SSTOP[1:0].
3. On the first rising edge of the shifter clock after the compare, the timer counter will reload the current value of the compare register.

When detect the condition configured by timer disable (TMDIS[2:0]), the following events will occur.

1. The timer counter will load the current value from the compare register, decrement according to the configuration of TMDEC[1:0].
2. The timer output clears shifters controlled by the timer, which do not treat it as a falling

edge on the timer shift clock, but can generate a shift event if the timer shift clock will generate a shift event.

3. The transmit shifters controlled by this timer will output their stop values.
4. The receiving shifter controlled by this timer will store the contents of the shift register and the shift buffer configured by SSTOP

The timer counter continues to decrease and, if the timer stop bit is enabled, does not end until the next rising edge of the shifted clock is detected. The timer shift clock can be switched during the stop bit without generating a shift event. The timer output is forced down during the stop bit.

In the same cycle as a timer disable condition (stop bit is disabled) timer enable condition can be detected, or on the first rising edge of the shift clock after the disable condition (stop bit is enabled). Receive shift registers with stop bit enabled will store the contents of the shift register into the shift buffer and verify the state of the input data on the configured shift edge while the timer is in the stop state condition. If the un-configured edge is between the disabled timer and the next rising edge of the shift clock, the final storage and validation will not occur.

9.4.4. Pin

Each timer and shifter can be configured for input, output data, output enabled or bidirectional output. Pins configured to output enable can be used as an open drain or to control the enable on a bidirectional output. Any timer or shifter may be driven by another timer or shifter, configured to control output so that data can be output bidirectional on one pin.

Pin synchronization

When a pin is configured as input, the input signal is first synchronized with the MFCOM clock before the signal is used by a timer or shifter. A small delay between 0.5 and 1.5 flexible clock cycles is introduced when the external pin input is used to generate the output or control the shifter. This sets the maximum setup time to 1.5 MFCOM clock cycles. If an input is used by more than one timer or shifter, synchronization occurs only once to ensure that all timers and shifters using the input see the edge at the same time.

MFCOM pins are interlinked, and configure a shifter or timer to output data on one pin will establish an internal connection that other shifters and timers can use as input. Means that the shifter output can be used to trigger a timer or output a timer to a shifter. This process is synchronized with the MFCOM clock, so there is a 1 cycle delay.

So when using a pin input as a timer trigger, timer clock, or shifter data input, a synchronization delay occurs, external pin 0.5 to 1.5 MFCOM clock cycles and Internal drive pin for one MFCOM clock cycle.

9.4.5. Interrupts and DMA requests

MFCOM interrupt can be generated by shift status flag, shift error flag and timer status flag. The specific interrupt event description and DMA request are shown in ***Table 9-2. MFCOM***

interrupts and DMA requests.

Table 9-2. MFCOM interrupts and DMA requests

Interrupts events	description	Interrupt enable bit	DMA requests
SSTAT	Shifter Status Flag	SSIEN	Y
SERR	Shifter Error Flag	SEIEN	N
TMSTAT	Timer Status Flag	TMSIEN	N

9.4.6. Triggers

Peripheral triggers

The connection of the MFCOM peripheral trigger to other peripherals is specific.

Input trigger

Multiple external trigger inputs are supported, the external trigger is synchronized with the MFCOM_CLK, and at least two clock cycles must be asserted to be properly sampled. One or more MFCOM timers can be triggered.

Output triggers

The output trigger of each MFCOM timer is equal to the timer output, and the timer output is not affected by the timer's pin polarity configuration.

9.5. Register definition

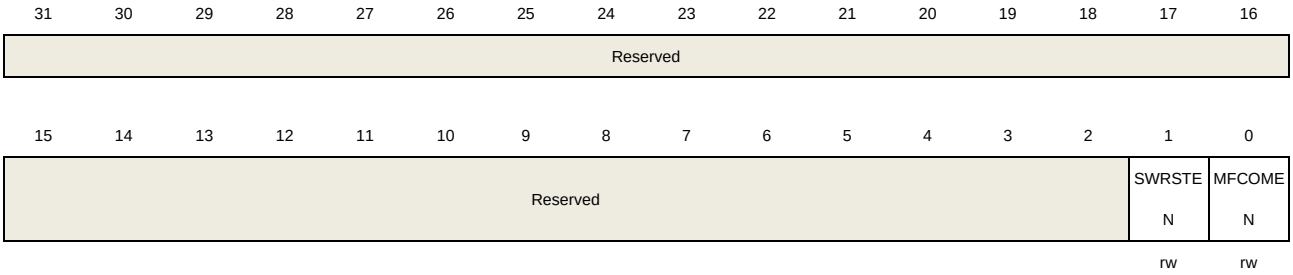
MFCOM base address: 0x4900 3400

9.5.1. Control register (MFCOM_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



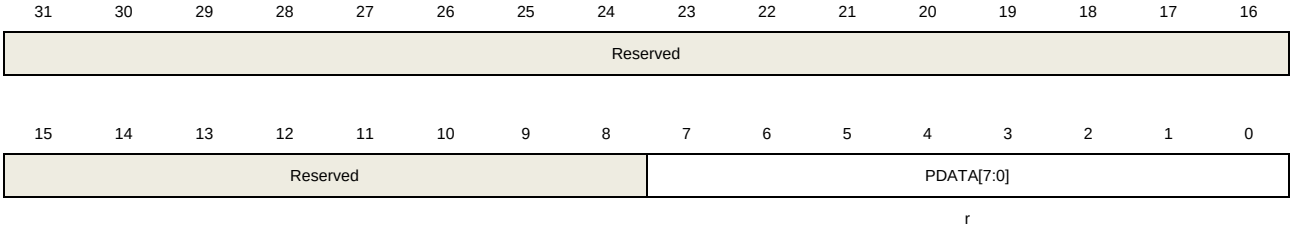
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	SWRSTEN	Software reset enable register accesses are ignored except the control register until this bit is cleared. 0: Disable software reset 1: Enable software reset, all MFCOM registers except the control register are reset.
0	MFCOMEN	MFCOM enable 0: Disable MFCOM module. 1: Enable MFCOM module.

9.5.2. Pin data register (MFCOM_PINDATA)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value

7:0	PDATA[7:0]	data of pins The input/output data of each MFCOM pins.

9.5.3. Shifter status register (MFCOM_SSTAT)

Address offset: 0x08
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SSTAT[3:0]			
												rc_w1			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	SSTAT[3:0]	Shifter x status flag The shifter status flag is set when one of the following events occurs: SMOD = Receive, the status flag is set when MFCOM_SBUF has loaded data from the shifter, and cleared when the MFCOM_SBUF register is read. SMOD = Transmit, the status flag is set when MFCOM_SBUF data is transferred to the shifter or initially configured for this mode, and cleared when data is written to the MFCOM_SBUF register. 0: Shifter x status flag is not set 1: Shifter x status flag is set.

9.5.4. Shifter error register (MFCOM_SERR)

Address offset: 0x0C
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SERR[3:0]			
												rc_w1			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	SERR[3:0]	Shifter x error flags

The shift error flag is set when one of the following events occurs:
SMOD = Receive, MFCOM_SBUF overrun, or the receive start or stop bit does not match the expected value.
SMOD = Send, MFCOM_SBUF underrun.
This bit can be cleared by software writing 1.
0: Shifter x error flag is not set.
1: Shifter x error flag is set.

9.5.5. Timer status register (MFCOM_TMSTAT)

Address offset: 0x10
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TMSTAT[3:0]			
												rc_w1			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	TMSTAT[3:0]	Timer x status flags Depending on the timer configuration, the timer status flag is set when the following events occur. In 8-bit baud counter configuration, the timer status flag will be set when the upper 8-bit counter down to zero and the decrement is active. In 8-bit high PWM configuration, the timer status flag will be set when the upper 8-bit counter down to zero and the decrement is active. In 16-bit counter configuration, the timer status flag will be set when the 16-bit counter decrements to zero and the decrement is active. This bit can be cleared by software writing 1. 0: Timer x status flag is not set. 1: Timer x status flag is set.

9.5.6. Shifter status interrupt enable register (MFCOM_SSIEN)

Address offset: 0x18
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SSIEN[3:0]			
												rw			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	SSIEN[3:0]	Shifter status interrupt enable Enable interrupt when the shifter x status flags in bit field SSTAT[3:0] are set. 0: Shifter status flags do not generate interrupts 1: Shifter status flags generate interrupts

9.5.7. Shifter error interrupt enable register (MFCOM_SEIEN)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SEIEN[3:0]			
												rw			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	SEIEN[3:0]	Shifter error interrupt enable Enable interrupt when the shifter x error flag bits in bit field SERR[3:0] are set 0: Shifter error flags do not generate an interrupt 1: Shifter error flags generate an interrupt

9.5.8. Timer status interrupt enable register (MFCOM_TMSIEN)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TMSIEN[3:0]			
												rw			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	TMSIEN[3:0]	Timer status interrupt enable Enable interrupt when timer x status flags in bit field TMSTAT[3:0] are set. 0: Timer status flags do not generate interrupts 1: Timer status flags generate interrupts

9.5.9. Shifter status DMA enable register (MFCOM_SSDMAEN)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SSDMAEN[3:0]			
												rw			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	SSDMAEN[3:0]	Shifter status DMA enable DMA is enabled when the timer x status flags in bit field SSTAT[3:0] are set 0: Shifter status flags do not generate DMA requests 1: Shifter status flags generate DMA requests

9.5.10. Shifter control x register (MFCOM_SCTLx)

Address offset: 0x80 + 0x04 * x, (x = 0 to 3)

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TMSEL[1:0]		TMPL	Reserved					SPCFG[1:0]	
						rw		rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SPSEL[2:0]			SPPL	Reserved				SMOD[2:0]		
					rw			rw					rw		

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.

25:24	TMSEL[1:0]	Timer select Selects the timer used to generate the shift clock and control the shift logic. 00：Select timer 0 01：Select timer 1 10：Select timer 2 11：Select timer 3
23	TMPL	Timer polarity 0: Shift on rising edge of shift clock 1: Shift on falling edge of shift clock
22:18	Reserved	Must be kept at reset value.
17:16	SPCFG[1:0]	Shifter pin configuration 00: Shifter pin input 01: Shifter pin open drain 10: Shifter cascade pin input/output data 11: Shifter pin output
15:11	Reserved	Must be kept at reset value.
10:8	SPSEL[2:0]	Shifter pin select Select the pin to use for the shifter input or output.
7	SPPL	Shifter pin polarity 0: Pin active high 1: Pin active low
6:3	Reserved	Must be kept at reset value.
2:0	SMOD[2:0]	Shifter mode Configures the mode of the shifter. 000: Disabled. 001: Receive mode. 010: Transmit mode. 011: Reserved. 100: Reserved. 101: Reserved. 110: Reserved. 111: Reserved.

9.5.11. Shifter configuration x register (MFCOM_SCFGx)

Address offset: 0x100 + 0x004 * x, (x = 0 to 3)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							INSRC	Reserved		SSTOP[1:0]		Reserved		SSTART[1:0]	
							rw			rw				rw	

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	INSRC	Input source Selects the input source for the shifter. 0: Pin 1: Output of shifter x+1(x<3)
7:6	Reserved	Must be kept at reset value.
5:4	SSTOP[1:0]	Shifter stop bit 00: Disable stop bit 01: Reserved 10: In transmit mode, the stop bit is valid at low level. In receive mode, the stop bit is not low level 11: In transmit mode, stop bit high is valid. In receive mode, stop bit is not high. Error indicates set bit **Note:** In transmit mode, if the selected timer has enabled stop bits, the data frame allows automatic insertion of stop bits. In receive mode, if the selected timer has the stop bit enabled, the data frame allows automatic verification of the stop bit.
3:2	Reserved	Must be kept at reset value.
1:0	SSTART[1:0]	Shifter start bit 00: Disable the start bit, send data when the start bit is enabled 01: Disable start bit, send data on first shift 10: The start bit is valid at low level before the first shift to send data in transmit mode. If the start bit is not low level in receive mode, an error flag is set 11: The start bit high level is valid before the first shift to send data in the transmit mode. If the start bit is not high level in the receive or match storage mode, the error indicates the set bit **Note:** In transmit mode, if the selected timer has enabled the start bit, the data frame allows the start bit to be automatically inserted. In receive mode, if the selected timer has the start bit enabled, the data frame allows automatic check of the start bit.

9.5.12. Shifter buffer x register (MFCOM_SBUFx)

Address offset: 0x200 + 0x004 * x, (x = 0 to 3)

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBUF[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBUF[15:0]															
rw															

Bits	Fields	Descriptions
31:0	SBUF[31:0]	Shift buffer Based on SMOD settings, shift buffer data can be used for the following functions: SMOD = Receive, shifter data is stored into MFCOM_SBUF at the trigger event of timer. SMOD = Transmit, SHIFTBUF data is loaded into the shifter before the timer begins.

9.5.13. Shifter buffer x bit swapped register (MFCOM_SBUFBISx)

Address offset: 0x280 + 0x004 * x, (x = 0 to 3)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBUFBIS[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBUFBIS[15:0]															
rw															

Bits	Fields	Descriptions
31:0	SBUFBIS[31:0]	Shift buffer bit swapped Same as the MFCOM_SBUF register, except that the read/write register is bit swapped, and reads return SBUF[0:31].

9.5.14. Shifter buffer x byte swapped register (MFCOM_SBUFBYSx)

Address offset: 0x300 + 0x004 * x, (x = 0 to 3)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBUFBYS[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBUFBYS[15:0]															
rw															

Bits	Fields	Descriptions
31:0	SBUFBYS[31:0]	Shift buffer byte swapped Same as the MFCOM_SBUF register, except that the read/write register is byte swapped, and reads return {SBUF[7:0], SBUF[15:8], SBUF[23:16], SBUF[31:24]}.

9.5.15. Shifter buffer x bit byte swapped register (MFCOM_SBUFBBSx)

Address offset: 0x380 + 0x004 * x, (x = 0 to 3)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBUFBBS[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBUFBBS[15:0]															
rw															

Bits	Fields	Descriptions
31:0	SBUFBBS[31:0]	Shift buffer bit byte swapped Same as the MFCOM_SBUF register, except that the read/write register is bit swapped within each byte, and reads return {SBUF[24:31], SBUF[16:23], SBUF[8:15], SBUF[0:7]}.

9.5.16. Timer control x register (MFCOM_TMCTLx)

Address offset: 0x400 + 0x004 * x, (x = 0 to 3)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TRIGSEL[3:0]				TRIGPL	TRIGSRC	Reserved				TMPCFG[1:0]	
				rw				rw	rw					rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TMPSEL[2:0]			TMPPL	Reserved					TMMOD[1:0]	
					rw			rw						rw	

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27:24	TRIGSEL[3:0]	Trigger select

		Select external trigger (TRIGSRC = 0): 0001: external trigger 0 input 0010: external trigger 1 input 0100: external trigger 2 input 1000: external trigger 3 input Select internal trigger (TRIGSRC = 1): 0000: Pin 0 0001: Shifter 0 flag 0010: Pin 1 0011: Timer 0 trigger 0100: Pin 2 0101: Shifter 1 flag 0110: Pin 3 0111: Timer 1 trigger 1000: Pin4 1001: Shifter 2 flag 1010: Pin5 1011: Timer 2 trigger 1100: Pin6 1101: Shifter 3 flag 1110: Pin7 1111: Timer 2 trigger
23	TRIGPL	Trigger polarity 0: Trigger is activated at high 1: Trigger is activated at low
22	TRIGSRC	Trigger source 0: Select external trigger 1: Select internal trigger
21:18	Reserved	Must be kept at reset value
17:16	TMPCFG[1:0]	Timer pin configuration Configures the direction of the timer pin. 00: Timer pin input 01: Timer pin open drain 10: Timer cascade pin input/output 11: Timer pin output
15:11	Reserved	Must be kept at reset value
10:8	TMPSEL[2:0]	Timer Pin Select Select the pin to use for the timer input or output.
7	TMPPL	Timer Pin Polarity Configures pins as an output 0: Pin active high

		1: Pin active low
6:2	Reserved	Must be kept at reset value
1:0	TMMOD[1:0]	Timer configuration 00: Disable timer 01: Dual 8-bit counters baud configuration 10: Dual 8-bit counters PWM high configuration 11: Single 16-bit counter configuration

9.5.17. Timer configuration x register (MFCOM_TMCFGx)

Address offset: 0x480 + 0x004 * x, (x = 0 to 3)
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					TMOUT[1:0]			Reserved		TMDEC[1:0]		Reserved	TMRST[2:0]		
					rw					rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TMDIS[2:0]			Reserved	TMEN[2:0]			Reserved	TMSTOP[1:0]			Reserved	TMSTART		Reserved
	rw				rw				rw				rw		

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value
25:24	TMOUT[1:0]	Timer output Configures the initial state of the timer output and whether it is affected by the timer reset. 00: Enable and unaffected by timer reset when timer output is logic 1 01: Enable and unaffected by timer reset when timer output is logic 0 10: Timer output is logical 1 when enabled and timer reset 11: Timer output is logical 0 when enabled and timer reset
23:22	Reserved	Must be kept at reset value
21:20	TMDEC[1:0]	Timer decrement Configures the source of the timer decrement and the source of the shift clock. 00: Decrement counter on MFCOM clock, shift clock equal to timer output. 01: Decrement counter on trigger input, shift clock equal to timer output. 10: Decrement counter on pin input, shift clock equal to pin input. 11: Decrement counter on trigger input, shift clock equal to trigger input.
19	Reserved	Must be kept at reset value
18:16	TMRST[2:0]	Timer reset Configure the condition that causes the timer counter and the timer output to be reset.

		Note: In 8-bit counter configuration, the timer reset will only reset the lower 8-bits 000: Timer never reset 001: Reserved 010: Timer reset on timer pin equal to timer output 011: Timer reset on timer trigger equal to timer output 100: Timer reset on timer pin rising edge 101: Reserved 110: Timer reset on trigger rising edge 111: Timer reset on trigger rising or falling edge
15	Reserved	Must be kept at reset value
14:12	TMDIS[2:0]	Timer disable Configure conditions that can disable timers and stop decrement. 000: Never disabled 001: Disabled on timer x-1 disable 010: Disabled on timer compare 011: Disabled on timer compare and trigger low 100: Disabled on pin rising or falling edge 101: Disabled on pin rising or falling edge and provided trigger is high 110: Disabled on trigger falling edge 111: Reserved
11	Reserved	Must be kept at reset value
10:8	TMEN[2:0]	Timer enable Configure the conditions that enable the timer and start decrement 000: Always enabled 001: Enabled on timer x-1 enable 010: Enabled on trigger high 011: Enabled on trigger high and pin high 100: Enabled on pin rising edge 101: Enabled on pin rising edge and trigger high 110: Enabled on trigger rising edge 111: Enabled on trigger rising or falling edge
7:6	Reserved	Must be kept at reset value
5:4	TMSTOP[1:0]	Timer stop bit 00: Disable stop bit 01: Enable stop bit on timer compare 10: Enable stop bit on timer disable 11: Enable stop bit on timer compare and timer disable
3:2	Reserved	Must be kept at reset value
1	TMSTART	Timer start bit 0: Disable start bit

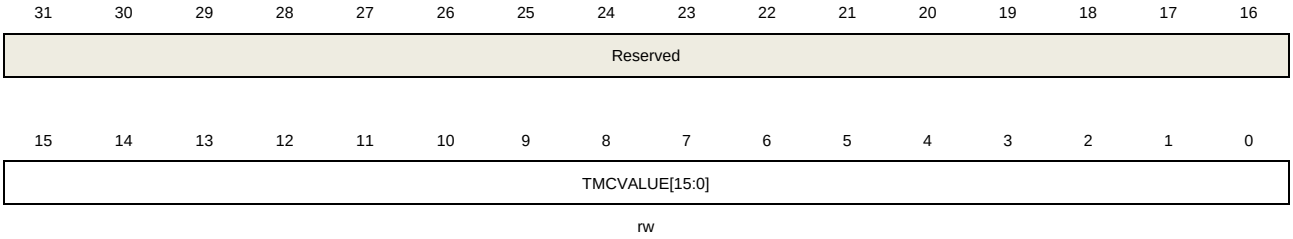
		1: Enabled start bit
0	Reserved	Must be kept at reset value

9.5.18. Timer compare x register (MFCOM_TMCMPx)

Address offset: 0x500 + 0x004 * x, (x = 0 to 3)

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	TMCVALUE[15:0]	Timer compare value The timer compare value is loaded into the timer counter when the timer is first enabled, the timer is reset, and the timer is reduced to zero. 8-bit baud counter configuration: lower 8-bits configure the baud rate divider = (TMCVALUE [7:0] + 1) * 2 upper 8-bits configure the number of bits in each word = (TMCVALUE [15:8] + 1) / 2. 8-bit PWM high configuration: lower 8-bits configure the high period of the output to (TMCVALUE [7:0] + 1) upper 8-bits configure the low period of the output to (TMCVALUE [15:8] + 1). 16-bit counter configuration: baud rate divider = (TMCVALUE [15:0] + 1) * 2. When the shift clock source is a pin or trigger input, the number of bits of each word = (TMCVALUE [15:0] + 1) / 2.

10. CRC calculation unit (CRC)

10.1. Overview

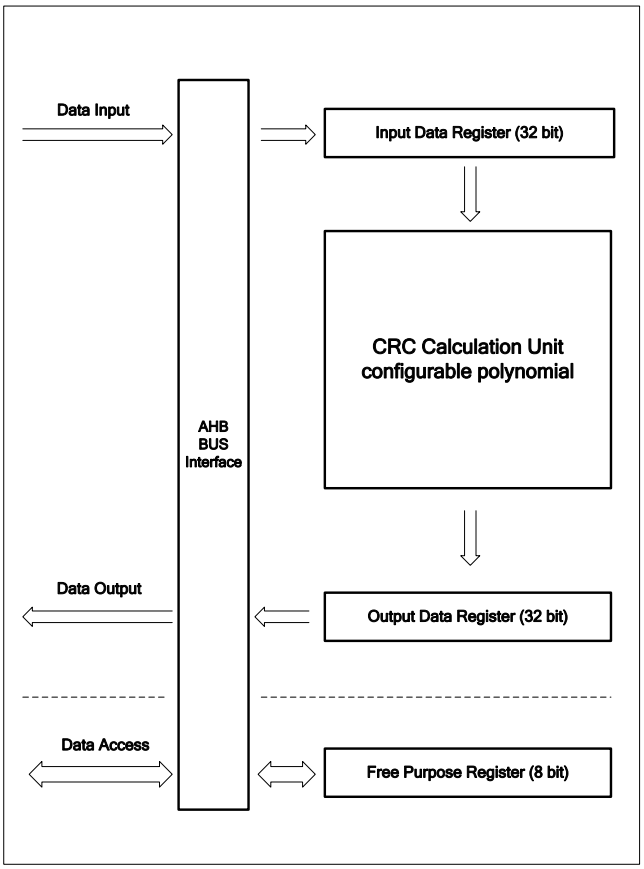
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

This CRC calculation unit can be used to calculate 7/8/16/32 bit CRC code within user configurable polynomial.

10.2. Characteristics

- Supports 7/8/16/32 bit data input
- For 7(8)/16/32 bit input data length, the calculation cycles are 1/2/4 AHB clock cycles
- User configurable polynomial value and size
- After CRC module·reset, user can configure initial value
- Free 8-bit register is unrelated to calculation and can be used for any other goals by any other peripheral devices

Figure 10-1. Block diagram of CRC calculation unit



10.3. Function overview

- CRC calculation unit is used to calculate the 32-bit raw data, and CRC_DATA register will receive the raw data and store the calculation result.

  If the CRC_DATA register has not been cleared by setting the CRC_CTL register, the new input raw data will be calculated based on the result of previous value of CRC_DATA.

  CRC calculation will spend 4/2/1 AHB clock cycles for 32/16/8(7) bit data size. During this period, AHB will not be hanged because of the existence of the 32bit input buffer.

- This module supplies an 8-bit free register CRC_FDATA.

  CRC_FDATA is unrelated to the CRC calculation. Independent read and write operations can be performed at any time.

- Reversible function can reverse the input data and output data.

  For input data, 3 reverse types can be selected.

  Original data is 0x1A2B3C4D:

  1) byte reverse:
  32-bit data is divided into 4 groups and reverse implement in group inside. Reversed data: 0x58D43CB2

  2) half-word reverse:
  32-bit data is divided into 2 groups and reverse implement in group inside. Reversed data: 0xD458B23C

  3) word reverse:
  32-bit data is divided into 1 groups and reverse implement in group inside. Reversed data: 0xB23CD458

  For output data, reverse type is word reverse.

  For example: when REV_O=1, calculation result 0x22CC4488 will be converted to 0x11223344.

- User configurable initial calculation data is available.

  When RST bit is set or write operation to CRC_IDATA register, the CRC_DATA register will be automatically initialized to the value in CRC_IDATA.

- User configurable polynomial.

  Depends on PS[1:0] bits, the valid polynomial and output bit width can be selected by user. If the polynomial is less than 32 bit, the high bits of the input data and output data is unavailable. It is strongly recommend resetting the CRC calculation unit after change the PS[1:0] bits or polynomial.

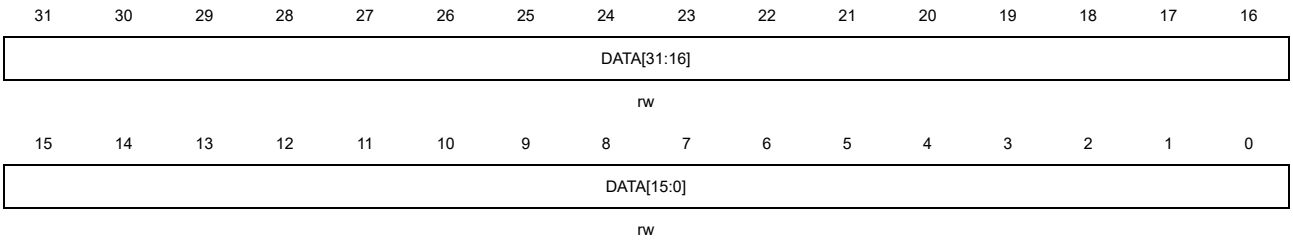
10.4. Register definition

CRC base address: 0x4002 3000

10.4.1. Data register (CRC_DATA)

Address offset: 0x00
Reset value: 0xFFFF FFFF

This register has to be accessed by word (32-bit).

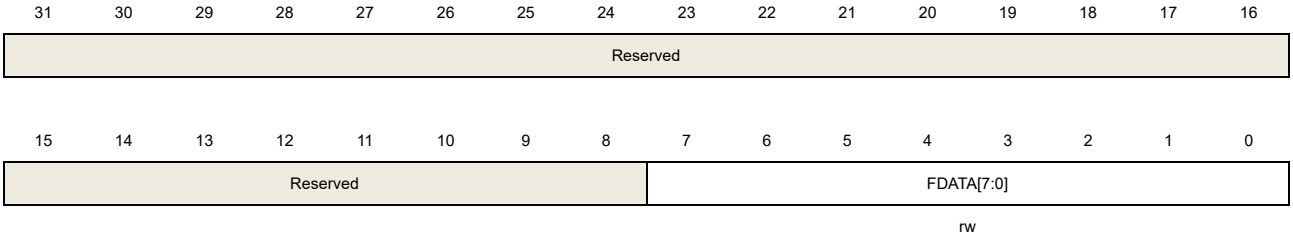


Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software writes and reads. This register is used to calculate new data, and the register can be written the new data directly. Write value cannot be read because the read value is the previous CRC calculation result.

10.4.2. Free data register (CRC_FDATA)

Address offset: 0x04
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



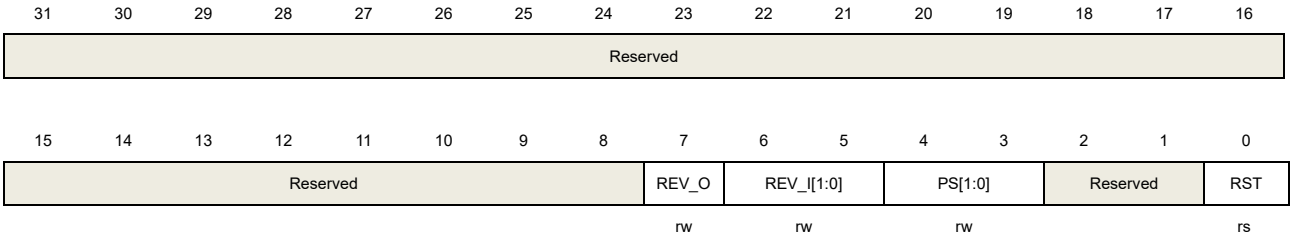
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free data register bits Software writes and reads. These bits are unrelated with CRC calculation. This byte can be used for any goal by any other peripheral. The CRC_CTL register will generate no effect to the byte.

10.4.3. Control register (CRC_CTL)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	REV_O	Reverse output data value in bit order 0: Not bit reversed for output data 1: Bit reversed for output data
6:5	REV_I[1:0]	Reverse type for input data 0: Dot not use reverse for input data 1: Reverse input data with every 8-bit length 2: Reverse input data with every 16-bit length 3: Reverse input data with whole 32-bit length
4:3	PS[1:0]	Size of polynomial 0: 32 bit 1: 16 bit (POLY [15:0] is used for calculation.) 2: 8 bit (POLY [7:0] is used for calculation.) 3: 7 bit (POLY [6:0] is used for calculation.)
2:1	Reserved	Must be kept at reset value.
0	RST	Software writes and reads. Set this bit can reset the CRC_DATA register. When set, the value of the CRC_DATA register is automatically initialized to the value in the CRC_IDATA register and then automatically cleared by hardware. This bit will take no effect to CRC_FDATA.

10.4.4. Initialization data register (CRC_IDATA)

Address offset: 0x10

Reset value: 0xFFFF FFFF

This register has to be accessed by word (32-bit).



IDATA[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDATA[15:0]															
rw															

Bits	Fields	Descriptions
31:0	IDATA[31:0]	Configurable initial CRC data value When RST bit in CRC_CTL asserted, CRC_DATA will be programmed to this value.

10.4.5. Polynomial register (CRC_POLY)

Address offset: 0x14

Reset value: 0x04C1 1DB7

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:0	POLY[31:0]	User configurable polynomial value This value is used together with PS[1:0] bits.

11. Direct memory access controller (DMA)

11.1. Overview

The direct memory access (DMA) controller provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, thereby freeing up bandwidth for other system functions. Data can be quickly moved by DMA between peripherals and memory as well as memory and memory without any CPU actions. There are 12 channels in the DMA controller(7 for DMA0 and 5 for DMA1). Each channel is dedicated to manage memory access requests from one or more peripherals. An arbiter is implemented inside to handle the priority among DMA requests.

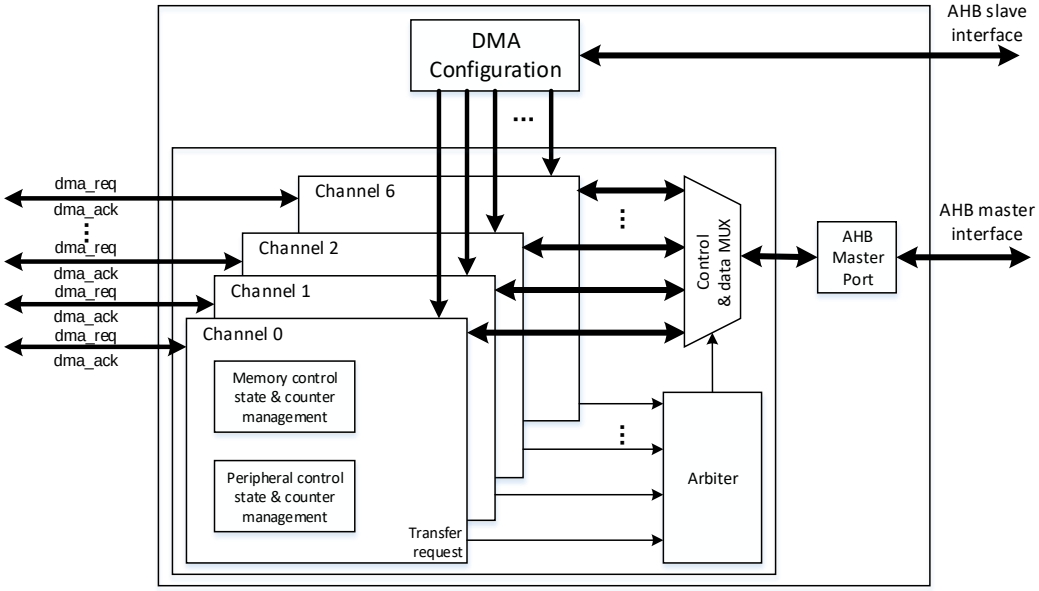
The system bus is shared by the DMA controller and the Cortex®-M33 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system bus bandwidth for the CPU.

11.2. Characteristics

- Programmable length of data to be transferred, max to 65536.
- 12 channels (7 for DMA0 and 5 for DMA1) and each channel are configurable.
- AHB and APB peripherals, FLASH, SRAM can be accessed as source and destination.
- Each channel is connected to fixed hardware DMA request.
- Software DMA channel priority (low, medium, high, ultra high) and hardware DMA channel priority (DMA channel 0 has the highest priority and DMA channel 6 has the lowest priority).
- Support independent 8, 16, 32-bit memory and peripheral transfer.
- Support independent fixed and increasing address generation algorithm of memory and peripheral.
- Support circular transfer mode.
- Support peripheral to memory, memory to peripheral, and memory to memory transfers.
- One separate interrupt per channel with three types of event flags.
- Support interrupt enable and clear.

11.3. Block diagram

Figure 11-1. Block diagram of DMA



As shown in ***Figure 11-1. Block diagram of DMA***, a DMA controller consists of four main parts:

- DMA configuration through AHB slave interface.
- Data transmission through two AHB master interfaces for memory access and peripheral access.
- An arbiter inside to manage multiple peripheral requests coming at the same time.
- Channel management to control address/data selection and data counting.

11.4. Function overview

11.4.1. DMA operation

Each DMA transfer consists of two operations, including the loading of data from the source and the storage of the loaded data to the destination. The source and destination addresses are computed by the DMA controller based on the programmed values in the DMA_CHxPADDR, DMA_CHxMADDR, and DMA_CHxCTL registers. The DMA_CHxCNT register controls how many transfers to be transmitted on the channel. The PWIDTH and MWIDTH bits in the DMA_CHxCTL register determine how many bytes to be transmitted in a transfer.

Suppose DMA_CHxCNT is 4, and both PNAGA and MNAGA are set. The DMA transfer operations for each combination of PWIDTH and MWIDTH are shown in the following ***Table 11-1. DMA transfer operation***.

Table 11-1. DMA transfer operation

Transfer size		Transfer operations	
Source	**Destination**	**Source**	**Destination**
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

The CNT bits in the DMA_CHxCNT register control how many data to be transmitted on the channel and must be configured before enable the CHEN bit in the register. During the transmission, the CNT bits indicate the remaining number of data items to be transferred.

The DMA transmission is disabled by clearing the CHEN bit in the DMA_CHxCTL register.

- If the DMA transmission is not completed when the CHEN bit is cleared, two situations may be occurred when restart this DMA channel:
  - If no register configuration operations of the channel occurs before restart the DMA channel, the DMA will continue to complete the rest of the transmission.
  - If any register configuration operations to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel occur, the DMA will restart a new transmission.
- If the DMA transmission has been finished when clearing the CHEN bit, enable the DMA channel without any register configuration operation to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel will not launch any DMA transfer.

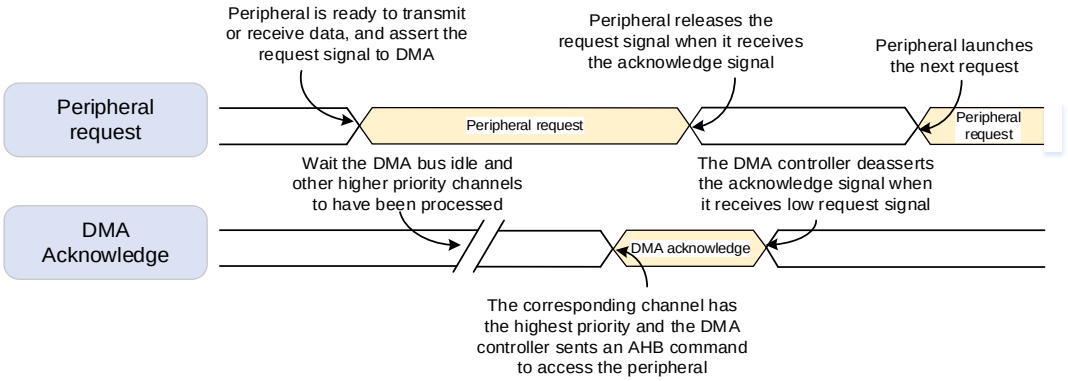
11.4.2. Peripheral handshake

To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and an acknowledge signal:

- Request signal asserted by peripheral to DMA controller, indicating that the peripheral is ready to transmit or receive data.
- Acknowledge signal responded by DMA to peripheral, indicating that the DMA controller has initiated an AHB command to access the peripheral.

Figure 11-2. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 11-2. Handshake mechanism



11.4.3. Arbitration

When two or more requests are received at the same time, the arbiter determines which request is served based on the priorities of channels. There are two-stage priorities, including the software priority and the hardware priority. The arbiter determines which channel is selected to respond according to the following priority rules:

- Software priority: Four levels, including low, medium, high and ultra-high by configuring the PRIO bits in the DMA_CHxCTL register.
- For channels with equal software priority level, priority is given to the channel with lower channel number.

11.4.4. Address generation

Two kinds of address generation algorithm are implemented independently for memory and peripheral, including the fixed mode and the increased mode. The PNAGA and MNAGA bit in the DMA_CHxCTL register are used to configure the next address generation algorithm of peripheral and memory.

In the fixed mode, the next address is always equal to the base address configured in the base address registers (DMA_CHxPADDR, DMA_CHxMADDR).

In the increasing mode, the next address is equal to the current address plus 1 or 2 or 4, depending on the transfer data width.

11.4.5. Circular mode

Circular mode is implemented to handle continue peripheral requests (for example, ADC scan mode). The circular mode is enabled by setting the CMEN bit in the DMA_CHxCTL register.

In circular mode, the CNT bits are automatically reloaded with the pre-programmed value and the full transfer finish flag is asserted at the end of every DMA transfer. DMA can always responds the peripheral request until the CHEN bit in the DMA_CHxCTL register is cleared.

11.4.6. Memory to memory mode

The memory to memory mode is enabled by setting the M2M bit in the DMA_CHxCTL register. In this mode, the DMA channel can also work without being triggered by a request from a peripheral. The DMA channel starts transferring as soon as it is enabled by setting the CHEN bit in the DMA_CHxCTL register, and completed when the DMA_CHxCNT register reaches zero.

11.4.7. Channel configuration

When starting a new DMA transfer, it is recommended to respect the following steps:

1. Read the CHEN bit and judge whether the channel is enabled or not. If the channel is enabled, clear the CHEN bit by software. When the CHEN bit is read as '0', configuring and starting a new DMA transfer is allowed.
2. Configure the M2M bit and DIR bit in the DMA_CHxCTL register to set the transfer mode.
3. Configure the CMEN bit in the DMA_CHxCTL register to enable/disable the circular mode.
4. Configure the PRIO bits in the DMA_CHxCTL register to set the channel software priority.
5. Configure the memory and peripheral transfer width, memory and peripheral address generation algorithm in the DMA_CHxCTL register.
6. Configure the enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer error interrupt in the DMA_CHxCTL register.
7. Configure the DMA_CHxPADDR register for setting the peripheral base address.
8. Configure the DMA_CHxMADDR register for setting the memory base address.
9. Configure the DMA_CHxCNT register to set the total transfer data number.
10. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the channel.

11.4.8. Interrupt

Each DMA channel has a dedicated interrupt. There are three types of interrupt event, including full transfer finish, half transfer finish, and transfer error.

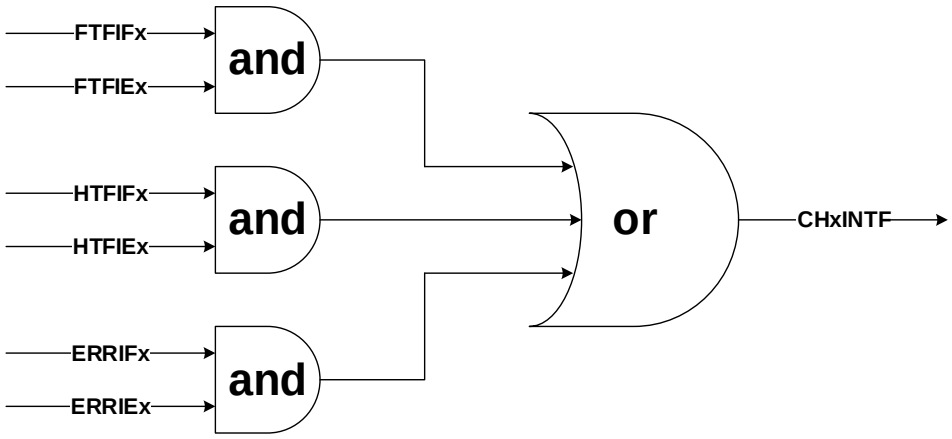
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following ***Table 11-2. interrupt events***.

Table 11-2. interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
	DMA_INTF	DMA_INTC	DMA_CHxCTL
Full transfer finish	FTFIF	FTFIFC	FTFIE
Half transfer finish	HTFIF	HTFIFC	HTFIE
Transfer error	ERRIF	ERRIFC	ERRIE

The DMA interrupt logic is shown in the ***Figure 11-3. DMA interrupt logic***, an interrupt can be produced when any type of interrupt event occurs and enabled on the channel.

Figure 11-3. DMA interrupt logic



Note: “x” indicates channel number (x=0…6).

11.4.9. DMA request mapping

The DMA requests of a channel are coming from the AHB/APB peripherals through the corresponding channel output of DMAMUX request multiplexer, refer to ***Table 12-3. Request multiplexer input mapping***.

11.5. Register definition

DMA base address: 0x4002 0000

DMA1 base address: 0x4002 0400

Note: For DMA1 having 5 channels, all bits related to channel 5 and channel 6 in the following registers are not suitable for DMA1

11.5.1. Interrupt flag register (DMA_INTF)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27/23/19/15 /11/7/3	ERRIFx	Error flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: Transfer error has not occurred on channel x 1: Transfer error has occurred on channel x
26/22/18/14 /10/6/2	HTFIFx	Half transfer finish flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
25/21/17/13 /9/5/1	FTFIFx	Full Transfer finish flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
24/20/16/12 /8/4/0	GIFx	Global interrupt flag of channel x (x=0…6) Hardware set and software cleared by configuring DMA_INTC register. 0: None of ERRIF, HTFIF or FTFIF occurs on channel x 1: At least one of ERRIF, HTFIF or FTFIF occurs on channel x

11.5.2. Interrupt flag clear register (DMA_INTC)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
27/23/19/15 /11/7/3	ERRIFCx	Clear bit for error flag of channel x (x=0…6) 0: No effect 1: Clear error flag
26/22/18/14 /10/6/2	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0…6) 0: No effect 1: Clear half transfer finish flag
25/21/17/13 /9/5/1	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=0…6) 0: No effect 1: Clear full transfer finish flag
24/20/16/12 /8/4/0	GIFCx	Clear global interrupt flag of channel x (x=0…6) 0: No effect 1: Clear GIFx, ERRIFx, HTFIFx and FTFIFx bits in the DMA_INTF register

11.5.3. Channel x control register (DMA_CHxCTL)

x = 0...6, where x is a channel number

Address offset: 0x08 + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	M2M	Memory to Memory mode

		Software set and cleared 0: Disable Memory to Memory mode 1: Enable Memory to Memory mode This bit can not be written when CHEN is ‘1’.
13:12	PRIO[1:0]	Priority level Software set and cleared 00: Low 01: Medium 10: High 11: Ultra high These bits can not be written when CHEN is ‘1’.
11:10	MWIDTH[1:0]	Transfer data size of memory Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is ‘1’.
9:8	PWIDTH[1:0]	Transfer data size of peripheral Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is ‘1’.
7	MNAGA	Next address generation algorithm of memory Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is ‘1’.
6	PNAGA	Next address generation algorithm of peripheral Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit can not be written when CHEN is ‘1’.
5	CMEN	Circular mode enable Software set and cleared 0: Disable circular mode 1: Enable circular mode This bit can not be written when CHEN is ‘1’.
4	DIR	Transfer direction

		Software set and cleared 0: Read from peripheral and write to memory 1: Read from memory and write to peripheral This bit can not be written when CHEN is ‘1’.
3	ERRIE	Enable bit for channel error interrupt Software set and cleared 0: Disable the channel error interrupt 1: Enable the channel error interrupt
2	HTFIE	Enable bit for channel half transfer finish interrupt Software set and cleared 0: Disable channel half transfer finish interrupt 1: Enable channel half transfer finish interrupt
1	FTFIE	Enable bit for channel full transfer finish interrupt Software set and cleared 0: Disable channel full transfer finish interrupt 1: Enable channel full transfer finish interrupt
0	CHEN	Channel enable Software set and cleared 0: Disable channel 1: Enable channel

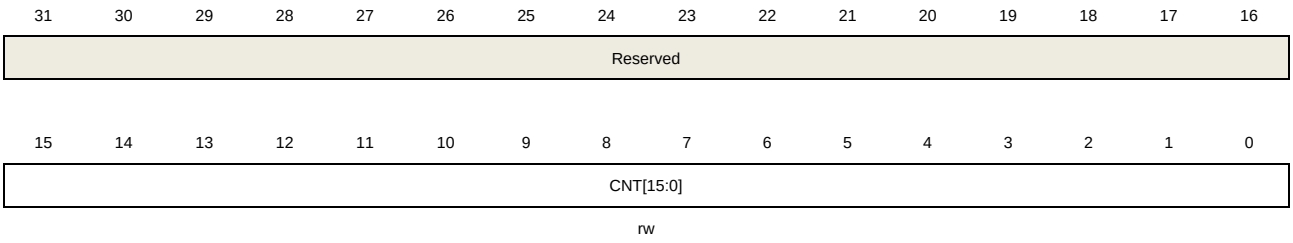
11.5.4. Channel x counter register (DMA_CHxCNT)

x = 0...6, where x is a channel number

Address offset: 0x0C + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Transfer counter These bits can not be written when CHEN in the DMA_CHxCTL register is ‘1’. This register indicates how many transfers remain. Once the channel is enabled, it is read-only, and decreases after each DMA transfer. If the register is zero, no transaction can be issued whether the channel is enabled or not. Once the

transmission of the channel is complete, the register can be reloaded automatically by the previously programmed value if the channel is configured in circular mode.

11.5.5. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...6, where x is a channel number

Address offset: 0x10 + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:0	PADDR[31:0]	Peripheral base address These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. When PWIDTH is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address. When PWIDTH is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

11.5.6. Channel x memory base address register (DMA_CHxMADDR)

x = 0...6, where x is a channel number

Address offset: 0x14 + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:0	MADDR[31:0]	Memory base address These bits can not be written when CHEN in the DMA_CHxCTL register is '1'. When MWIDTH in the DMA_CHxCTL register is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address.

When MWIDTH in the DMA_CHxCTL register is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

12. DMA request multiplexer (DMAMUX)

12.1. Overview

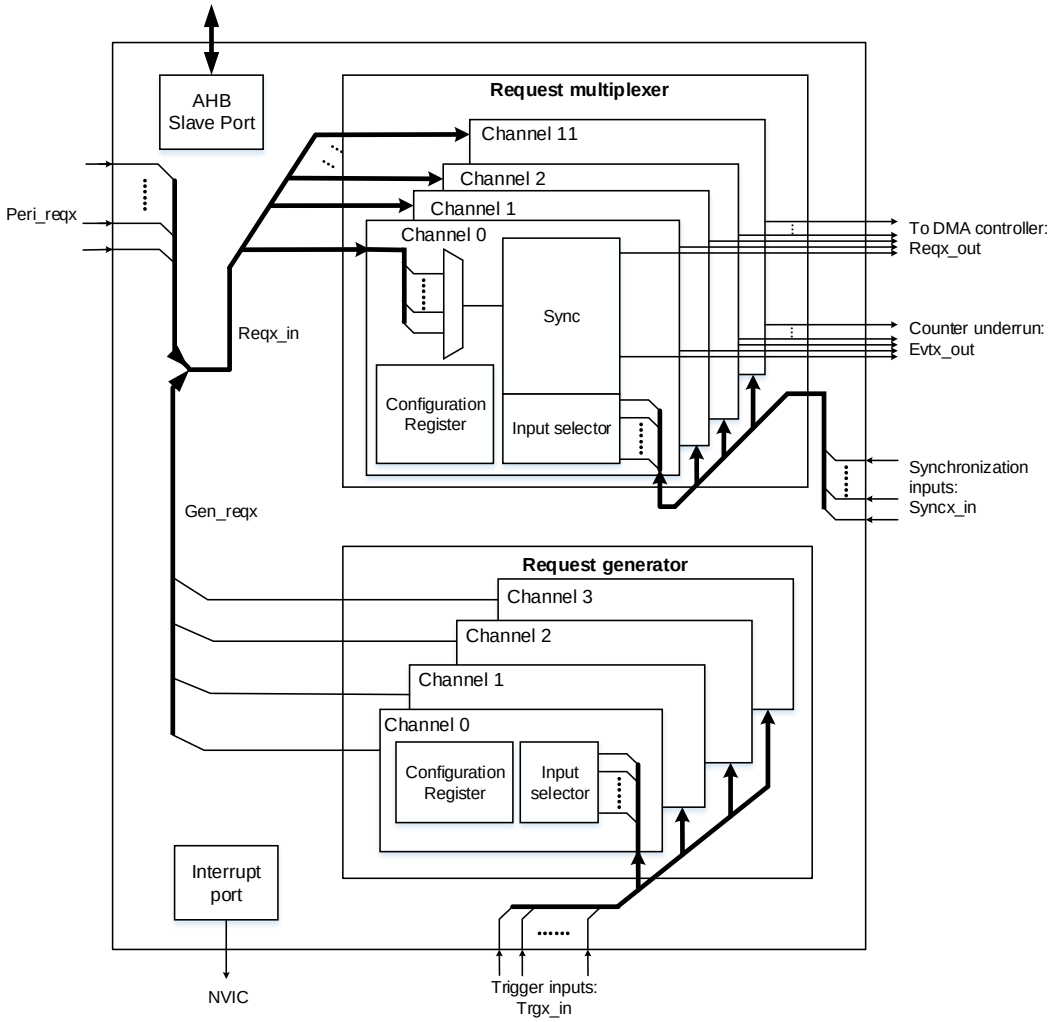
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, unconditionally or synchronously with events from its DMAMUX synchronization inputs. The DMA request is pending until it is served by the DMA controller which generates a DMA acknowledge signal (the DMA request signal is de-asserted).

12.2. Characteristics

- 12 channels for DMAMUX request multiplexer.
- 4 channels for DMAMUX request generator.
- Support 27 trigger inputs.
- Support 27 synchronization inputs.
- Each DMAMUX request generator channel:
  - DMA request trigger input selector
  - DMAMUX request generator counter
  - Trigger overrun flag
- Each DMAMUX request multiplexer channel:
  - 79 input DMA request lines from peripherals
  - Synchronization input selector
  - One DMA request line output
  - One channel event output, for DMA request chaining
  - DMAMUX request multiplexer counter
  - Synchronization overrun flag

12.3. Block diagram

Figure 12-1. Block diagram of DMAMUX



12.4. Function overview

As shown in ***Figure 12-1. Block diagram of DMAMUX***, DMAMUX includes two sub-blocks:

- DMAMUX request multiplexer.

  DMAMUX request multiplexer inputs (Reqx_in) source from:
  - Peripherals (Peri_reqx).
  - DMAMUX request generator outputs (Gen_reqx).

  DMAMUX request multiplexer outputs (Reqx_out) is connected to channels of DMA controller.

  Synchronization inputs (Syncx_in) source from internal or external signals.

- DMAMUX request generator.

  Trigger inputs (Trgx_in) source from internal or external signals.

12.4.1. DMAMUX signals

Table 12-1. DMAMUX signals

Signal name	Discription
Reqx_in	DMAMUX request multiplexer inputs (from peripheral requests and request generator channels)
Peri_reqx	DMAMUX DMA request line inputs from peripherals
Gen_reqx	DMAMUX generated DMA request from request generator
Reqx_out	DMAMUX requests outputs (to DMA controller)
Trgx_in	DMAMUX DMA request triggers inputs (to request generator)
Syncx_in	DMAMUX synchronization inputs (to request multiplexer)
Evtx_out	DMAMUX request multiplexer counter underrun event outputs

12.4.2. DMAMUX request multiplexer

The DMAMUX request multiplexer enables routing a DMA request line between the peripherals / generated DMA request and the DMA controllers of the product. Its component unit is the request multiplexer channels. DMA request lines are connected in parallel to all request multiplexer channels. There is a synchronization unit for each request multiplexer channel. The synchronization inputs are connected in parallel to all synchronization unit of request multiplexer channels. And there is a built-in DMAMUX request multiplexer counter for each request multiplexer channel.

Request multiplexer channel

A DMA request input for the DMAMUX request multiplexer channel x is configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 12-3. Request multiplexer input mapping***. A DMAMUX request multiplexer channel is connected and dedicated to one single channel of the DMA controller.

Note: The value 0 of MUXID[6:0] bits corresponds to no DMA request line is selected. It is not allowed to configure the same DMA request line (same non-null MUXID[6:0]) to two different request multiplexer channels.

When synchronization mode is disabled

Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register.

If the channel event generation is enabled by setting EVGEN bit, the number of DMA requests before an output event generation is NBR[4:0] + 1.

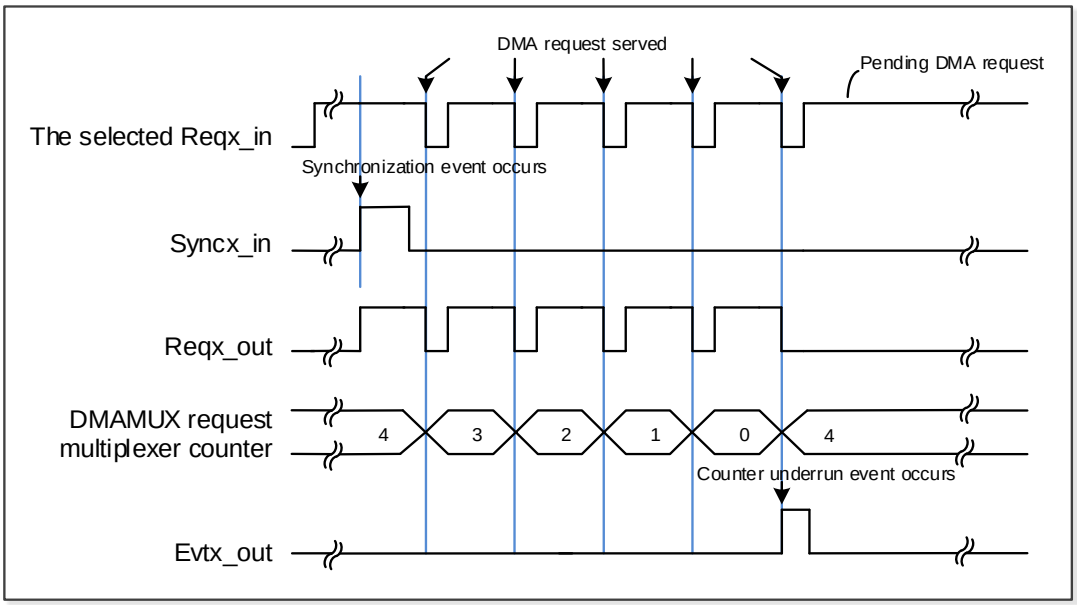
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

When synchronization mode is enabled

A channel x in synchronization mode, when a rising/falling edge on the selected synchronization input is detected, the pending selected input DMA request line is routed to the multiplexer channel x output. Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the input DMA request line is disconnected from the request multiplexer channel x output, and the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register. The number of DMA requests transferred to the request multiplexer channel x output following a detected synchronization event is NBR[4:0] + 1.

Figure 12-2. Synchronization mode shows an example when NBR[4:0]=4, SYNCEN=1, EVGEN=1, SYNCP[1:0]=01.

Figure 12-2. Synchronization mode



DMAMUX request multiplexer channel x can be synchronized by setting the synchronization enable bit SYNCEN in the DMAMUX_RM_CHxCFG register. The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to ***Table 12-5. Synchronization input mapping***. The synchronization input valid edge is configured by the SYNCP[1:0] bits of the DMAMUX_RM_CHxCFG register.

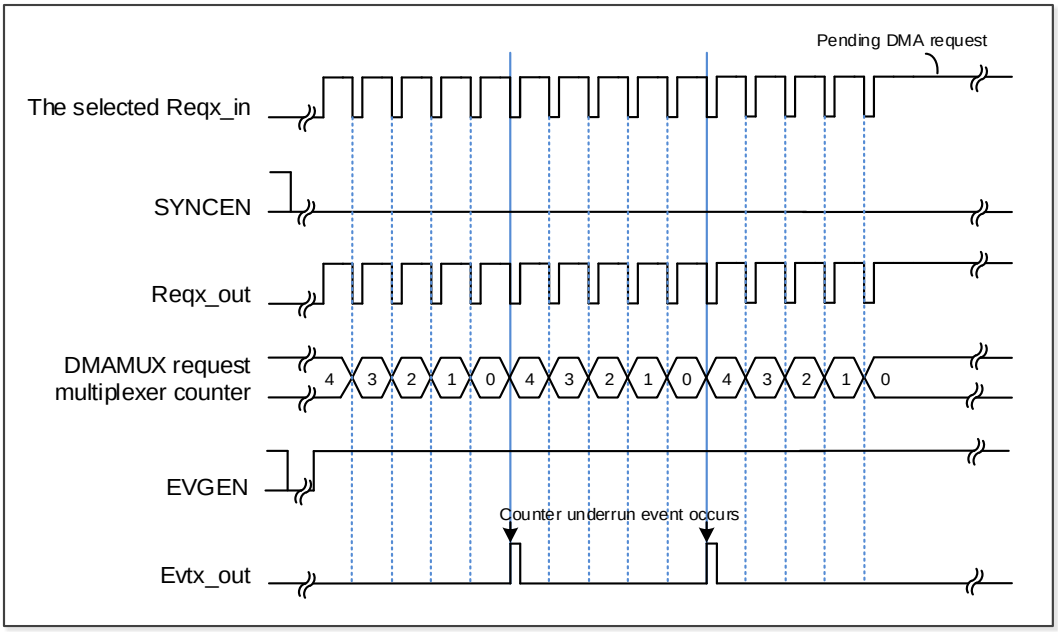
Note: If a synchronization input event occurs when there is no pending selected input DMA request line, the input event is discarded. The following asserted input request lines will not be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

Channel event generation

Each DMA request line multiplexer channel has an event output called Evtx_out, which is the DMA request multiplexer counter underrun event. Signals dmamux_evt0 ~ dmamux_evt3 can be used for DMA request chaining. If event generation bit EVGEN in the DMAMUX_RM_CHxCFG register is enabled on the channel x output, when its DMA request multiplexer counter is automatically reloaded with the value of the programmed NBR[4:0] field, the multiplexer channel generates a channel event, as a pulse of one AHB clock cycle.

Figure 12-3. Event generation shows an example when NBR[4:0]=4, SYNCEN=0, EVGEN=1.

Figure 12-3. Event generation



Note: If EVGEN = 1 and NBR[4:0] = 0, an event is generated after each served DMA request.

Synchronization overrun

If a new synchronization event occurs before the built-in DMAMUX request multiplexer counter underrun, the synchronization overrun flag bit SOIFx is set in the DMAMUX_RM_INTF register.

Note: The synchronization mode of request multiplexer channel x shall be disabled by resetting SYNCEN bit in DMAMUX_RM_CHxCFG register at the completion of the use of the related channel of the DMA controller. Otherwise, when a new synchronization event occurs,

there will be a synchronization overrun due to the absence of a DMA acknowledge (that is, no served request) received from the DMA controller.

12.4.3. DMAMUX request generator

The DMAMUX request generator produces DMA requests upon trigger input event. Its component unit is the request generator channels. DMA request trigger inputs are connected in parallel to all request generator channels. And there is a built-in DMAMUX request generator counter for each request generator channel.

The active edge of trigger input events is selected through the RGTP[1:0] bits in DMAMUX_RG_CHxCFG register. The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 12-4. Trigger input mapping***. DMAMUX request generator channel x can be enabled by setting RGEN to 1 in DMAMUX_RG_CHxCFG register.

Request generator channel

Upon the trigger input event, the corresponding request generator channel starts generating DMA requests on its output, and the output goes to the input of the DMAMUX request multiplexer. Each time the DMAMUX generated request is served by the connected DMA controller, the served request will be de-asserted, and the built-in DMAMUX request generator counter of the request generator channel is decremented. At the request generator counter underrun, the request generator channel stops generating DMA requests. The built-in DMAMUX request generator counter will be automatically reloaded to its programmed value upon the next trigger input event, the built-in counter is programmed by the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register.

Note: The number of generated DMA requests after the trigger input event is NBRG[4:0] + 1. The NBRG[4:0] value shall only be written by software when the RGEN bit of the corresponding generator channel x is disabled.

Trigger overrun

If a request generator channel x was enabled by RGEN bit, when a new DMA request trigger event for the request generator channel x occurs before the DMAMUX request generator counter underrun, then the request trigger overrun event flag bit TOIFx is set by hardware in the DMAMUX_RG_INTF register.

Note: The request generator channel x shall be disabled by resetting RGEN bit in DMAMUX_RG_CHxCFG register at the completion of the usage of the related channel of the DMA controller. Otherwise, when a new detected trigger event occurs, there will be a trigger overrun due to the absence of an acknowledge (that is, no served request) received from the DMA.

12.4.4. Channel configurations

The following sequence should be followed to configure a DMAMUX channel y and the related DMA channel x:

1. Set and configure the DMA channel x completely, except enabling the channel x.
2. Set and configure the related DMAMUX channel y completely.
3. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the DMA channel x.

12.4.5. Interrupt

There are two types of interrupt event, including synchronization overrun event on each DMAMUX request multiplexer channel, and trigger overrun event on each DMAMUX request generator channel.

Each interrupt event has a dedicated flag bit, a dedicated clear bit, and a dedicated enable bit. The relationship is described in the following ***Table 12-2. Interrupt events***.

Table 12-2. Interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
Synchronization overrun event on DMAMUX request multiplexer channel x	SOIFx	SOIFCx	SOIE
Trigger overrun event on DMAMUX request generator channel y	TOIFy	TOIFCy	TOIE

Trigger overrun interrupt

When the DMAMUX request trigger overrun flag TOIFx is set, and the trigger overrun interrupt is enabled by setting TOIE bit, a trigger overrun interrupt will be generated. The overrun flag TOIFx is reset by writing 1 to the corresponding clear bit of overrun flag TOIFCx in the DMAMUX_RG_INTC register.

Synchronization overrun interrupt

When the synchronization overrun flag SOIFx is set, and the synchronization overrun interrupt is enabled by setting SOIE bit, a synchronization overrun interrupt will be generated. The overrun flag SOIFx is reset by writing 1 to the corresponding clear bit of synchronization overrun flag bit SOIFCx in the DMAMUX_RM_INTC register.

12.4.6. DMAMUX mapping

Request multiplexer input mapping

A DMA request is sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 12-3. Request multiplexer input mapping***, configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register for the DMAMUX request multiplexer channel x.

Table 12-3. Request multiplexer input mapping

Request multiplexer channel input identification MUXID[6:0]	Source
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC
6	DAC_CH0
7	Reserved
8	I2C1_RX
9	I2C1_TX
10	I2C0_RX
11	I2C0_TX
12	MFCOM_SSTAT0
13	MFCOM_SSTAT1
14	MFCOM_SSTAT2
15	MFCOM_SSTAT3
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	TIMER0_CH0
21	TIMER0_CH1
22	TIMER0_CH2
23	TIMER0_CH3
24	TIMER0_TI
25	TIMER0_UP
26	TIMER0_CO
27	TIMER0_MCH0
28	TIMER0_MCH1
29	TIMER0_MCH2
30	TIMER0_MCH3
31	TIMER1_CH0
32	TIMER1_CH1
33	TIMER1_CH2
34	TIMER1_CH3

Request multiplexer channel input identification MUXID[6:0]	Source
35	TIMER1_TI
36	TIMER1_UP
37	TIMER7_CH0
38	TIMER7_CH1
39	TIMER7_CH2
40	TIMER7_CH3
41	TIMER7_TI
42	TIMER7_UP
43	TIMER7_CO
44	TIMER7_MCH0
45	TIMER7_MCH1
46	TIMER7_MCH2
47	TIMER7_MCH3
48	CAN1
49	CAN0
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	USART2_RX
55	USART2_TX
56	TIMER5_UP
57	TIMER6_UP
58	TIMER19_CH0
59	TIMER19_CH1
60	TIMER19_CH2
61	TIMER19_CH3
62	TIMER19_TI
63	TIMER19_UP
64	TIMER19_CO
65	TIMER19_MCH0
66	TIMER19_MCH1
67	TIMER19_MCH2
68	TIMER19_MCH3
69	TIMER20_CH0
70	TIMER20_CH1
71	TIMER20_CH2
72	TIMER20_CH3
73	TIMER20_TI
74	TIMER20_UP

Request multiplexer channel input identification MUXID[6:0]	Source
75	TIMER20_CO
76	TIMER20_MCH0
77	TIMER20_MCH1
78	TIMER20_MCH2
79	TIMER20_MCH3

Trigger input mapping

The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 12-4. Trigger input mapping***.

Table 12-4. Trigger input mapping

Trigger input identification TID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	Reserved
21	Reserved
22	TIMER20_CH0_O
23	Reserved
24	Reserved
25	Reserved

Trigger input identification TID[4:0]	Source
26	Reserved

Synchronization input mapping

The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to ***Table 12-5. Synchronization input mapping***.

Table 12-5. Synchronization input mapping

Synchronization input identification SYNCID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	Reserved
21	Reserved
22	TIMER20_CH0_O
23	Reserved
24	Reserved
25	Reserved
26	Reserved

12.5. Register definition

DMAMUX base address: 0x4002 0800

12.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

x = 0...11, where x is a channel number

Address offset: 0x00 + 0x04 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			SYNCID[4:0]					NBR[4:0]					SYNCP[1:0]		SYNCEN
			rw					rw					rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EVGEN	SOIE	Reserved	MUXID[6:0]						
						rw	rw		rw						

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:24	SYNCID[4:0]	Synchronization input identification Selects the synchronization input source.
23:19	NBR[4:0]	Number of DMA requests to forward The number of DMA requests to forward to the DMA controller after a synchronization event / before an output event is generated equals to NBR[4:0] + 1. These bits shall only be written when both SYNCEN and EVGEN bits are disabled.
18:17	SYNCP[1:0]	Synchronization input polarity 00: No event detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	SYNCEN	Synchronization enable 0: Disable synchronization 1: Enable synchronization
15:10	Reserved	Must be kept at reset value.
9	EVGEN	Event generation enable 0: Disable event generation 1: Enable event generation

8	SOIE	Synchronization overrun interrupt enable 0: Disable interrupt 1: Enable interrupt
7	Reserved	Must be kept at reset value.
6:0	MUXID[6:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

12.5.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF)

Address offset: 0x80
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SOIF11	SOIF10	SOIF9	SOIF8	SOIF7	SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
				r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	SOIFx	Synchronization overrun event flag of request multiplexer channel x (x=0..11) The flag is set when a synchronization event occurs on a DMA request line multiplexer channel x, while the DMA request counter value is lower than NBR[4:0]. The flag is cleared by writing 1 to the corresponding SOIFCx bit in DMAMUX_RM_INTC register.

12.5.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC)

Address offset: 0x084
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SOIFC11	SOIFC10	SOIFC9	SOIFC8	SOIFC7	SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
				w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	SOIFCx	Clear bit for synchronization overrun event flag of request multiplexer channel x (x=0..11) Writing 1 clears the corresponding overrun flag SOIFx in the DMAMUX_RM_INTF register.

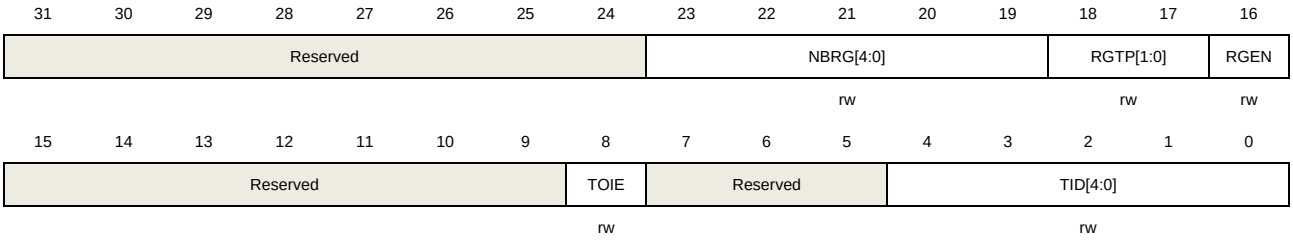
12.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

x = 0...3, where x is a channel number

Address offset: 0x100 + 0x04 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:19	NBRG[4:0]	Number of DMA requests to be generated The number of DMA requests to be generated after a trigger event equals to NBRG[4:0] + 1. **Note:** These bits shall only be written when RGEN bit is disabled.
18:17	RGTP[1:0]	DMA request generator trigger polarity 00: No event trigger detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	RGEN	DMA request generator channel x enable 0: Disable DMA request generator channel x 1: Enable DMA request generator channel x
15:9	Reserved	Must be kept at reset value.
8	TOIE	Trigger overrun interrupt enable 0: Disable interrupt 1: Enable interrupt

7:5	Reserved	Must be kept at reset value.
4:0	TID[4:0]	Trigger input identification Selects the DMA request trigger input source.

12.5.5. Request generator interrupt flag register (DMAMUX_RG_INTF)

Address offset: 0x140

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TOIF3	TOIF2	TOIF1	TOIF0
												r	r	r	r

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	TOIFx	Trigger overrun event flag of request generator channel x (x=0..3) The flag is set when a new trigger event occurs on DMA request generator channel x, before the request counter underrun (the internal request counter programmed via the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register). The flag is cleared by writing 1 to the corresponding TOIFCx bit in the DMAMUX_RG_INTC register.

12.5.6. Request generator interrupt flag clear register (DMAMUX_RG_INTC)

Address offset: 0x144

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	TOIFCx	Clear bit for trigger overrun event flag of request generator channel x (x=0..3)

Writing 1 in each bit clears the corresponding overrun flag TOIFx in the DMAMUX_RG_INTF register.

13. Debug (DBG)

13.1. Introduction

The GD32A50x series provide a large variety of debug and test features. They are implemented with a standard configuration of the Arm® CoreSightTM module together with a daisy chained standard TAP controller. Debug function is integrated into the Arm® Cortex®-M33. The debug system supports serial wire debug (SWD) function in addition to standard JTAG debug. The debug function refer to the following documents:

- Cortex®-M33 Technical Reference Manual
- Arm Debug Interface v5 Architecture Specification

The DBG hold unit helps debugger to debug power saving mode, TIMER, I2C, WWDGT, FWDGT, CAN and MFCOM. When corresponding bit is set, provide clock when in power saving mode or hold the state for TIMER, WWDGT, FWDGT, CAN, I2C or MFCOM.

13.2. JTAG/SW function overview

Debug capabilities can be accessed by a debug tool via serial wire (SW - Debug Port) or JTAG interface (JTAG - Debug Port).

13.2.1. Switch JTAG or SW interface

By default, the JTAG interface is active. The sequence for switching from JTAG to SWD is:

- Send 50 or more TCK cycles with TMS = 1.
- Send the 16-bit sequence on TMS = 1110011110011110 (0xE79E LSB first).
- Send 50 or more TCK cycles with TMS = 1.

The sequence for switching from SWD to JTAG is:

- Send 50 or more TCK cycles with TMS = 1.
- Send the 16-bit sequence on TMS = 1110011100111100 (0xE73C LSB first).
- Send 50 or more TCK cycles with TMS = 1.

13.2.2. Pin assignment

The JTAG interface provides 5-pin standard JTAG, known as JTAG clock (JTCK), JTAG mode selection (JTMS), JTAG data input (JTDI), JTAG data output (JTDO) and JTAG reset (NJTRST, active low). The serial wire debug (SWD) provide 2-pin SW interface, known as SW data input/output (SWDIO) and SW clock (SWCLK). The two SW pin are multiplexed with two of five JTAG pin, which is SWDIO multiplexed with JTMS, SWCLK multiplexed with JTCK.

The JTDO is also used as trace async data output (TRACESWO) when async trace enabled.

Table 13-1. Pin assignment

Pin	Debug interface
PB7	JTDI
PB8	JTCK/SWCLK
PB9	JTMS/SWDIO
PB3	NJTRST
PB4	JTDO

By default, 5-pin standard JTAG debug mode is chosen after reset. Users can also use JTAG function without NJTRST pin, then the PB3 can be used to other GPIO functions (NJTRST tied to 1 by hardware). If switch to SW debug mode, the PB7/PB4/PB3 are released to other GPIO functions. If JTAG and SW not used, all 5-pin can be released to other GPIO functions. Please refer to ***GPIO pin configuration***.

13.2.3. JTAG daisy chained structure

The Cortex®-M33 JTAG TAP is connected to a boundary-scan (BSD) JTAG TAP. The BSD JTAG IR is 5-bit width, while the Cortec-M33 JTAG IR is 4-bit width. So when JTAG in IR shift step, it first shift 5-bit BYPASS instruction (5'b 11111) for BSD JTAG, and then shift normal 4-bit instruction for Cortex®-M33 JTAG. Because of the data shift under BSD JTAG BYPASS mode, adding 1 extra bit to the data chain is needed.

The BSD JTAG IDCODE is 0x06418041.

13.2.4. Debug reset

The JTAG-DP and SW-DP registers are in the power on reset domain. The system reset initializes the majority of the Cortex®-M33, excluding NVIC and debug logic, (FPB, DWT, and ITM). The NJTRST reset can reset JTAG TAP controller only. So, it can perform debug feature under system reset. Such as, halt-after-reset, which is the debugger sets halt under system reset, and the core halts immediately after the system reset is released.

13.2.5. JEDEC-106 ID code

The Cortex®-M33 integrates JEDEC-106 ID code, which is located in ROM table and mapped on the address of 0xE00FF000_0xE00FFFFF.

13.3. Debug hold function overview

13.3.1. Debug support for power saving mode

When STB_HOLD bit in DBG control register (DBG_CTL) is set and entering the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, and the debugger can debug in standby mode. When exit the standby mode, a system reset generated.

When DSLP_HOLD bit in DBG control register (DBG_CTL) is set and entering the deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M, and the debugger can debug in deep-sleep mode.

When SLP_HOLD bit in DBG control register (DBG_CTL) is set and entering the sleep mode, the clock of AHB bus for CPU is not closed, and the debugger can debug in sleep mode.

13.3.2. Debug support for TIMER, I2C, WWDGT and FWDGT

When the core halted and the corresponding bit in DBG control register (DBG_CTL) is set, the following behaved.

For TIMER, the timer counters stopped and hold for debug.

For I2C, SMBUS timeout hold for debug.

For WWDGT or FWDGT, the counter clock stopped for debug.

For CAN, the receive register is stopped counting for debugging.

For MFCOM, the counter clock stopped for debug.

13.4. Registers definition

DEBUG base address: 0xE004 4000

13.4.1. ID code register (DBG_ID)

Address offset: 0x00
Read only

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID_CODE[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_CODE[15:0]															
r															

Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register These bits read by software. These bits are unchanged constant.

13.4.2. Control register (DBG_CTL)

Address offset: 0x04
Reset value: 0x0000 0000, power reset only.

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER19_HOLD	TIMER20_HOLD	Reserved						CAN1_HOLD	CAN0_HOLD	MFCOM_HOLD	TIMER6_HOLD	TIMER5_HOLD	Reserved	TIMER7_HOLD	I2C1_HOLD
rw	rw							rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	Reserved			TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	Reserved					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw				rw	rw	rw	rw	rw		rw			rw	rw	rw

Bits	Fields	Descriptions
31	TIMER19_HOLD	TIMER19 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER19 counter for debug when core halted.
30	TIMER20_HOLD	TIMER20 hold bit This bit is set and reset by software. 0: no effect

		1: Hold the TIMER10 counter for debug when core halted.
29:24	Reserved	Must be kept at reset value.
23	CAN1_HOLD	CAN1 hold bit This bit is set and reset by software. 0: no effect 1: Hold the CAN1 counter for debug when core halted.
22	CAN0_HOLD	CAN0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the CAN0 counter for debug when core halted.
21	MFCOM_HOLD	MFCOM hold bit This bit is set and reset by software. 0: no effect 1: Hold the MFCOM counter for debug when core halted.
20	TIMER6_HOLD	TIMER6 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER6 counter for debug when core halted.
19	TIMER5_HOLD	TIMER5 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER5 counter for debug when core halted.
18	Reserved	Must be kept at reset value.
17	TIMER7_HOLD	TIMER7 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER7 counter for debug when core halted.
16	I2C1_HOLD	I2C1 hold bit This bit is set and reset by software. 0: no effect 1: Hold the I2C1 SMBUS timeout for debug when core halted.
15	I2C0_HOLD	I2C0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the I2C0 SMBUS timeout for debug when core halted.
14:12	Reserved	Must be kept at reset value.
11	TIMER1_HOLD	TIMER1 hold bit This bit is set and reset by software.

		0: no effect 1: Hold the TIMER1 counter for debug when core halted.
10	TIMER0_HOLD	TIMER0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER0 counter for debug when core halted.
9	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software. 0: no effect 1: Hold the WWDGT counter clock for debug when core halted.
8	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software. 0: no effect 1: Hold the FWDGT counter clock for debug when core halted.
7:3	Reserved	Must be kept at reset value.
2	STB_HOLD	Standby mode hold register This bit is set and reset by software. 0: no effect 1: At the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M (Configured by software), a system reset generated when exit standby mode.
1	DSLP_HOLD	Deep-sleep mode hold register This bit is set and reset by software. 0: no effect 1: At the deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M.
0	SLP_HOLD	Sleep mode hold register This bit is set and reset by software. 0: no effect 1: At the sleep mode, the clock of AHB is on.

14. Analog-to-digital converter (ADC)

14.1. Overview

A 12-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip, which can sample analog signals from 16 external channels and 2 internal channels. The 18 ADC sampling channels all support a variety of operation modes. After sampling and conversion, the conversion results can be stored in the corresponding data registers according to the least significant bit (LSB) alignment or the most significant bit (MSB) alignment. An on-chip hardware oversample scheme improves performances and reduces the computational burden of MCU.

For motors, power supplies and other applications that have a higher demand for ADC, you can contact our sales staff for more ADC details.

14.2. Characteristics

- High performance.
  - ADC sampling resolution:12-bit, 10-bit, 8-bit or 6-bit.
  - Foreground calibration function.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit.
  - DMA support.
- Analog input channels.
  - 16 external analog inputs.
  - 1 channel for internal temperature sensor (V_{SENSE}).
  - 1 channel for internal reference voltage (V_{REFINT}).
- Start-of-conversion can be initiated.
  - By software.
  - By TRIGSEL.
- Operation modes.
  - Converts a single channel or scans a sequence of channels.
  - Single operation mode converts selected inputs once per trigger.
  - Continuous operation mode converts selected inputs continuously.
  - Discontinuous operation mode.
  - SYNC mode (the device with two ADCs).
- Conversion result threshold monitor function: analog watchdog.
- Interrupt generation.
  - At the end of routine sequence conversion.
  - Analog watchdog event.
- Oversampler.
  - 16-bit data register.
  - Oversampling ratio adjustable from 2x to 256x.

  - Programmable data shift up to 8-bit.
- Channel input range: $V_{REFN} \leq V_{IN} \leq V_{REFP}$.

14.3. Pins and internal signals

Figure 14-1. ADC module block diagram shows the ADC block diagram. *Table 14-1. ADC internal input signals* gives the ADC internal signals and *Table 14-2. ADC input pins definition* gives the ADC pin description.

Table 14-1. ADC internal input signals

Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{REFINT}	Internal voltage reference output voltage

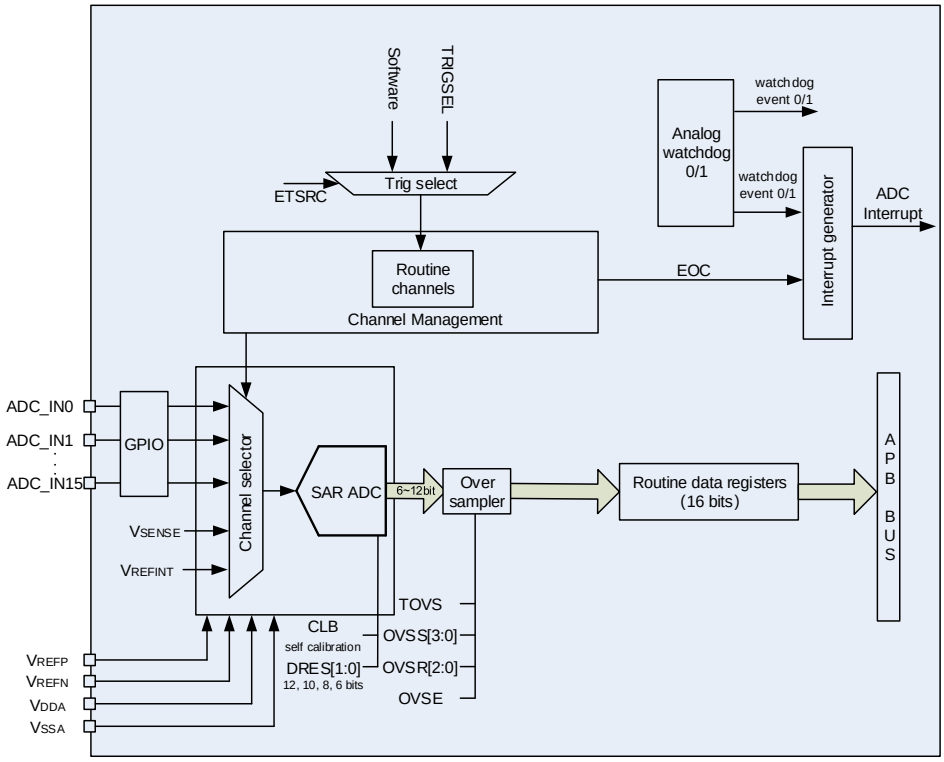
Table 14-2. ADC input pins definition

Name	Description
V_{DDA}	Analog power supply equals to V_{DD}
V_{SSA}	Ground for analog power supply equals to V_{SS}
V_{REFP}	The positive reference voltage for the ADC.
V_{REFN}	The negative reference voltage for the ADC.
ADCx_IN[15:0]	Up to 16 external channels

Note: V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS} respectively.

14.4. Function overview

Figure 14-1. ADC module block diagram



14.4.1. Foreground calibration function

During the foreground calibration procedure, the ADC calculates a calibration factor which is internally applied to the ADC until the next ADC power-off. The application must not use the ADC during calibration and must wait until it is completed. Calibration should be performed before starting A/D conversion. The calibration is initiated by setting bit CLB=1. CLB bit stays at 1 during all the calibration sequence. It is then cleared by hardware as soon as the calibration is completed.

When the ADC operating conditions change (such as supply power voltage V_{DDA}, positive reference voltage V_{REFP}, temperature and so on), it is recommended to re-run a calibration cycle.

The internal analog calibration can be reset by setting the RSTCLB bit in ADC_CTL1 register.

Calibration software procedure:

1. Ensure that ADCON=1.
2. Delay 14 CK_ADC to wait for ADC stability.
3. Set RSTCLB (optional).
4. Set CLB=1.
5. Wait until CLB=0.

14.4.2. ADC clock

The CK_ADC clock is synchronous with the AHB clock and provided by the clock controller. ADC clock can be divided and configured by RCU controller.

14.4.3. ADC enable

The ADCON bit on the ADC_CTL1 register is the enable switch of the ADC module. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is reset, the analog submodule will be put into power down mode. After ADC is enabled, you need delay t_{SU} time for sampling (the value of t_{SU} please refer to the device datasheet).

14.4.4. Routine sequence

The channel management circuit can organize the sampling conversion channels into a sequence: routine sequence.

The routine sequence supports up to 16 channels, and each channel is called routine channel. The RL[3:0] bits in the ADC_RSQ0 register specify the total conversion sequence length.The ADC_RSQ0~ADC_RSQ2 registers specify the selected channels of the routine sequence.

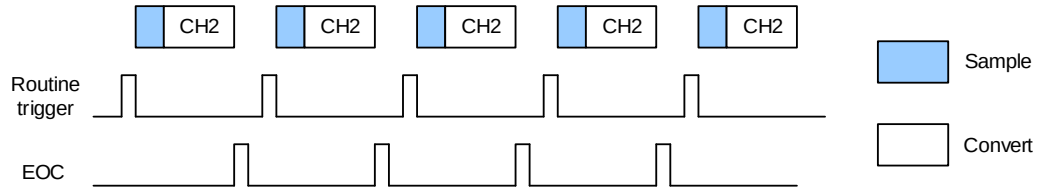
Note: Although the ADC supports 18 multiplexed channels, the maximum length of the routine sequence is only 16.

14.4.5. Operation modes

Single operation mode

This mode can be running on routine sequence. In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits of ADC_RSQ2. When the ADCON has been set high, the ADC samples and converts a single channel, once the corresponding software trigger or TRIGSEL trigger is active.

Figure 14-2. Single operation mode



After conversion of a single routine channel, the conversion data will be stored in the ADC_RDATA register, the EOC will be set. An interrupt will be generated if the EOCIE bit is set.

Software procedure for single operation mode of a routine channel:

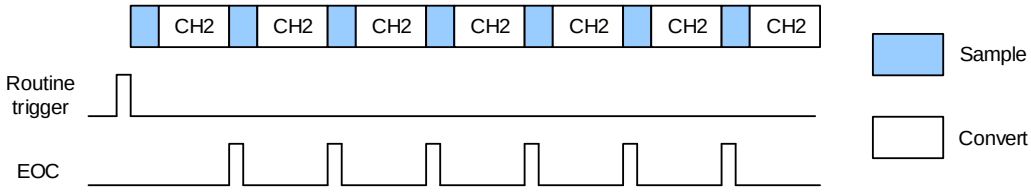
1. Make sure the DISRC, SM in the ADC_CTL0 register and CTN bit in the ADC_CTL1 register are reset.

2. Configure RSQ0 in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Set the SWRCST bit, or generate a TRIGSEL trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Read the converted data in the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.

Continuous operation mode

This mode can be running on routine sequence. The continuous operation mode will be enabled when CTN bit in the ADC_CTL1 register is set. In this mode, the ADC performs conversion on the channel specified in the RSQ0[4:0]. When the ADCON has been set high, the ADC samples and converts specified channel, once the corresponding software trigger or TRIGSEL trigger is active. The conversion data will be stored in the ADC_RDATA register.

Figure 14-3. Continuous operation mode



Software procedure for continuous operation mode on a routine channel:

1. Set the CTN bit in the ADC_CTL1 register.
2. Configure RSQ0 in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Set the SWRCST bit, or generate a TRIGSEL trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Read the converted data in the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.
9. Repeat steps 6~8 as soon as the conversion is in need.

To get rid of checking EOC bit, DMA can be used to transfer the converted data:

1. Set the CTN and DMA bit in the ADC_CTL1 register.
2. Configure RSQ0 in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate a TRIGSEL trigger for the routine sequence.

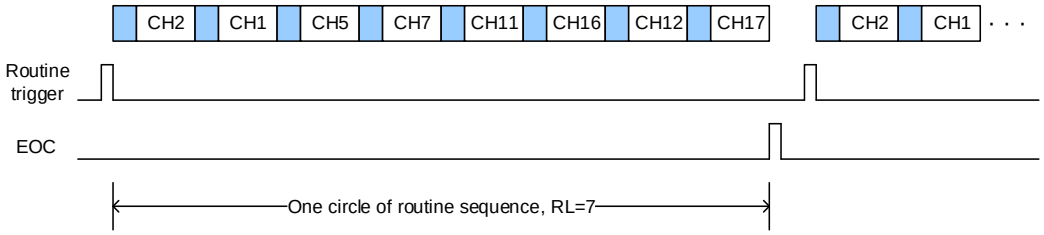
Scan operation mode

The scan operation mode will be enabled when SM bit in the ADC_CTL0 register is set. In

this mode, the ADC performs conversion on the channels with a specific sequence specified in the ADC_RSQ0~ADC_RSQ2 registers. When the ADCON has been set high, the ADC samples and converts specified channels one by one in the routine sequence till the end of the routine sequence, once the corresponding software trigger or TRIGSEL trigger is active. The conversion data will be stored in the ADC_RDATA register. After conversion of the routine sequence, the EOC will be set. An interrupt will be generated if the EOCIE bit is set. The DMA bit in ADC_CTL1 register must be set when the routine sequence works in scan mode.

After conversion of a routine sequence, the conversion can be restarted automatically if the CTN bit in the ADC_CTL1 register is set.

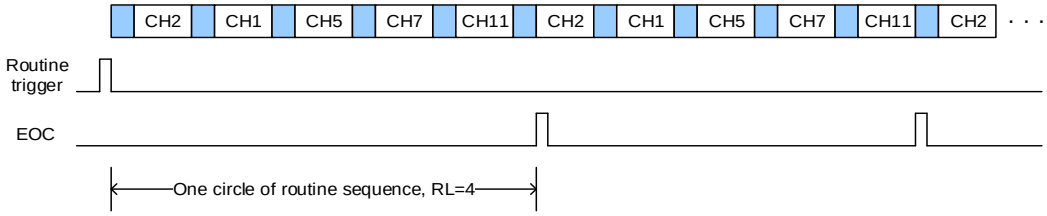
Figure 14-4. Scan operation mode, continuous disable



Software procedure for scan operation mode on a routine sequence:

1. Set the SM bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register.
2. Configure ADC_RSQx and ADC_SAMPTx registers.
3. Configure ETERC and ETSRC bits in the ADC_CTL1 register if in need.
4. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
5. Set the SWRCST bit, or generate a TRIGSEL trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Clear the EOC flag by writing 0 to it.

Figure 14-5. Scan operation mode, continuous enable

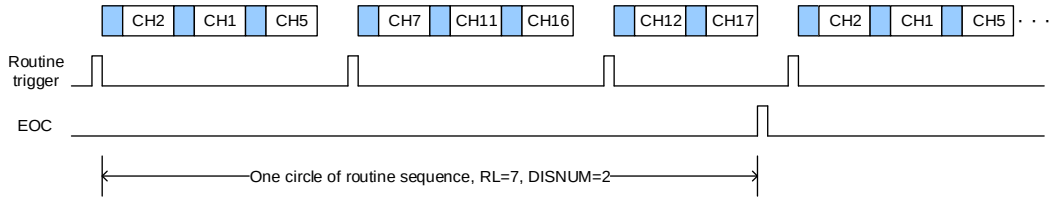


Discontinuous operation mode

For routine sequence, the discontinuous operation mode will be enabled when DISRC bit in the ADC_CTL0 register is set. In this mode, the ADC performs a short sequence of n conversions (n does not exceed 8) which is a part of the conversions selected in the ADC_RSQ0~ADC_RSQ2 registers. The value of n is configured by the DISNUM[2:0] bits in the ADC_CTL0 register. When the corresponding software trigger or TRIGSEL trigger is active, the ADC samples and converts the next n channels configured in the ADC_RSQ0~ADC_RSQ2 registers until all the channels of routine sequence are done. The EOC will be set after every circle of the routine sequence. An interrupt will be generated if the

EOCIE bit is set.

Figure 14-6. Discontinuous operation mode



Software procedure for discontinuous operation mode on a routine sequence:

1. Set the DISRC bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register.
2. Configure DISNUM[2:0] bits in the ADC_CTL0 register.
3. Configure ADC_RSQx and ADC_SAMPTx registers.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate a TRIGSEL trigger for the routine sequence.
7. Repeat step6 if in need.
8. Wait the EOC flag to be set.
9. Clear the EOC flag by writing 0 to it.

14.4.6. Conversion result threshold monitor

Analog watchdog 0

The analog watchdog 0 is enabled when the RWD0EN bit in the ADC_CTL0 register is set for routine sequence. This function is used to monitor whether the conversion result exceeds the thresholds. The WDE0 bit in ADC_STAT register will be set if the conversion result exceeds the thresholds. An interrupt will be generated if the WDE0IE bit is set. The ADC_WDHT0 and ADC_WDLT0 registers are used to specify the high and low threshold. The comparison is done before the alignment, so the threshold value is independent of the alignment, which is specified by the DAL bit in the ADC_CTL1 register. One or more channels, which are select by the RWD0EN, WD0SC and WD0CHSEL [4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog 0.

Analog watchdog 1

The analog watchdog 1 are more flexible, and can configure the watchdog function of single or serveral channels.

The analog watchdog 1 function can be enabled by configuring the corresponding bits in the AWD1CS [17: 0] bits in the ADC_WD1SR register. The high/low threshold of the analog watchdog 1 can be configured in the ADC_WDT1 register.

14.4.7. Data storage mode

The alignment of data stored after conversion can be specified by DAL bit in the ADC_CTL1 register.

Figure 14-7. Data storage mode of 12-bit resolution

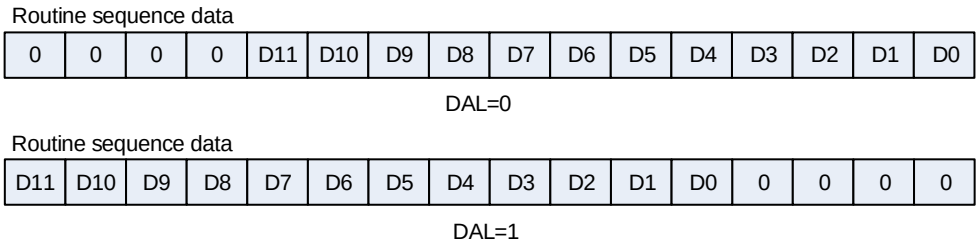


Figure 14-8. Data storage mode of 10-bit resolution

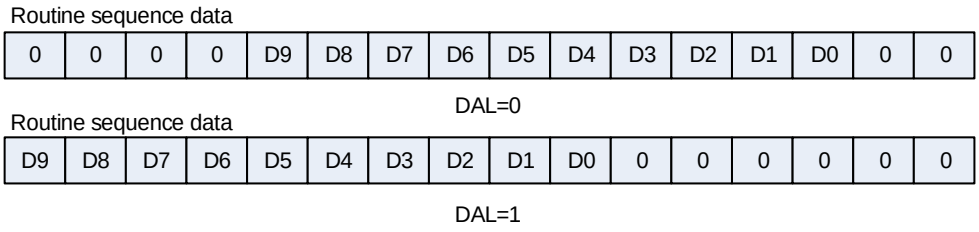
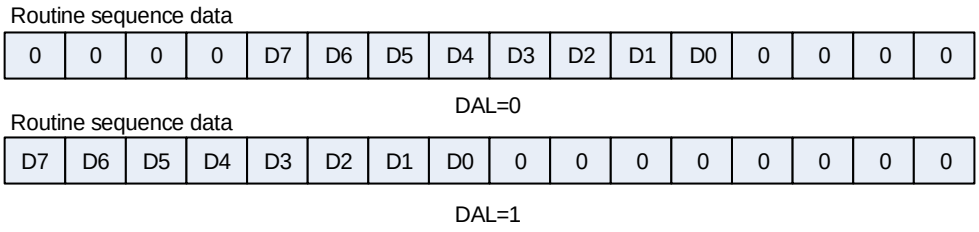
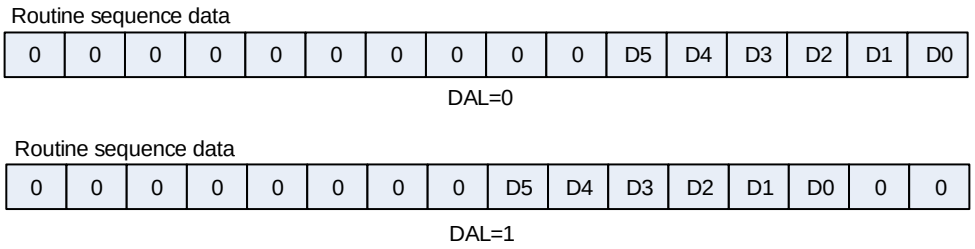


Figure 14-9. Data storage mode of 8-bit resolution



6-bit resolution data alignment is different from 12-bit/10-bit/8-bit resolution data alignment, shown as ***Figure 14-10. Data storage mode of 6-bit resolution***.

Figure 14-10. Data storage mode of 6-bit resolution



14.4.8. Sample time configuration

The number of CK_ADC cycles which is used to sample the input voltage can be specified by the SPTn[2:0] bits in the ADC_SAMPT0 and ADC_SAMPT1 registers. A different sample time can be specified for each channel. For 12-bits resolution, the total sampling and

conversion time is “sampling time + 12.5” CK_ADC cycles.

For example:

CK_ADC = 15MHz and sample time is 2.5 cycles, the total time is “2.5+12.5” CK_ADC cycles, that means 1us.

14.4.9. External trigger configuration

The conversion of routine sequence can be triggered by rising edge of TRIGSEL or SWRCST. The trigger source of routine sequence is controlled by the ETSRC bit in the ADC_CTL1 register.

Table 14-3. External trigger source for routine sequence

ETSRC	Trigger Source	Trigger Type
0	TRIGSEL	Signal from TRIGSEL
1	SWICST	Software trigger

14.4.10. DMA request

The DMA request, which is enabled by the DMA bit of ADC_CTL1 register, is used to transfer data of routine sequence for conversion of more than one channel. The ADC generates a DMA request at the end of conversion of a routine channel. When this request is received, the DMA will transfer the converted data from the ADC_RDATA register to the destination location which is specified by the user.

Note: ADC1 has no DMA request.

14.4.11. ADC internal channels

When the TSVEN bit in ADC_CTL1 register is set, the temperature sensor channel (ADC_IN16) is enabled. The temperature sensor can be used to measure the ambient temperature of the device. The sensor output voltage can be converted into a digital value by ADC. The sampling time for the temperature sensor is recommended to be set to at least t_{s_temp} μs (please refer to the datasheet). When this sensor is not in use, it can be put in power down mode by resetting the TSVEN bit.

The output voltage of the temperature sensor changes linearly with temperature. Due to the diversification of production process, the deviation of the temperature change curve is different on chip to chip (refer to the device datasheet for more information).

To use the temperature sensor:

1. Configure the ADC clock (not greater than 5MHz).
2. Configure the conversion sequence (ADC0_IN16) and the sampling time (t_{s_temp} μs) for the channel.
3. Enable the temperature sensor by setting the TSVEN bit in the ADC control register 1 (ADC_CTL1).
4. Start the ADC conversion by setting the ADCON bit or by the triggers.

5. Read the internal temperature sensor output voltage ($V_{temperature}$), and get the temperature using the following formula ***(14-1)***:

$$\text{Temperature (°C)} = \frac{V_{temperature} - V_{30}}{\text{Avg_Slope}} + 30 \quad (14\text{-}1)$$

$V_{temperature}$: The output voltage of temperature sensor.

V_{30}: Internal temperature sensor output voltage at 30°C. The ADC conversion results of the temperature sensor corresponding to 30°C were calibrated during chip production (VDDA=5.0V). The factory-calibrated offset value is stored in the read-only area of flash, and the typical value please refer to the datasheet.

Avg_Slope: Average slope for curve between temperature vs. internal temperature sensor output voltage, the typical value please refer to the datasheet.

Note:

1) After the temperature sensor is enabled, it is necessary to wait for at least 3 ADC sampling cycles before the ADC conversion code value is considered valid, and the first 3 conversion data should be discarded;
2) The sampling accuracy of temperature sensor can be improved by means of hardware on-chip over-sampling or software averaging. If higher precision temperature is required, it must be sampled several times (more than 50 times is recommended) for average.

When the INREFEN bit in ADC_CTL1 register is set, the V_{REFINT} channel (ADC0_IN17) is enabled. The internal reference voltage (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and Comparators. V_{REFINT} is internally connected to the ADC0_IN17 input channel.

14.4.12. Programmable resolution (DRES)

The resolution is configured by programming the DRES[1:0] bits in the ADC_OVSAMPCTL register. For applications that do not require high data accuracy, lower resolution allows faster conversion time. The DRES[1:0] bits must only be changed when the ADCON bit is reset. The result of the conversion is always 12 bits wide and any unused LSB bits are read as zeroes. Lower resolution reduces the conversion time needed for the successive approximation steps as shown in ***Table 14-4. tCONV timings depending on resolution***.

Table 14-4. t_{CONV} timings depending on resolution

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=15MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=15MHz
12	12.5	833 ns	2.5	15	1000 ns
10	10.5	700 ns	2.5	13	867 ns
8	8.5	567ns	2.5	11	733 ns
6	6.5	433 ns	2.5	9	600 ns

14.4.13. On-chip hardware oversampling

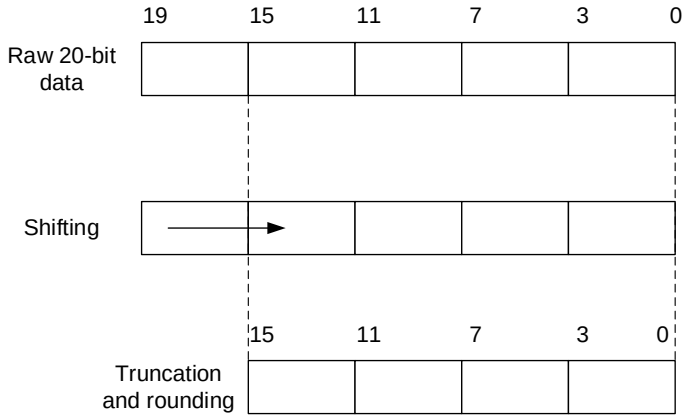
The on-chip hardware oversampling circuit performs data preprocessing to offload the CPU. It can handle multiple conversions and average them into a single data with increased data width, up to 16-bit. It provides a result with the following form, where N and M can be adjusted, and $D_{out}(n)$ is the n-th output digital signal of the ADC:

$$Result=\frac{1}{M}*\sum_{n=0}^{N-1} D_{out}(n) \quad (14\text{-}2)$$

The on-chip hardware oversampling circuit performs the following functions: summing and bit right shifting. The oversampling ratio N is defined by the OVSR[2:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to 256x. The division coefficient M means bit right shifting up to 8-bit. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

Summation units can produce up to 20 bits (256 x 12-bit), which is first shifted right. The upper bits of the result are then truncated, keeping only the 16 least significant bits rounded to the nearest value using the least significant bits left apart by the shifting, before being finally transferred into the data register, as shown in ***Figure 14-11. 20-bit to 16-bit result truncation***.

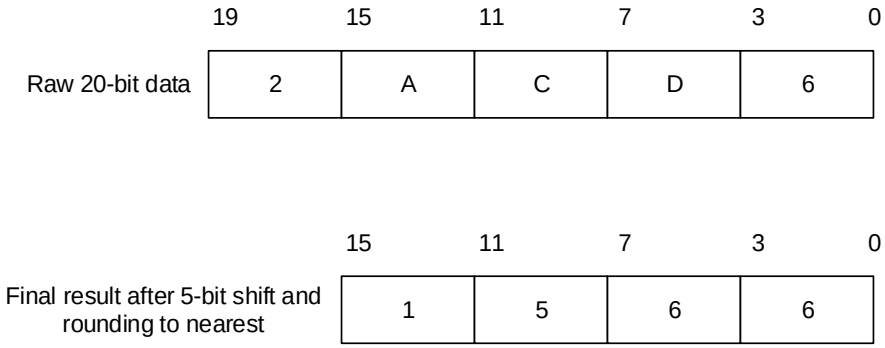
Figure 14-11. 20-bit to 16-bit result truncation



Note: If the intermediate result after the shifting exceeds 16 bits, the upper bits of the result are simply truncated.

Figure 14-12. Numerical example with 5-bits shift and rounding shows a numerical example of the processing, from a raw 20-bit accumulated data to the final 16-bit result.

Figure 14-12. Numerical example with 5-bits shift and rounding



The ***Table 14-5. Maximum output results for N and M (Grayed values indicates truncation)*** below gives the data format for the various N and M combination, for a raw conversion data equal to 0xFFF.

Table 14-5. Maximum output results for N and M (Grayed values indicates truncation)

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

The conversion timings in oversampled mode do not change compared to standard conversion mode: the sampling time remains equal throughout the oversampling sequence. New data is supplied every N conversions, and the equivalent delay is equal to:

$$N*t_{ADC}=N*(t_{SMPL}+t_{CONV}) \quad (14\text{-}3)$$

14.5. ADC sync mode

In devices with more than one ADC, the ADC sync mode can be used.

In ADC sync mode, the conversion starts alternately or simultaneously triggered by ADC0 to ADC1, according to the sync mode configured by the SYNCM[3:0] bits in ADC0_CTL0 register.

In sync mode, when the conversion is configured to be triggered by an external event, the ADC1 must be configured as triggered by the software. However, the external trigger must be enabled for ADC0 and ADC1. The converted data of routine channel is stored in the ADC0 routine data register (ADC0_RDATA).

The modes in ***Table 14-6. ADC sync mode table*** can be configured.

In ADC sync mode, the DMA bit must be set even if it is not used. The converted data of ADC1 routine channel can be read from the ADC0 routine data register (ADC0_RDATA).

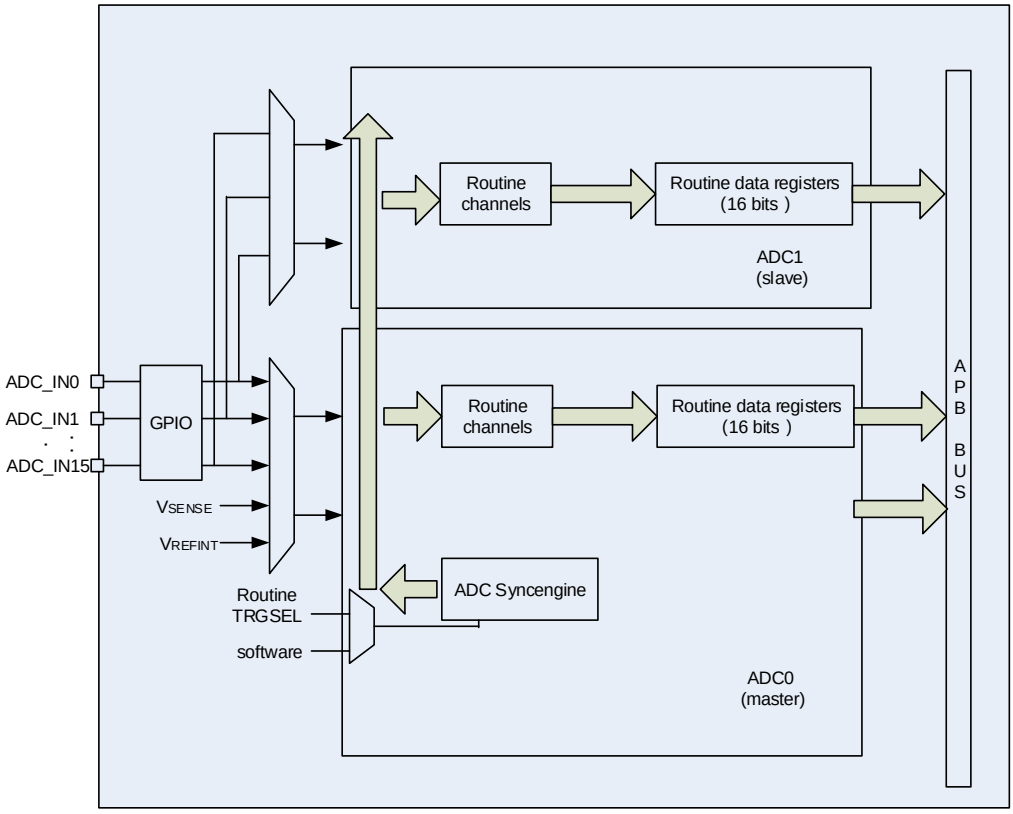
Table 14-6. ADC sync mode table

SYNCM[3:0]	Mode
0000	Free mode
0110	ADC0 and ADC1 work in routine parallel mode
0111	ADC0 and ADC1 work in routine follow-up fast mode
1000	ADC0 and ADC1 work in routine follow-up slow mode

When the ADCs are in a sync mode other than free mode, they shold be configured to free mode before being configured to another sync mode.

The ADC sync scheme is shown in ***Figure 14-13. ADC sync block diagram***.

Figure 14-13. ADC sync block diagram



14.5.1. Free mode

In this mode, each ADC works independently and does not interfere with each other.

14.5.2. Routine parallel mode

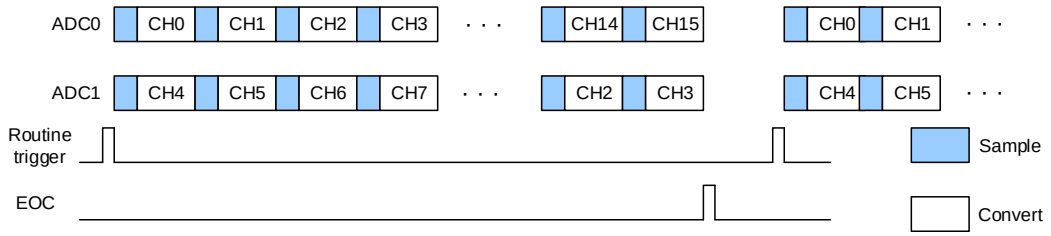
The routine parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4b'0110. In the routine parallel mode. All of the ADCs convert the routine sequence

parallelly at the selected external trigger of ADC0. The trigger is selected by configuring the ETSRC bit in the ADC_CTL1 register of ADC0.

EOC interrupts (if enabled on the ADC interfaces) are generated at the end of conversion events. The behavior of routine parallel mode is shown in the ***Figure 14-14. Routine parallel mode on 16 channels***.

Also A 32-bit DMA is used, which transfers ADC_RDATA 32-bit register (the ADC_RDATA 32-bit register containing the ADC1 converted data in the upper half-word and the ADC0 converted data in the lower half-word) to SRAM.

Figure 14-14. Routine parallel mode on 16 channels



Note:

1. Do not convert the same channel on two ADCs at a given time (no overlapping sampling times for the ADCs when converting the same channel).
2. Make sure to trigger the ADCs when none of them is converting (do not trigger ADC0 when some of the conversions are not finished).

14.5.3. Routine follow-up fast mode

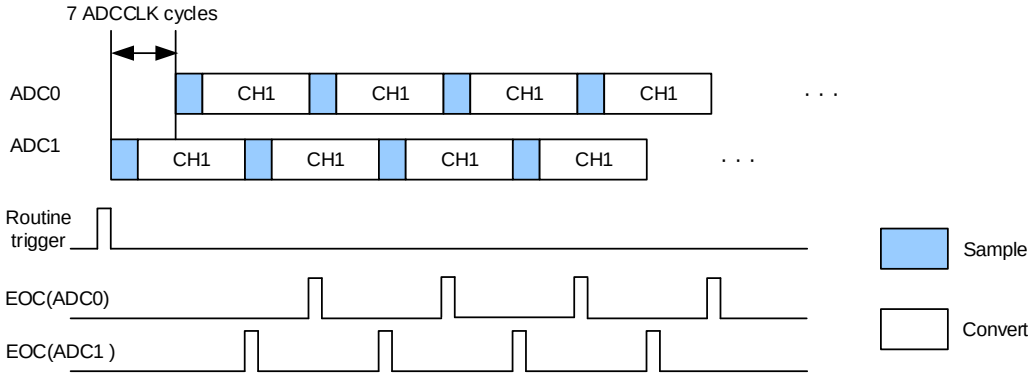
The routine follow-up fast mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0111. In the follow-up fast mode, ADC1 converts the routine sequence at the selected external trigger. The trigger is selected by configuring the ETSRC bit in the ADC_CTL1 register of ADC0. After 7 ADC clock cycles, ADC0 converts the routine sequence. The routine sequence in above descriptions only includes one routine channel.

If the CNT bit in ADC_CTL1 register is set, the selected routine channels are continuously converted. EOC interrupts (if enabled on the ADC interfaces) are generated at the end of conversion events. The behavior of routine follow-up mode is shown in the ***Figure 14-15. Routine follow-up fast mode on 1 channel in continuous operation mode***.

If EOCIE bit is set, an EOC interrupt is generated by ADC0 at the end of conversion event on ADC0. Also a 32-bit DMA can be used, which transfers ADC_RDATA 32-bit register (the ADC_RDATA 32-bit register containing the ADC1 converted data in the upper half-word and the ADC0 converted data in the lower half-word) to SRAM.

Note: The maximum sampling time allowed is < 7 CK_ADC cycles to avoid the overlap between ADC0 and ADC1 sampling phases in the event that they convert the same channel.

Figure 14-15. Routine follow-up fast mode on 1 channel in continuous operation mode



14.5.4. Routine follow-up slow mode

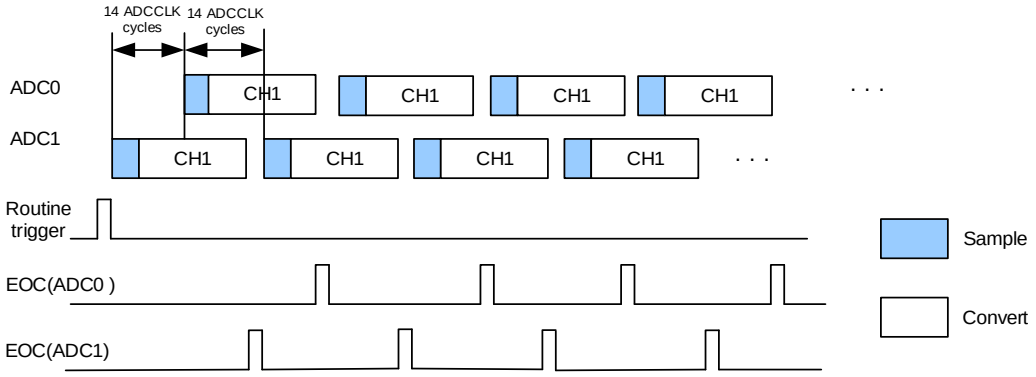
The routine follow-up slow mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b1000. In the follow-up slow mode, ADC1 converts the routine sequence at the selected external trigger. The trigger is selected by configuring the ETSRC bit in the ADC_CTL1 register of ADC0. After 14 ADC clock cycles, ADC0 converts the routine sequence. The routine sequence in above descriptions only includes one routine channel.

Continuous mode can't be used in this mode, because it continuously converts the routine channel. The behavior of follow-up slow mode shows in the ***Figure 14-16. Routine follow-up slow mode on 1 channel***.

If EOCIE bit is set, an EOC interrupt is generated by ADC0 at the end of conversion event on ADC0. Also a 32-bit DMA can be used, which transfers ADC_RDATA 32-bit register (the ADC_RDATA 32-bit register containing the ADC1 converted data in the upper half-word and the ADC0 converted data in the lower half-word) to SRAM.

Note: The maximum sampling time allowed is <14 CK_ADC cycles to avoid the overlap between ADC0 and ADC1 sampling phases in the event that they convert the same channel.

Figure 14-16. Routine follow-up slow mode on 1 channel



14.6. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for routine sequence.
- The analog watchdog 0/1 event.

Separate interrupt enable bits are available for flexibility.

The interrupts of ADC0 and ADC1 are mapped into the same interrupt vector IRQ18.

14.7. Register definition

ADC0 base address: 0x4001 2400

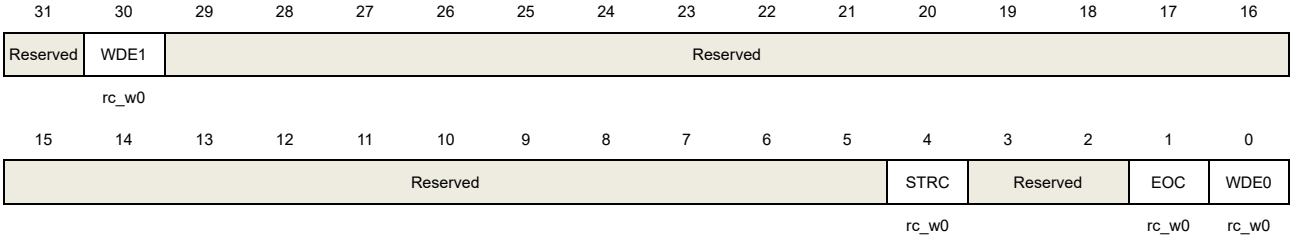
ADC1 base address: 0x4001 2800

14.7.1. Status register (ADC_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	WDE1	Analog watchdog 1 event flag 0: Analog watchdog 1 event is not happened 1: Analog watchdog 1 event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WDT1 register. Cleared by software writing 0 to it.
29:5	Reserved	Must be kept at reset value.
4	STRC	Start flag of routine sequence 0: Routine sequence conversion is not started 1: Routine sequence conversion is started Set by hardware when routine sequence conversion starts. Cleared by software writing 0 to it.
31:5	Reserved	Must be kept at reset value.
1	EOC	End flag of sequence conversion 0: No end of sequence conversion 1: End of sequence conversion Set by hardware at the end of a sequence conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.
0	WDE0	Analog watchdog 0 event flag 0: Analog watchdog 0 event is not happened

1: Analog watchdog 0 event is happening

Set by hardware when the converted voltage crosses the values programmed in the ADC_WDLT0 and ADC_WDHT0 registers.

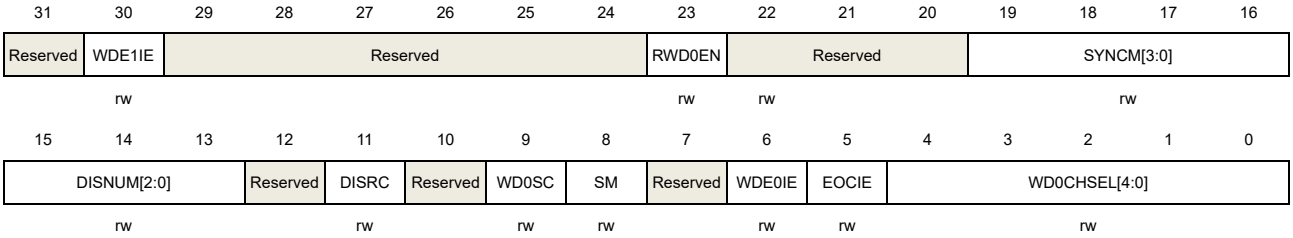
Cleared by software writing 0 to it.

14.7.2. Control register 0 (ADC_CTL0)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	WDE1IE	Interrupt enable for WDE1 0: WDE1 interrupt disable 1: WDE1 interrupt enable
29:24	Reserved	Must be kept at reset value.
23	RWD0EN	Routine channel analog watchdog 0 enable 0: Routine channel analog watchdog 0 disable 1: Routine channel analog watchdog 0 enable
22:20	Reserved	Must be kept at reset value.
19:16	SYNCM[3:0]	Sync mode selection When ADC sync mode is enabled these bits should be set to 4b'0000 firstly before change to another value. 4b'0000: Free mode. All the ADCs work independently. 4b'0110: ADC0 and ADC1 work in routine parallel mode only 4b'0111: ADC0 and ADC1 work in routine follow-up fast mode only 4b'1000: ADC0 and ADC1 work in routine follow-up slow mode only All other values are reserved. **Note**: These bits are only used in ADC0.
15:13	DISNUM[2:0]	Number of conversions in discontinuous mode The number of channels to be converted after a trigger will be DISNUM+1 in routine sequence.

12	Reserved	Must be kept at reset value.
11	DISRC	Discontinuous mode on routine sequence 0: Discontinuous operation mode on routine sequence disable 1: Discontinuous operation mode on routine sequence enable
10	Reserved	Must be kept at reset value.
9	WD0SC	When in scan mode, analog watchdog 0 is effective on a single channel. 0: All channels have analog watchdog 0 function 1: A single channel has analog watchdog 0 function
8	SM	Scan mode 0: Scan operation mode disable 1: Scan operation mode enable
7	Reserved	Must be kept at reset value.
6	WDE0IE	Interrupt enable for WDE0 0: Interrupt disable 1: Interrupt enable
5	EOCIE	Interrupt enable for EOC 0: Interrupt disable 1: Interrupt enable
4:0	WD0CHSEL[4:0]	Analog watchdog 0 channel select 00000: ADC channel 0 00001: ADC channel 1 00010: ADC channel 2 00011: ADC channel 3 00100: ADC channel 4 00101: ADC channel 5 00110: ADC channel 6 00111: ADC channel 7 01000: ADC channel 8 01001: ADC channel 9 01010: ADC channel 10 01011: ADC channel 11 01100: ADC channel 12 01101: ADC channel 13 01110: ADC channel 14 01111: ADC channel15 10000: ADC channel16 10001: ADC channel17 Other values are reserved. **Note**: 1. ADC0 analog inputs Channel16 and Channel17 are internally connected to

the temperature sensor, and to V_{REFINT} inputs.

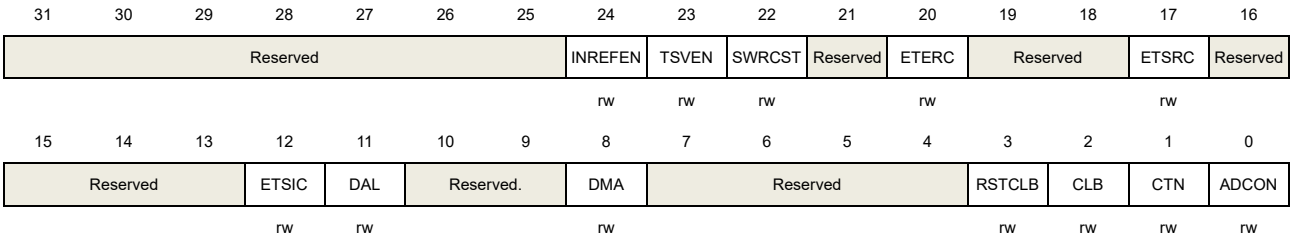
2. ADC1 analog inputs Channel16, and Channel17 are internally connected to V_{SSA}.

14.7.3. Control register 1 (ADC_CTL1)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	INREFEN	Channel 17 (internal reference voltage) enable of ADC0. 0: Channel 17 of ADC0 disable 1: Channel 17 of ADC0 enable
23	TSVEN	Channel 16 (temperature sensor) enable of ADC0. 0: Channel 16 of ADC0 disable 1: Channel 16 of ADC0 enable
22	SWRCST	Software start on routine sequence. Setting 1 on this bit starts a conversion of a routine sequence if ETSRC is 1. It is set by software and cleared by software or by hardware immediately after the conversion starts.
21	Reserved	Must be kept at reset value.
20	ETERC	External trigger enable for routine sequence 0: External trigger for routine sequence disable 1: External trigger for routine sequence enable
19:18	Reserved	Must be kept at reset value.
17	ETSRC	External trigger selection for routine sequence 0: TRIGSEL 1: SWRCST
16:12	Reserved	Must be kept at reset value.
11	DAL	Data alignment 0: LSB alignment

		1: MSB alignment
10:9	Reserved	Must be kept at reset value.
8	DMA	DMA request enable 0: DMA request disable 1: DMA request enable **Note**: This bit is only used in ADC0.
7:4	Reserved	Must be kept at reset value.
3	RSTCLB	Reset calibration registers This bit is set by software and cleared by hardware after the calibration registers are initialized. 0: Calibration registers initialize done 1: Initialize calibration registers start
2	CLB	ADC calibration 0: Calibration done 1: Calibration start
1	CTN	Continuous mode 0: Continuous operation mode disable 1: Continuous operation mode enable
0	ADCON	ADC ON. The ADC will be wake up when this bit is changed from low to high and take a stabilization time (t_{SU}). When this bit is high and “1” is written to it with other bits of this register unchanged, the conversion will start. For power saving, when this bit is reset, the analog submodule will be put into power down mode. 0: ADC disable and power down 1: ADC enable

14.7.4. Sample time register 0 (ADC_SAMPT0)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
								rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]		
rw	rw			rw			rw			rw			rw		

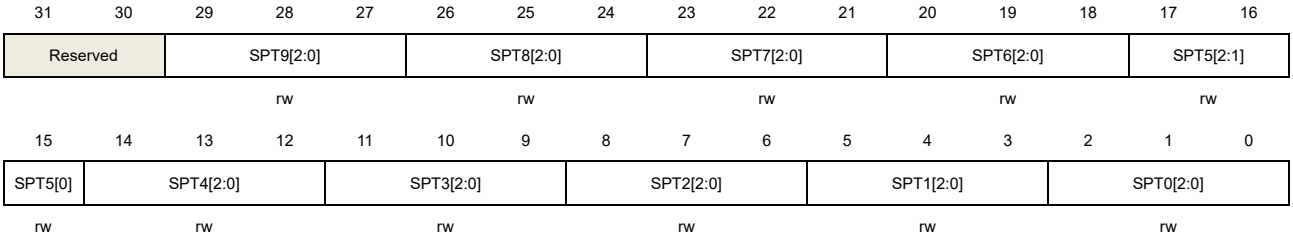
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:21	SPT17[2:0]	Refer to SPT10[2:0] description
20:18	SPT16[2:0]	Refer to SPT10[2:0] description
17:15	SPT15[2:0]	Refer to SPT10[2:0] description
14:12	SPT14[2:0]	Refer to SPT10[2:0] description
11:9	SPT13[2:0]	Refer to SPT10[2:0] description
8:6	SPT12[2:0]	Refer to SPT10[2:0] description
5:3	SPT11[2:0]	Refer to SPT10[2:0] description
2:0	SPT10[2:0]	Channel sample time 000: Channel sampling time is 2.5 cycles 001: Channel sampling time is 14.5 cycles 010: Channel sampling time is 27.5 cycles 011: Channel sampling time is 55.5 cycles 100: Channel sampling time is 83.5 cycles 101: Channel sampling time is 111.5 cycles 110: Channel sampling time is 143.5 cycles 111: Channel sampling time is 479.5 cycles

14.7.5. Sample time register 1 (ADC_SAMPT1)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



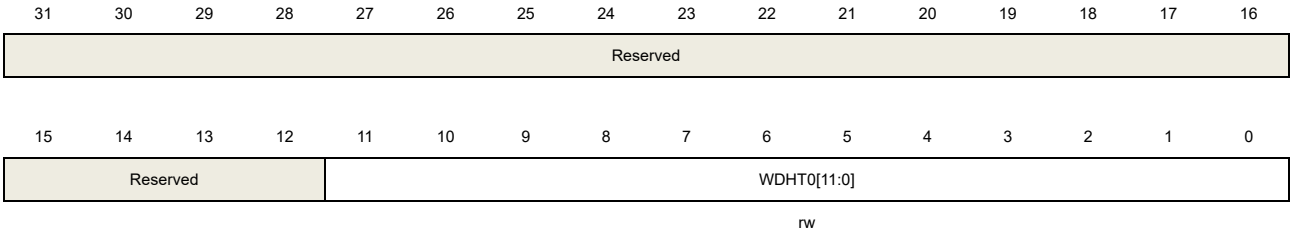
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:27	SPT9[2:0]	Refer to SPT0[2:0] description
26:24	SPT8[2:0]	Refer to SPT0[2:0] description
23:21	SPT7[2:0]	Refer to SPT0[2:0] description

20:18	SPT6[2:0]	Refer to SPT0[2:0] description
17:15	SPT5[2:0]	Refer to SPT0[2:0] description
14:12	SPT4[2:0]	Refer to SPT0[2:0] description
11:9	SPT3[2:0]	Refer to SPT0[2:0] description
8:6	SPT2[2:0]	Refer to SPT0[2:0] description
5:3	SPT1[2:0]	Refer to SPT0[2:0] description
2:0	SPT0[2:0]	Channel sample time 000: Channel sampling time is 2.5 cycles 001: Channel sampling time is 14.5 cycles 010: Channel sampling time is 27.5 cycles 011: Channel sampling time is 55.5 cycles 100: Channel sampling time is 83.5 cycles 101: Channel sampling time is 111.5 cycles 110: Channel sampling time is 143.5 cycles 111: Channel sampling time is 479.5 cycles

14.7.6. Watchdog high threshold register 0 (ADC_WDHT0)

Address offset: 0x24
Reset value: 0x0000 0FFF

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	WDHT0[11:0]	High threshold for analog watchdog 0 These bits define the high threshold for the analog watchdog 0.

14.7.7. Watchdog low threshold register 0 (ADC_WDLT0)

Address offset: 0x28
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WDLT0[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WDLT0[11:0]	Low threshold for analog watchdog 0 These bits define the low threshold for the analog watchdog 0.

14.7.8. Routine sequence register 0 (ADC_RSQ0)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RL[3:0]				RSQ15[4:1]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]	RSQ14[4:0]					RSQ13[4:0]					RSQ12[4:0]				
rw	rw					rw					rw				

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	RL[3:0]	Routine sequence length. The total number of conversion in routine sequence equals to RL[3:0]+1.
19:15	RSQ15[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ14[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ13[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ12[4:0]	Refer to RSQ0[4:0] description

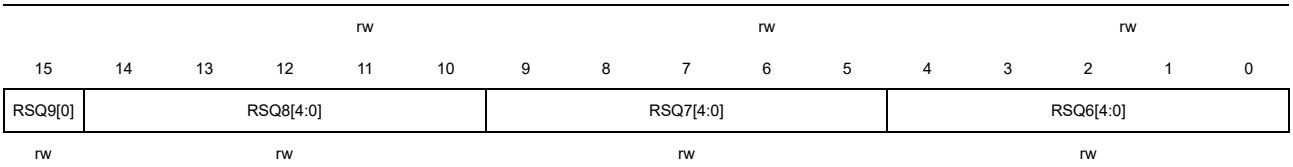
14.7.9. Routine sequence register 1 (ADC_RSQ1)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		RSQ11[4:0]						RSQ10[4:0]				RSQ9[4:1]			



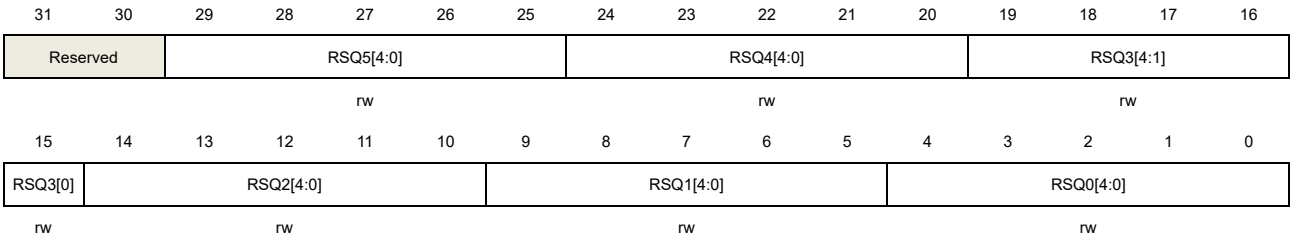
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ11[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ10[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ9[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ8[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ7[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ6[4:0]	Refer to RSQ0[4:0] description

14.7.10. Routine sequence register 2 (ADC_RSQ2)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



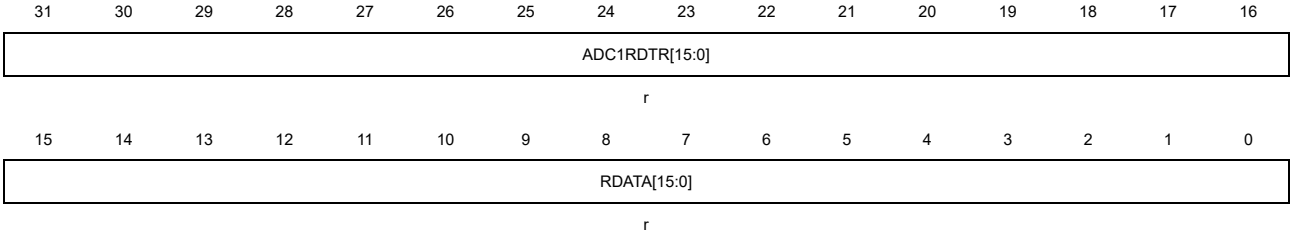
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ5[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ4[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ3[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ2[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ1[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ0[4:0]	The channel number (0..17) is written to these bits to select a channel as the nth conversion in the routine sequence.

14.7.11. Routine data register (ADC_RDATA)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



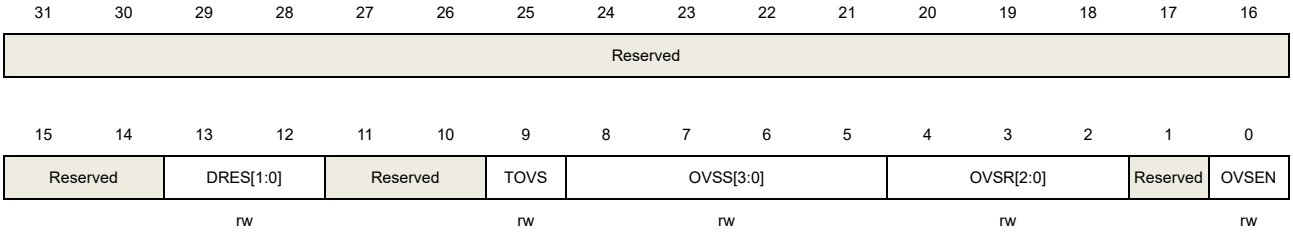
Bits	Fields	Descriptions
31:16	ADC1RDTR[15:0]	ADC1 routine channel data In ADC0: In sync mode, these bits contain the routine data of ADC1. In ADC1: These bits are reserved.
15:0	RDATA[15:0]	Routine channel data These bits contain routine channel conversion value, which is read only.

14.7.12. Oversample control register (ADC_OVSAMPCTL)

Address offset: 0x80

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:12	DRES[1:0]	ADC resolution 00: 12-bit 01: 10-bit 10: 8-bit 11: 6-bit
11:10	Reserved	Must be kept at reset value.
9	TOVS	Triggered Oversampling

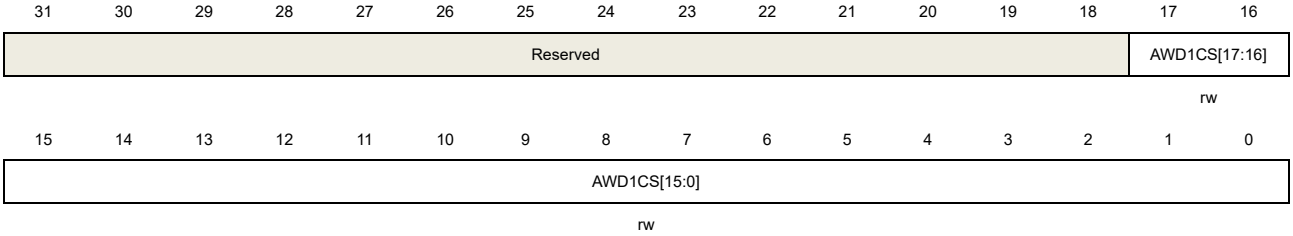
		This bit is set and cleared by software. 0: All oversampled conversions for a channel are done consecutively after a trigger. 1: Each conversion needs a trigger for an oversampled channel and the number of triggers is determined by the oversampling ratio (OVSR[2:0]). **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
8:5	OVSS[3:0]	Oversampling shift This bit is set and cleared by software. 0000: No shift 0001: Shift 1-bit 0010: Shift 2-bits 0011: Shift 3-bits 0100: Shift 4-bits 0101: Shift 5-bits 0110: Shift 6-bits 0111: Shift 7-bits 1000: Shift 8-bits Other values are reserved. **Note**: The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
4:2	OVSR[2:0]	Oversampling ratio This bit filed defines the number of oversampling ratio. 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x **Note**: The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
1	Reserved	Must be kept at reset value.
0	OVSEN	Oversampler enable This bit is set and cleared by software. 0: Oversampler disabled 1: Oversampler enabled **Note**: The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).

14.7.13. Watchdog 1 channel selection register (ADC_WD1SR)

Address offset: 0xA0

Reset value: 0x00000000

This register has to be accessed by word (32-bit).



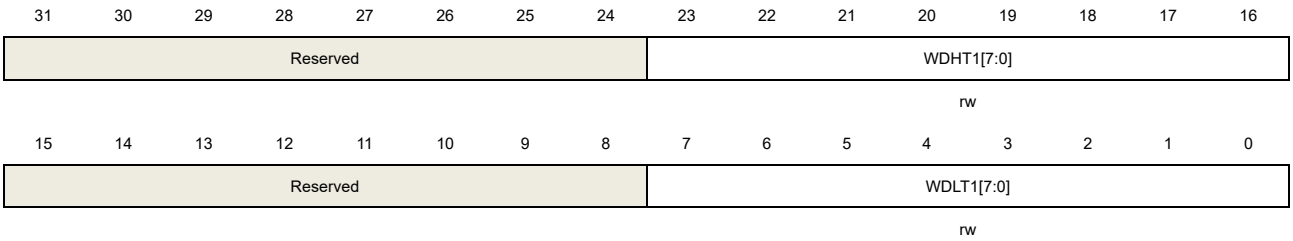
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:0	AWD1CS[17:0]	Analog watchdog 1 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 1. AWD1CS[n] = 0: ADC analog input channel n is not monitored by analog watchdog 1. AWD1CS[n] = 1: ADC analog input channel n is monitored by analog watchdog 1. When AWD1CS[17:0] = 000..0, the analog watchdog 1 is disabled. **Note:** 1) The channels selected by AWD1CS must be also selected into the ADC_RSQn registers. 2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

14.7.14. Watchdog threshold register 1 (ADC_WDT1)

Address offset: 0xA8

Reset value: 0x00FF 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.

23:16	WDHT1[7:0]	High threshold for analog watchdog 1 These bits define the high threshold for the analog watchdog 1. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).
15:8	Reserved	Must be kept at reset value.
7:0	WDLT1[7:0]	Low threshold for analog watchdog 1 These bits define the low threshold for the analog watchdog 1. **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

15. Digital-to-analog converter (DAC)

15.1. Overview

The Digital-to-analog converter converts 12-bit digital data to a voltage on the external pins. The digital data can be configured to 8-bit or 12-bit mode, left-aligned or right-aligned mode. DMA can be used to update the digital data on external triggers.

The output voltage can be optionally buffered for higher drive capability.

15.2. Characteristics

The main features of DAC are as follows:

- 8-bit or 12-bit resolution.
- Left or right data alignment.
- DMA capability for each channel and underrun function.
- Conversion update synchronously.
- Conversion triggered by external triggers.
- Configurable internal buffer.
- Extern voltage reference, V_{REFP}.
- Noise wave generation (LFSR noise mode and Triangle noise mode).

Figure 15-1. DAC block diagram and *Table 15-1. DAC I/O description* show the block diagram of DAC and the pin description of DAC, respectively.

Figure 15-1. DAC block diagram

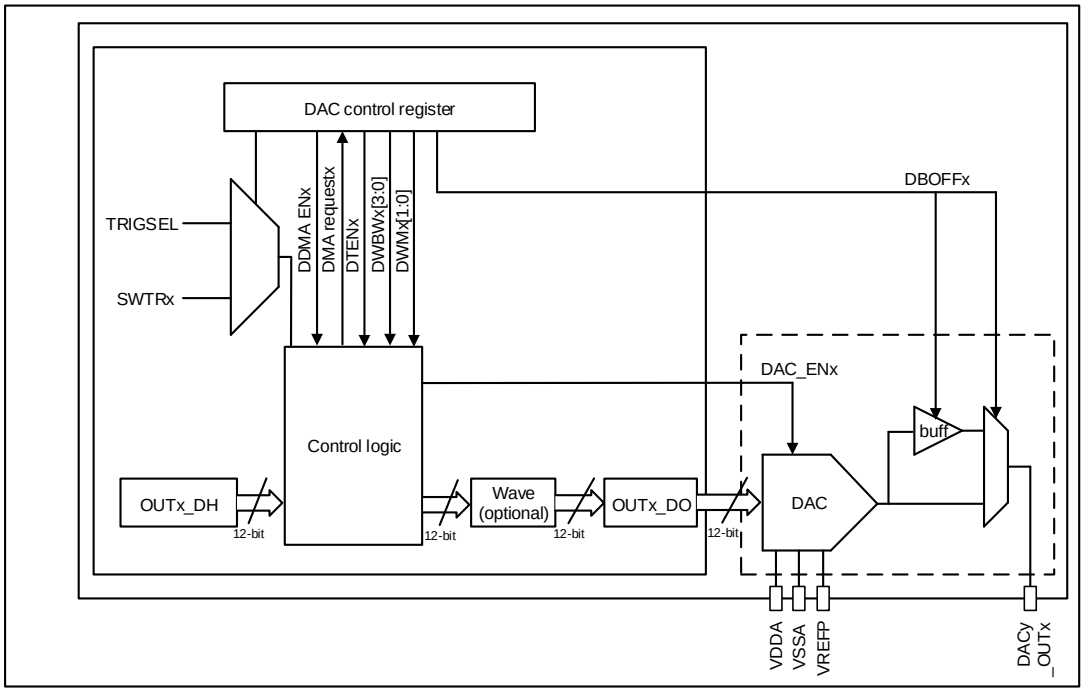


Table 15-1. DAC I/O description

Name	Description	Signal type
V_{DDA}	Analog power supply	Input, analog supply
V_{SSA}	Ground for analog power supply	Input, analog supply ground
V_{REFP}	Positive reference voltage of DAC	Input, analog positive reference
DACy_OUTx	DAC analog output	Analog output signal

The below table details the triggers and outputs of the DAC.

Table 15-2. DAC triggers and outputs summary

	DAC0
Channel	**Channel0**
DAC outputs connected to I / Os	PA7
DAC output buffer	•
DAC trigger signals from TRIGSEL	•
DAC software trigger	•

Note: The GPIO pin should be configured to analog mode before enable the DAC module.

15.3. Function overview

15.3.1. DAC enable

The DAC can be turned on by setting the DENx bit in the DAC_CTL0 register. A t_{WAKEUP} time is needed to startup the analog DAC submodule.

15.3.2. DAC output buffer

For reducing output impedance and driving external loads without an external operational amplifier, an output buffer is integrated inside each DAC module.

The output buffer, which is turned on by default to reduce the output impedance and improve the driving capability, can be turned off by setting the DBOFFx bit in the DAC_CTL0 register.

When DAC output buffer turns off, DAC can connect to on chip peripherals (CMP) independently by setting the DDISCx bit in the DAC_CTL0 register.

15.3.3. DAC data configuration

The 12-bit DAC holding data (OUTx_DH) can be configured by writing any one of the DAC_OUTx_R12DH, DAC_OUTx_L12DH and DAC_OUTx_R8DH registers. When the data is loaded by DAC_OUTx_R8DH register, only the MSB 8 bits are configurable, the LSB 4 bits

are forced to 4’b0000.

15.3.4. DAC trigger

The DAC conversion can be triggered by software or rising edge of external trigger source. The DAC external trigger is enabled by setting the DTENx bit in the DAC_CTL0 register. The DAC external triggers are selected by the DTSELx bit in the DAC_CTL0 register, which is shown as ***Table 15-3. Triggers of DAC***.

Table 15-3. Triggers of DAC

DTSELx[1:0]	Trigger Source	Trigger Type
2b’00	TRIGSEL	Hardware trigger
2b’01	SWTR	Software trigger
2b’10	Reserved	Reserved
2b’11		

The external trigger is generated from the TRIGSEL, while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

15.3.5. DAC conversion

If the external trigger is enabled by setting the DTENx bit in DAC_CTL0 register, the DAC holding data is transferred to the DAC output data (DAC_OUTx_DO) register when the selected trigger event happened. When the external trigger is disabled, the transfer is performed automatically.

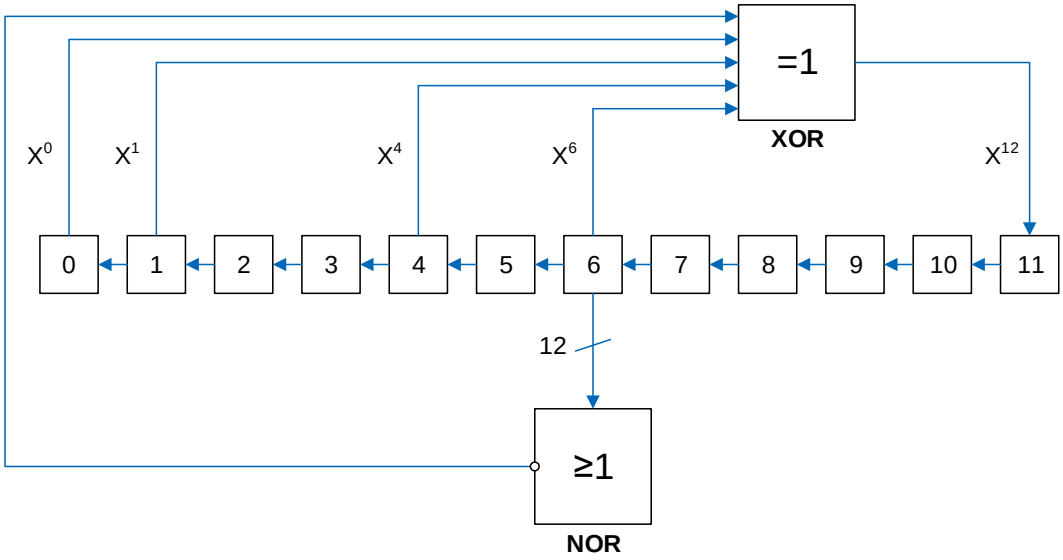
When the DAC holding data (OUTx_DH) is loaded into the DAC_OUTx_DO register, after the time $t_{SETTLING}$ which is determined by the analog output load and the power supply voltage, the analog output is valid.

15.3.6. DAC noise wave

There are two methods of adding noise wave to the DAC output data: LFSR noise wave mode and Triangle wave mode. The noise wave mode can be selected by the DWMx bit in the DAC_CTL0 register. The amplitude of the noise can be configured by the DAC noise wave bit width (DWBWx) bits in the DAC_CTL0 register.

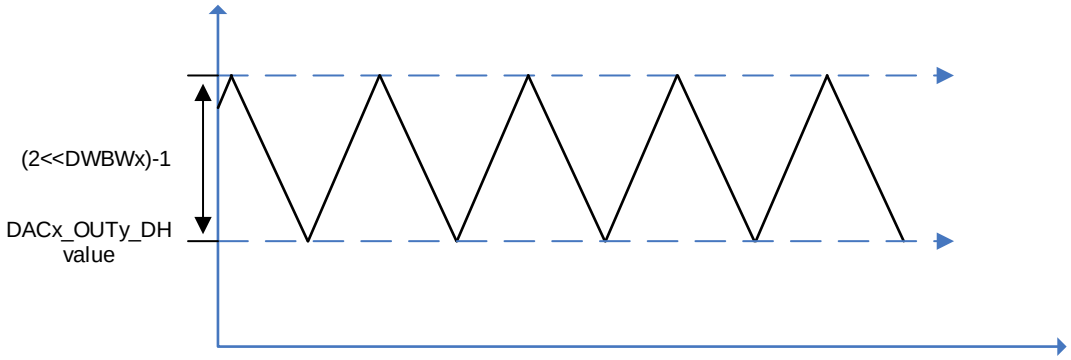
LFSR noise wave mode: there is a Linear Feedback Shift Register (LFSR) in the DAC control logic, it controls the LFSR noise signal which is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. When the configured DAC noise wave bit width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

Figure 15-2. DAC LFSR algorithm



Triangle noise mode: a triangle signal is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. The minimum value of the triangle signal is 0, while the maximum value of the triangle signal is (2 << DWBWx) - 1.

Figure 15-3. DAC triangle noise wave



15.3.7. DAC output voltage

The following equation determines the analog output voltage on the DAC pin.

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (15\text{-}1)$$

The digital input is linearly converted to an analog output voltage and its range is 0 to V_{REFP}.

15.3.8. DMA request

When the external trigger is enabled, the DMA request is enabled by setting the DDMAENx bit of the DAC_CTL0 register. A DMA request will be generated when an external hardware trigger (not a software trigger) occurs.

If the second external trigger arrives before confirming the previous request, the new request

will not be serviced, and an underrun error event occurs. The DDUDRx bit in the DAC_STAT0 register is set, an interrupt will be generated if the DDUDRIEx bit in the DAC_CTL0 register is set. The DMA request will be stalled until the DDUDRx bit is cleared.

15.4. Register definition

DAC0 base address: 0x4000 7400

15.4.1. DACx control register 0 (DAC_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DDISC0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		Reserved	DTSEL0[1:0]		DTEN0	DBOFF0	DEN0
	rw	rw	rw	rw				rw			rw		rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14	DDISC0	DACx_OUT0 connect GPIO selection 0: DACx_OUT0 is connected to the external pin and on chip peripherals (CMP). 1: DACx_OUT0 is connected to on chip peripherals (CMP) independently only if output buffer turns off (DBOFF=1). Otherwise it is connected to the external pin and to on chip peripherals (CMP).
13	DDUDRIE0	DACx_OUT0 DMA underrun interrupt enable 0: DACx_OUT0 DMA underrun interrupt disabled 1: DACx_OUT0 DMA underrun interrupt enabled
12	DDMAEN0	DACx_OUT0 DMA enable 0: DACx_OUT0 DMA mode disabled 1: DACx_OUT0 DMA mode enabled
11:8	DWBW0[3:0]	DACx_OUT0 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT0. These bits indicate that unmask LFSR bit [n-1, 0] in LFSR noise mode or the amplitude of the triangle is ((2<<(n-1))-1) in triangle noise mode, where n is the bit width of wave. 0000: The bit width of the wave signal is 1 0001: The bit width of the wave signal is 2 0010: The bit width of the wave signal is 3 0011: The bit width of the wave signal is 4 0100: The bit width of the wave signal is 5 0101: The bit width of the wave signal is 6 0110: The bit width of the wave signal is 7

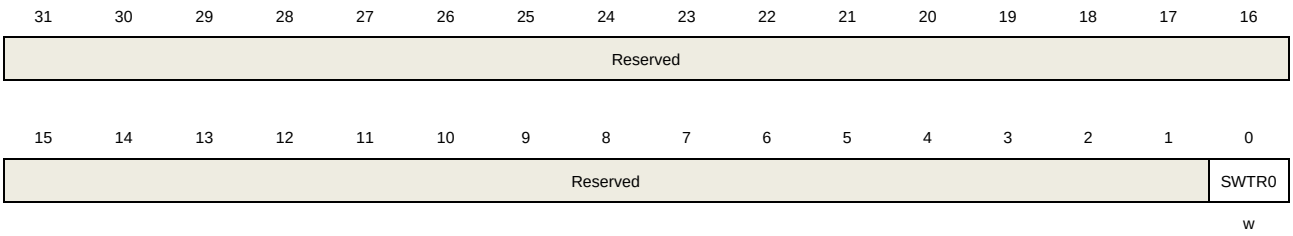
		0111: The bit width of the wave signal is 8 1000: The bit width of the wave signal is 9 1001: The bit width of the wave signal is 10 1010: The bit width of the wave signal is 11 ≥1011: The bit width of the wave signal is 12
7:6	DWM0[1:0]	DACx_OUT0 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT0 when external trigger of DACx_OUT0 is enabled (DTEN0=1). 00: Wave disabled 01: LFSR noise mode 1x: Triangle noise mode
5	Reserved	Must be kept at reset value
4:3	DTSEL0[1:0]	DACx_OUT0 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 00: EXTRIG (external trigger from TRIGSEL). 01: Software trigger. All other values: reserved.
2	DTEN0	DACx_OUT0 trigger enable 0: DACx_OUT0 trigger disabled 1: DACx_OUT0 trigger enabled
1	DBOFF0	DACx_OUT0 output buffer turn off 0: DACx_OUT0 output buffer turns on to reduce the output impedance and improve the driving capability 1: DACx_OUT0 output buffer turns off
0	DEN0	DACx_OUT0 enable 0: DACx_OUT0 disabled 1: DACx_OUT0 enabled

15.4.2. DACx software trigger register (DAC_SWT)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



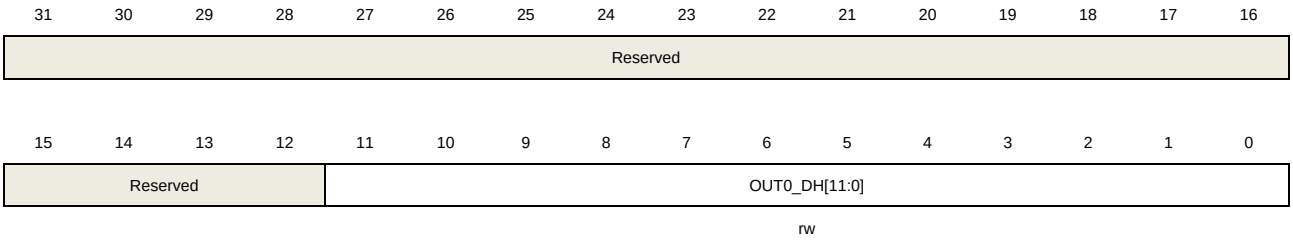
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SWTR0	DACx_OUT0 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled

15.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



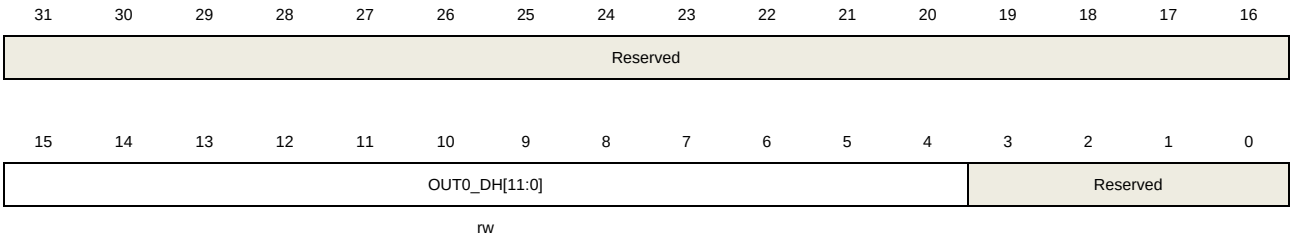
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT0.

15.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

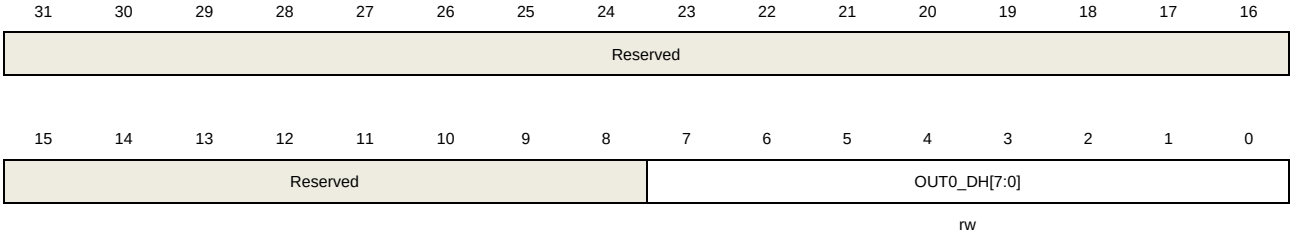
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

15.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



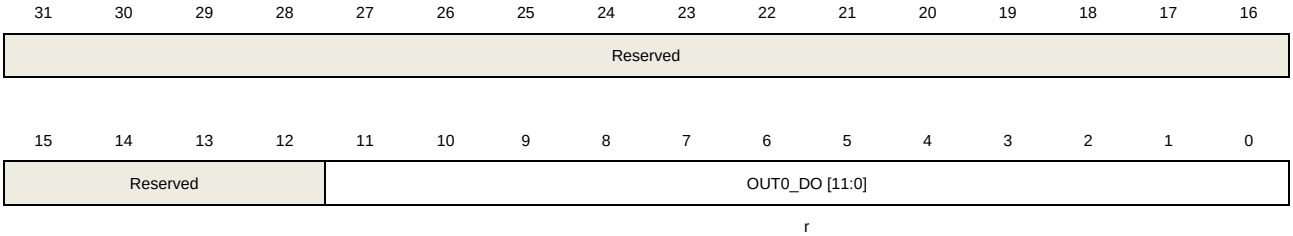
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

15.4.6. DACx_OUT0 data output register (DAC_OUT0_DO)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



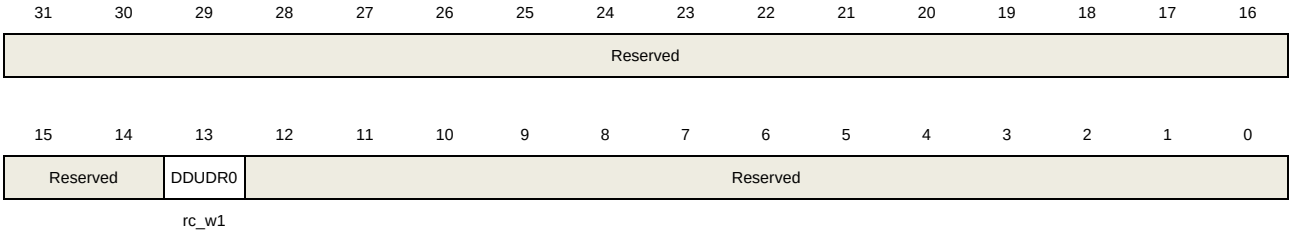
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DO [11:0]	DACx_OUT0 12-bit output data These bits, which are read only, storage the data that is being converted by DACx_OUT0.

15.4.7. DACx status register 0 (DAC_STAT0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	DDUDR0	DACx_OUT0 DMA underrun flag. This bit is set by hardware and cleared by software (by writing it to 1). 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is high than the DMA transfer).
12:0	Reserved	Must be kept at reset value.

16. Watchdog timer (WDGT)

The watchdog timer (WDGT) is a hardware timing circuitry that can be used to detect system failures due to software malfunctions. There are two watchdog timer peripherals in the chip: free watchdog timer (FWDGT) and window watchdog timer (WWDGT). They offer a combination of a high safety level, flexibility of use and timing accuracy. Both watchdog timers are offered to resolve malfunctions of software.

The watchdog timer will generate a reset when the internal counter reaches a given value. The watchdog timer counter can be stopped while the processor is in the debug mode.

16.1. Free watchdog timer (FWDGT)

16.1.1. Overview

The free watchdog timer (FWDGT) has free clock source (IRC40K). Thereupon the FWDGT can operate even if the main clock fails. It's suitable for the situation that requires an independent environment and lower timing accuracy.

The Free watchdog timer generate a reset when the internal down counter reaches 0 or the counter is refreshed when the value of the counter is greater than the window register value. The register write protection function in free watchdog can be enabled to prevent it from changing the configuration unexpectedly.

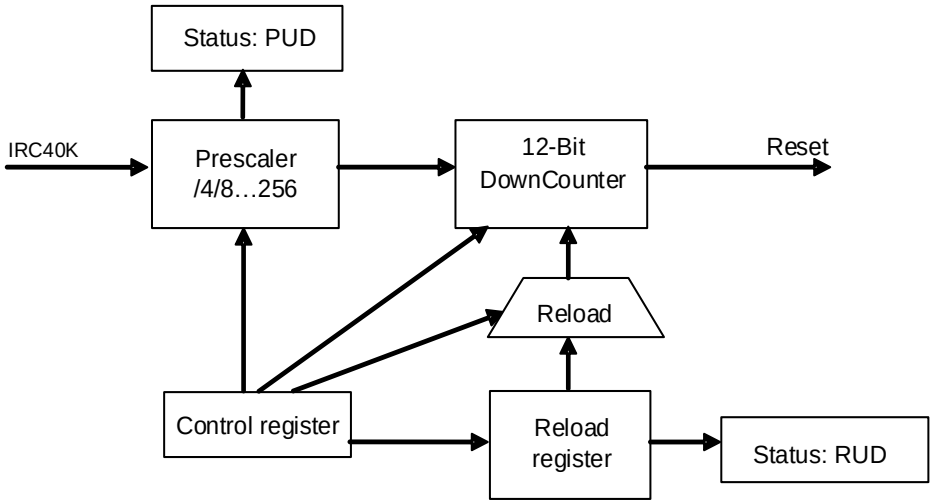
16.1.2. Characteristics

- Free-running 12-bit down counter.
- Generate reset in two conditions when FWDGT is enabled:
  - Reset when the counter reached 0.
  - The counter is refreshed when the value of the counter is greater than the window register value.
- Free clock source, FWDGT can operate even if the main clock fails such as in standby and Deep-sleep modes.
- Hardware free watchdog bit, automatically start the FWDGT or not when power on.
- FWDGT debug mode, the FWDGT can stop or continue to work in debug mode.

16.1.3. Function overview

The free watchdog consists of an 8-stage prescaler and a 12-bit down counter. ***Figure 16-1. Free watchdog block diagram*** shows the functional block of the free watchdog module.

Figure 16-1. Free watchdog block diagram



The free watchdog is enabled by writing the value (0xCCCC) to the control register (FWDGT_CTL), then counter starts counting down. When the counter reaches the value (0x000), there will be a reset.

The counter can be reloaded by writing the value (0xAAAA) to the FWDGT_CTL register at any time. The reload value comes from the FWDGT_RLD register. The software can prevent the watchdog reset by reloading the counter before the counter reaches the value (0x000).

By setting the appropriate window in the FWDGT_WND register, the FWDGT can also work as a window watchdog timer. A reset will occur if the reload operation is performed while the counter is greater than the value stored in the window register (FWDGT_WND). The default value of the FWDGT_WND is 0x0000 0FFF, so if it is not updated, the window option is disabled. A reload operation is performed in order to reset the downcounter to the FWDGT_RLD value and the prescaler counter to generate the next reload, as soon as the window value is changed.

The free watchdog can automatically start at power on when the hardware free watchdog bit in the device option bits is set. To avoid reset, the software should reload the counter before the counter reaches 0x000.

The FWDGT_PSC register, the FWDGT_RLD register and the FWDGT_WND register are write protected. Before writing these registers, the software should write the value (0x5555) to the FWDGT_CTL register. These registers will be protected again by writing any other value to the FWDGT_CTL register. When an update operation of the prescaler register (FWDGT_PSC), window register (FWDGT_WND) or the reload value register (FWDGT_RLD) is ongoing, the status bits in the FWDGT_STAT register are set.

If the FWDGT_HOLD bit in DBG module is cleared, the FWDGT continues to work even the Cortex®-M33 core halted (Debug mode). The FWDGT stops in Debug mode if the FWDGT_HOLD bit is set.

Table 16-1. Min/max FWDGT timeout period at 40KHz (IRC40K)

Prescaler divider	PSC[2:0] bits	Min timeout (ms) RLD[11:0]= 0x000	Max timeout (ms) RLD[11:0]= 0xFFF
1 / 4	000	0.025	409.525
1 / 8	001	0.025	819.025
1 / 16	010	0.025	1638.025
1 / 32	011	0.025	3276.025
1 / 64	100	0.025	6552.025
1 / 128	101	0.025	13104.025
1 / 256	110 or 111	0.025	26208.025

The FWDGT timeout can be more accurately by calibrating the IRC40K.

Note: When after the execution of watchdog reload operation, if the MCU needs enter the deepsleep / standby mode immediately, more than 3 IRC40K clock intervals must be inserted in the middle of reload and deepsleep / standby mode commands by software setting.

16.1.4. Register definition

FWDGT base address: 0x4000 3000

Control register (FWDGT_CTL)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD[15:0]															
w															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different functions are realized by writing these bits with different values. 0x5555: Disable the FWDGT_PSC, FWDGT_RLD and FWDGT_WND write protection. 0xCCCC: Start the free watchdog timer counter. When the counter reduces to 0, the free watchdog generates a reset 0xAAAA: Reload the counter

Prescaler register (FWDGT_PSC)

Address offset: 0x04
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PSC[2:0]		
													rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PSC[2:0]	Free watchdog timer prescaler selection. Write 0x5555 in the FWDGT_CTL register before writing these bits. During a write operation to this register, the PUD bit in the FWDGT_STAT register is set and the value read from this register is invalid.

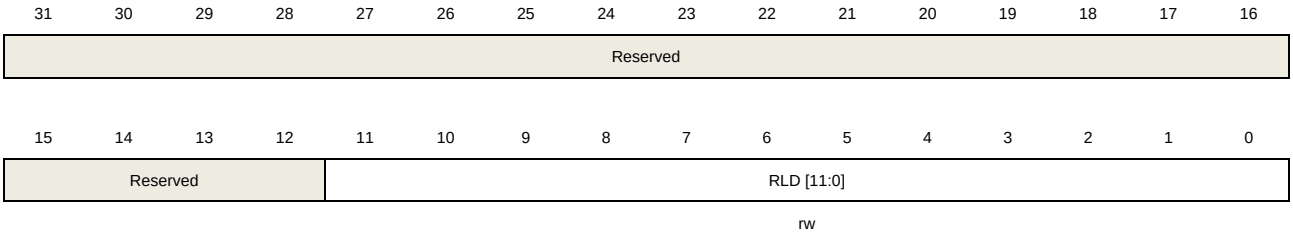
000: 1 / 4
001: 1 / 8
010: 1 / 16
011: 1 / 32
100: 1 / 64
101: 1 / 128
110: 1 / 256
111: 1 / 256

If several prescaler values are used by the application, it is mandatory to wait until PUD bit has been reset before changing the prescaler value. If the prescaler value has been updated, it is not necessary to wait until PUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until PUD is reset).

Reload register (FWDGT_RLD)

Address offset: 0x08
Reset value: 0x0000 0FFF

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT conter with the RLD value. These bits are write-protected. Write 0X5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit has been reset before changing the reload value. If the reload value has been updated, it is not necessary to wait until RUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until RUD is reset).

Status register (FWDGT_STAT)

Address offset: 0x0C
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													WUD	RUD	PUD
													r	r	r

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	WUD	Watchdog counter window value update When a write operation to FWDGT_WND register ongoing, this bit is set and the value read from FWDGT_WND register is invalid.
1	RUD	Free watchdog timer counter reload value update During a write operation to FWDGT_RLD register, this bit is set and the value read from FWDGT_RLD register is invalid.
0	PUD	Free watchdog timer prescaler value update During a write operation to FWDGT_PSC register, this bit is set and the value read from FWDGT_PSC register is invalid.

Window register (FWDGT_WND)

Address offset: 0x10

Reset value: 0x0000 0FFF

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WND[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WND[11:0]	Watchdog counter window value. These bits are used to contain the high limit of the window value to be compared to the downcounter. A reset will occur if the reload operation is performed while the counter is greater than the value stored in this register. The WUD bit in the FWDGT_STAT register must be reset in order to be able to change the reload value. These bits are write protected. Write 0x5555 in the FWDGT_CTL register before writing these bits. If several window values are used by the application, it is mandatory to wait until

WUD bit has been reset before changing the window value. However, after updating the window value it is not necessary to wait until WUD is reset before continuing code execution except in case of low-power mode entry(Before entering low-power mode, it is necessary to wait until WUD is reset).

16.2. Window watchdog timer (WWDGT)

16.2.1. Overview

The window watchdog timer (WWDGT) is used to detect system failures due to software malfunctions. After the window watchdog timer starts, the value of down counter reduces progressively. The watchdog timer causes a reset when the counter reached 0x3F (the CNT[6] bit has been cleared). The watchdog timer also causes a reset when the counter is refreshed before the counter reached the window register value. So the software should refresh the counter in a limited window. The window watchdog timer generates an early wakeup status flag when the counter reaches 0x40. Interrupt occurs if it is enabled.

The window watchdog timer clock is prescaled from the APB1 clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

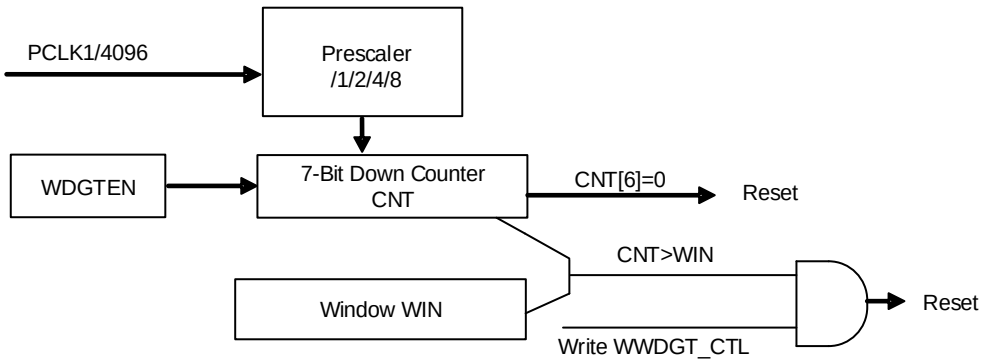
16.2.2. Characteristics

- Programmable free-running 7-bit down counter.
- Generate reset in two conditions when WWDGT is enabled:
  - Reset when the counter reached 0x3F.
  - The counter is refreshed when the value of the counter is greater than the window register value.
- Early wakeup interrupt (EWI): the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x40.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

16.2.3. Function overview

If the window watchdog timer is enabled (set the WDGTEN bit in the WWDGT_CTL), the watchdog timer cause a reset when the counter reaches 0x3F (the CNT[6] bit has been cleared), or the counter is refreshed before the counter reaches the window register value.

Figure 16-2. Window watchdog timer block diagram



The window watchdog timer is always disabled after power on reset. The software starts the

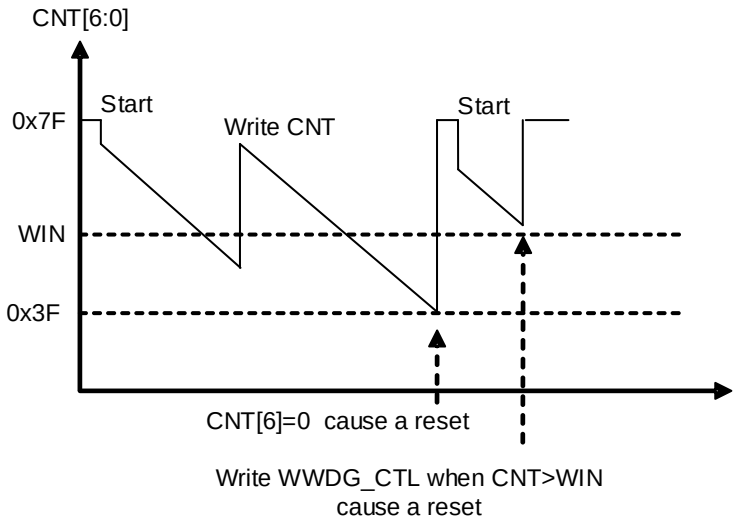
watchdog by setting the WDGTEN bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F(it implies that the CNT[6] bit should be set). The CNT[5:0] determine the maximum time interval between two reloading. The count down speed depends on the APB1 clock and the prescaler (PSC[1:0] bits in the WWDGT_CFG register).

The WIN[6:0] bits in the configuration register (WWDGT_CFG) specifies the window value. The software can prevent the reset event by reloading the down counter. The counter value is less than the window value and greater than 0x3F, otherwise the watchdog causes a reset.

The early wakeup interrupt (EWI) is enabled by setting the EWIE bit in the WWDGT_CFG register, and the interrupt will be generated when the counter reaches 0x40. The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyse the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things.

The EWI interrupt is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 16-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (16\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB1 clock period measured in ms

The ***Table 16-2. Min-max timeout value at 50 MHz (fPCLK1)*** shows the minimum and maximum values of the t_{WWDGT}.

Table 16-2. Min-max timeout value at 50 MHz (f_{PCLK1})

Prescaler divider	PSC[1:0]	Min timeout value	Max timeout value

		CNT[6:0] =0x40	CNT[6:0]=0x7F
1 / 1	00	81.92 μs	5.24 ms
1 / 2	01	163.84 μs	10.49 ms
1 / 4	10	327.68 μs	20.97 ms
1 / 8	11	655.36 μs	41.94 ms

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M33 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

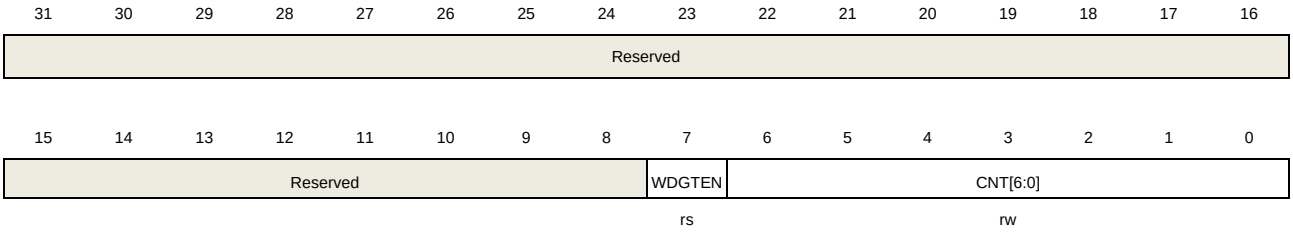
16.2.4. Register definition

WWDGT base address: 0x4000 2C00

Control register (WWDGT_CTL)

Address offset: 0x00
Reset value: 0x0000 007F

This register can be accessed by half-word (16-bit) or word (32-bit).

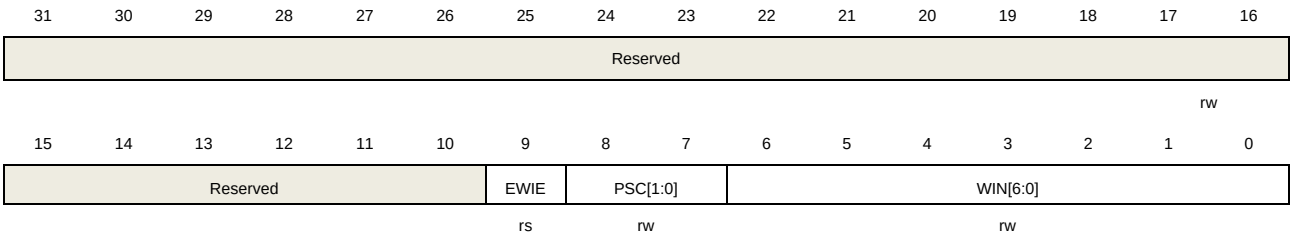


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	WDGTEN	Start the Window watchdog timer. Cleared by a hardware reset. Writing 0 has no effect. 0: Window watchdog timer disabled 1: Window watchdog timer enabled
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset occur when the value of this counter decreases from 0x40 to 0x3F. When the value of this counter is greater than the window value, writing this counter also causes a reset.

Configuration register (WWDGT_CFG)

Address offset: 0x04
Reset value: 0x0000 007F

This register can be accessed by half-word (16-bit) or word (32-bit)



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	EWIE	Early wakeup interrupt enable. If the bit is set, an interrupt occurs when the counter reaches 0x40. It can be cleared by a hardware reset or software reset by setting the

		WWDGTRST bit of the RCU module. A write operation of 0 has no effect.
8:7	PSC[1:0]	Prescaler. The time base of the watchdog counter 00: (PCLK1 / 4096) / 1 01: (PCLK1 / 4096) / 2 10: (PCLK1 / 4096) / 4 11: (PCLK1 / 4096) / 8
6:0	WIN[6:0]	The Window value. A reset occur if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is greater than the Window value.

Status register (WWDGT_STAT)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EWIF
															rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	EWIF	Early wakeup interrupt flag. When the counter reaches 0x40, this bit is set by hardware even the interrupt is not enabled (EWIE in WWDGT_CFG is cleared). This bit is cleared by writing 0. There is no effect when writing 1.

17. Real-time clock (RTC)

17.1. Overview

The RTC is usually used as a clock-calendar. The RTC circuits are located in two power supply domains. The ones in the Backup Domain consist of a 32-bit up-counter, an alarm, a prescaler, a divider and the RTC clock configuration register. That means the RTC settings and time are kept when the device resets or wakes up from Standby mode. While the circuits in the V_{DD} domain only include the APB interface and a control register. In the following sections, the details of the RTC function will be described.

17.2. Characteristics

- 32-bit programmable counter for counting elapsed time.
- Programmable prescaler: Max division factor is up to 2^{20}.
- Separate clock domains:
  - PCLK1 clock domain.
  - RTC clock domain (this clock must be at least 4 times slower than the PCLK1 clock).
- RTC clock source:
  - HXTAL clock divided by 128.
  - LXTAL oscillator clock.
  - IRC40K oscillator clock.
- Maskable interrupt source:
  - Alarm interrupt.
  - Second interrupt.
  - Overflow interrupt.

17.3. Function overview

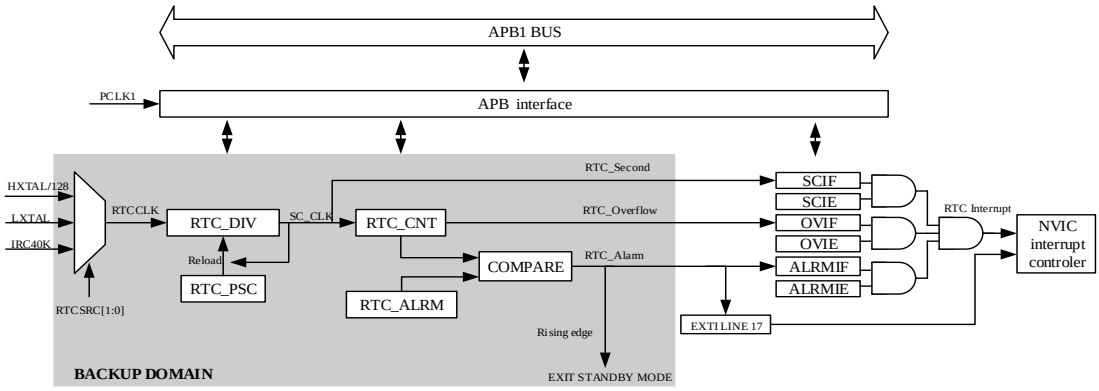
The RTC circuits consist of two major units: APB interface located in PCLK1 clock domain and RTC core located in RTC clock domain.

APB Interface is connected with the APB1 bus. It includes a set of registers, can be accessed by APB1 bus.

RTC core includes two major blocks. One is the RTC prescaler block, which generates the RTC time base clock SC_CLK. RTC prescaler block includes a 20-bit programmable divider (RTC prescaler) which can make SC_CLK is divided from RTC source clock. If second interrupt is enabled in the RTC_INTEN register, the RTC will generate an interrupt at every SC_CLK rising edge. Another block is a 32-bit programmable counter, which can be initialized with the value of current system time. If alarm interrupt is enabled in the RTC_INTEN register, the RTC will generate an alarm interrupt when the system time equals to the alarm time

(stored in the RTC_ALRMH/L register).

Figure 17-1. Block diagram of RTC



Note: The LXTAL is not supported to used as RTC clock source in 48-pin and 32-pin package.

17.3.1. RTC reset

The APB interface and the RTC_INTEN register are reset by system reset. The RTC core (prescaler, divider, counter and alarm) is reset only by a backup domain reset.

Steps to enable access to the backup registers and the RTC after reset are as follows:

1. Set the PMUEN and BKPEN bits in the RCU_APB1EN register to enable the power and backup interface clocks.
2. Enable access to the backup registers and RTC by setting the BKPWEN bit in the (PMU_CTL).

17.3.2. RTC reading

The APB interface and RTC core are located in two different power supply domains.

In the RTC core, only counter and divider registers are readable registers. And the values in the two registers and the RTC flags are internally updated at each rising edge of the RTC clock, which is resynchronized by the APB1 clock.

When the APB interface is immediately enabled from a disable state, the read operation is not recommended because the first internal update of the registers has not finished. That means, when a system reset, power reset, waking up from Standby mode or Deep-sleep mode occurs, the APB interface was in disabled state, but the RTC core has been kept running. In these cases, the correct read operation should first clear the RSYNF bit in the RTC_CTL register and wait for it to be set by hardware. While WFI and WFE have no effects on the RTC APB interface.

17.3.3. RTC configuration

The RTC_PSC, RTC_CNT and RTC_ALRM registers in the RTC core are writable. These

registers' value can be set only when the peripheral enter configuration mode. And the CMF bit in the RTC_CTL register is used to indicate the configuration mode status. The write operation executes when the peripheral exit configuration mode, and it takes at least three RTCCLK cycles to complete. The value of the LWOFF bit in the RTC_CTL register sets to '1', if the write operation finished. The new write operation should wait for the previous one finished.

The configuration steps are as follows:

1. Wait until the value of LWOFF bit in the RTC_CTL register sets to '1';
2. Enter Configuration mode by setting the CMF bit in the RTC_CTL register;
3. Write to the RTC registers;
4. Exit Configuration mode by clearing the CMF bit in the RTC_CTL register;
5. Wait until the value of LWOFF bit in the RTC_CTL register sets to '1'.

17.3.4. RTC flag assertion

Before the update of the RTC Counter, the RTC second interrupt flag (SCIF) is asserted on the last RTCCLK cycle.

Before the counter equal to the RTC Alarm value which stored in the Alarm register increases by one, the RTC Alarm interrupt flag (ALRMIF) is asserted on the last RTCCLK cycle.

Before the counter equals to 0x0, the RTC Overflow interrupt flag (OVIF) is asserted on the last RTCCLK cycle.

The RTC Alarm write operation and Second interrupt flag must be synchronized by using either of the following sequences:

- Use the RTC alarm interrupt and update the RTC Alarm and/or RTC Counter registers inside the RTC interrupt routine;
- Update the RTC Alarm and/or the RTC Counter registers after the SCIF bit to be set in the RTC Control register.

Figure 17-2. RTC second and alarm waveform example (RTC_PSC = 3, RTC_ALRM = 2)

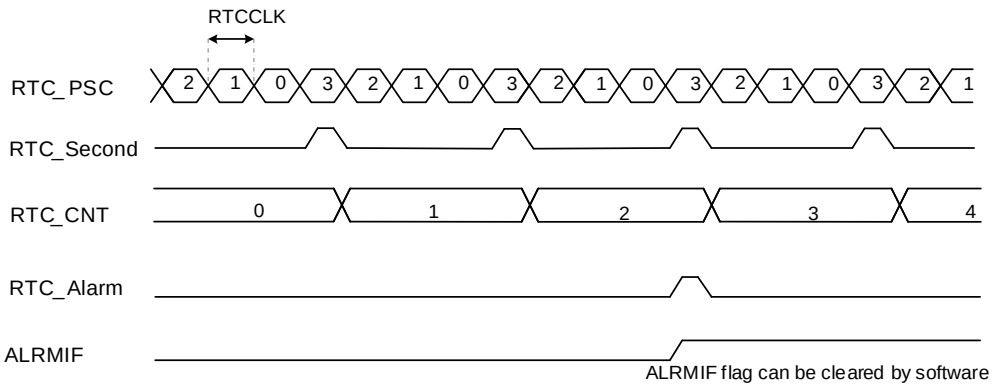
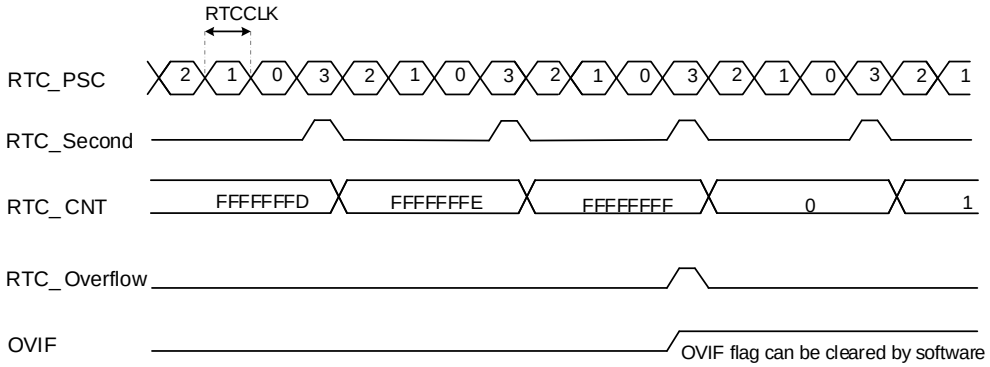


Figure 17-3. RTC second and overflow waveform example (RTC_PSC= 3)



17.4. Register definition

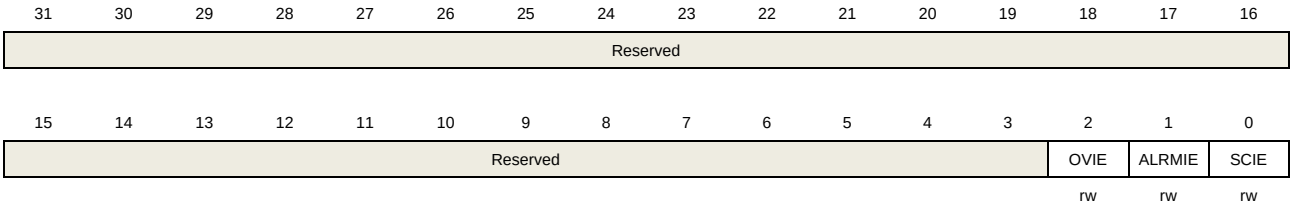
RTC base address: 0x4000 2800

17.4.1. RTC interrupt enable register (RTC_INTEN)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



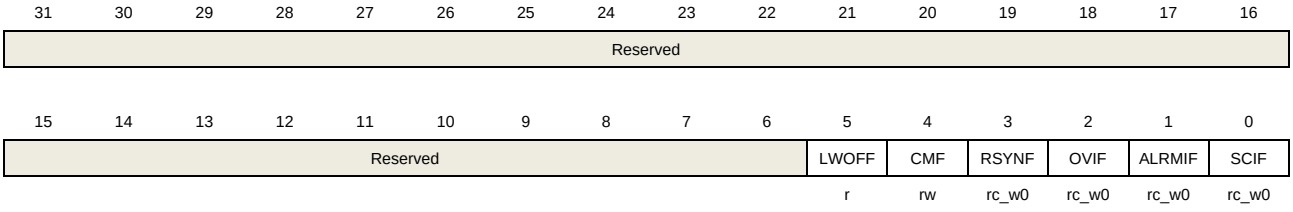
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	OVIE	Overflow interrupt enable 0: Disable overflow interrupt 1: Enable overflow interrupt
1	ALRMIE	Alarm interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
0	SCIE	Second interrupt enable 0: Disable second interrupt 1: Enable second interrupt

17.4.2. RTC control register (RTC_CTL)

Address offset: 0x04

Reset value: 0x0000 0020

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.

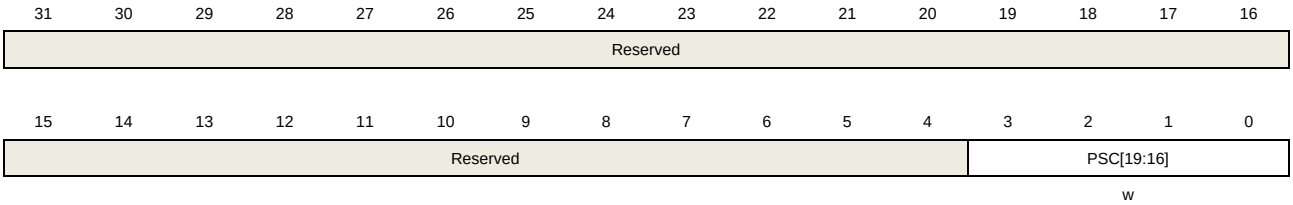
5	LWOFF	Last write operation finished flag 0: Last write operation on RTC registers did not finished. 1: Last write operation on RTC registers finished.
4	CMF	Configuration mode flag 0: Exit configuration mode. 1: Enter configuration mode.
3	RSYNF	Registers synchronized flag 0: Registers not yet synchronized with the APB1 clock. 1: Registers synchronized with the APB1 clock.
2	OVIF	Overflow interrupt flag 0: Overflow event not detected 1: Overflow event detected. An interrupt will occur if the OVIE bit is set in RTC_INTEN.
1	ALRMIF	Alarm interrupt flag 0: Alarm event not detected 1: Alarm event detected. An interrupt named RTC global interrupt will occur if the ALRMIE bit is set in RTC_INTEN. And another interrupt named the RTC Alarm interrupt will occur if the EXTI 17 is enabled in interrupt mode.
0	SCIF	Second interrupt flag 0: Second event not detected. 1: Second event detected. An interrupt will occur if the SCIE bit is set in RTC_INTEN. Set by hardware when the divider reloads the value in RTC_PSCH/L, thus incrementing the RTC counter.

17.4.3. RTC prescaler high register (RTC_PSCH)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	PSC[19:16]	RTC prescaler value high

17.4.4. RTC prescaler low register (RTC_PSCL)

Address offset: 0x0C

Reset value: 0x0000 8000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
w															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	RTC prescaler value low The frequency of SC_CLK is the RTCCLK frequency divided by (PSC[19:0]+1).

17.4.5. RTC divider high register (RTC_DIVH)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DIV[19:16]			
												r			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	DIV[19:16]	RTC divider value high

17.4.6. RTC divider low register (RTC_DIVL)

Address offset: 0x14

Reset value: 0x0000 8000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV[15:0]															

r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DIV[15:0]	RTC divider value low The RTC divider register is reloaded by hardware when the RTC prescaler or RTC counter register updated.

17.4.7. RTC counter high register (RTC_CNTH)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[31:16]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[31:16]	RTC counter value high

17.4.8. RTC counter low register (RTC_CNTL)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

rw

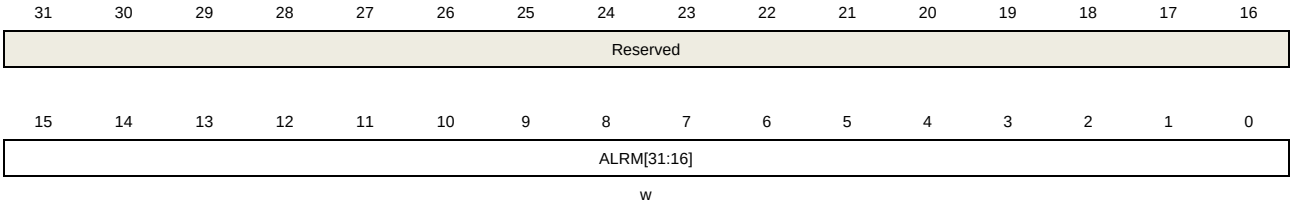
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	RTC counter value low

17.4.9. RTC alarm high register (RTC_ALRMH)

Address offset: 0x20

Reset value: 0x0000 FFFF

This register can be accessed by half-word (16-bit) or word (32-bit).



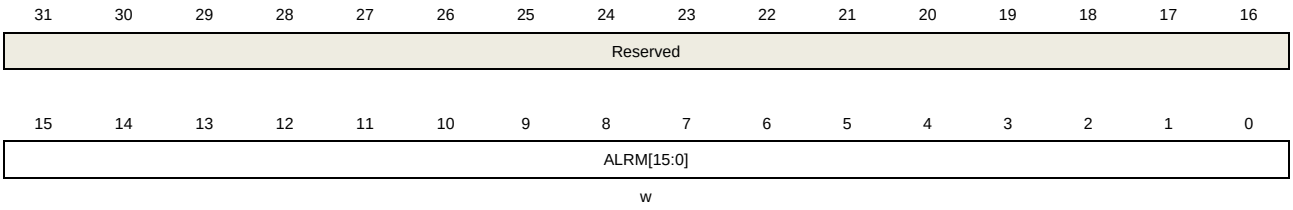
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	ALRM[31:16]	RTC alarm value high

17.4.10. RTC alarm low register (RTC_ALRML)

Address offset: 0x24

Reset value: 0x0000 FFFF

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	ALRM[15:0]	RTC alarm value low

18. TIMER

Table 18-1. Timers (TIMERx) are divided into three sorts

TIMER	TIMER0/7/19/20	TIMER1	TIMER5/6
TYPE	Advanced	General-L0	Basic
Prescaler	16-bit	16-bit	16-bit
Counter	16-bit	16-bit	16-bit
Count mode	UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP ONLY
Repetition	●	×	×
Channel Capture/ Compare	8	4	0
Complementary & Dead-time	●	×	×
Break	●	×	×
Single Pulse	●	●	●
Quadrature Decoder	●	●	×
Master-slave management	●	●	×
Inter Connection	● (1)	● (1)	TRGO TO DAC
DMA	●	●	● (2)
Debug Mode	●	●	●

(1) Please refer to ***Trigger selection controller (TRIGSEL)*** for more details.

(2) Only update events will generate a DMA request. TIMER5/6 do not have DMAS bit (DMA request source selection).

18.1. Advanced timer (TIMERx, x=0, 7, 19, 20)

18.1.1. Overview

The advanced timer module (TIMER0/7/19/20) is an eight-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The advanced timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the advanced timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time insertion module which is suitable for motor control applications.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison.

18.1.2. Characteristics

- Total channel num: 8.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is independent and user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode and trigger out.
- Programmable dead time insertion and Separated dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request: update event, trigger event, compare/capture event, commutation event and break input.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

18.1.3. Block diagram

Figure 18-1. Advanced timer block diagram provides details of the internal configuration of the advanced timer, and *Table 18-2. Advanced timer channel description* introduces the input and output of the channels.

Figure 18-1. Advanced timer block diagram

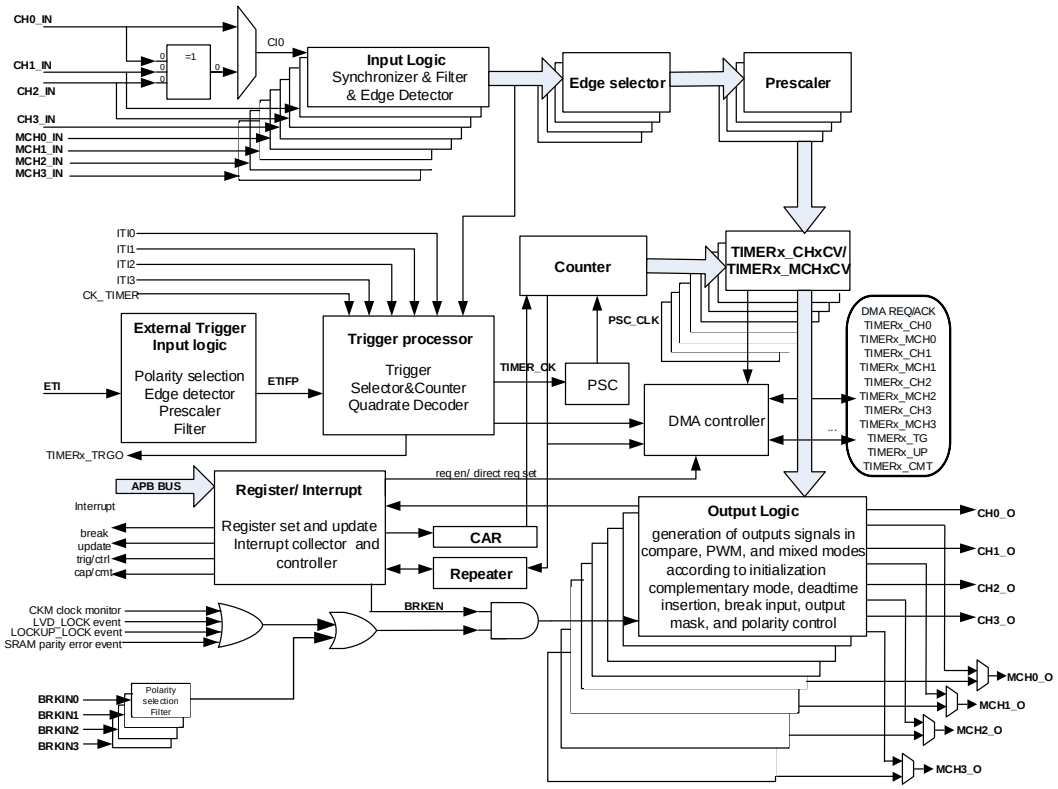


Table 18-2. Advanced timer channel description

Channel name (x=0..3)	MCHxMSEL[1:0]=00 independent mode	MCHxMSEL[1:0]=01 mirrored mode	MCHxMSEL[1:0]=11 complementary mode
CHx (Channel x)	CHx and MCHx can independently input capture and compare output	CHx output is the same as MCHx output (just used for output).	only the CHx is valid for input, and the outputs of MCHx and CHx are complementary
MCHx (Multi mode channel x)			

18.1.4. Function overview

Clock source configuration

The clock source of the advanced timer can be either the CK_TIMER or an alternate clock source controlled by SMC bits (TIMERx_SMCFG bit[2:0]).

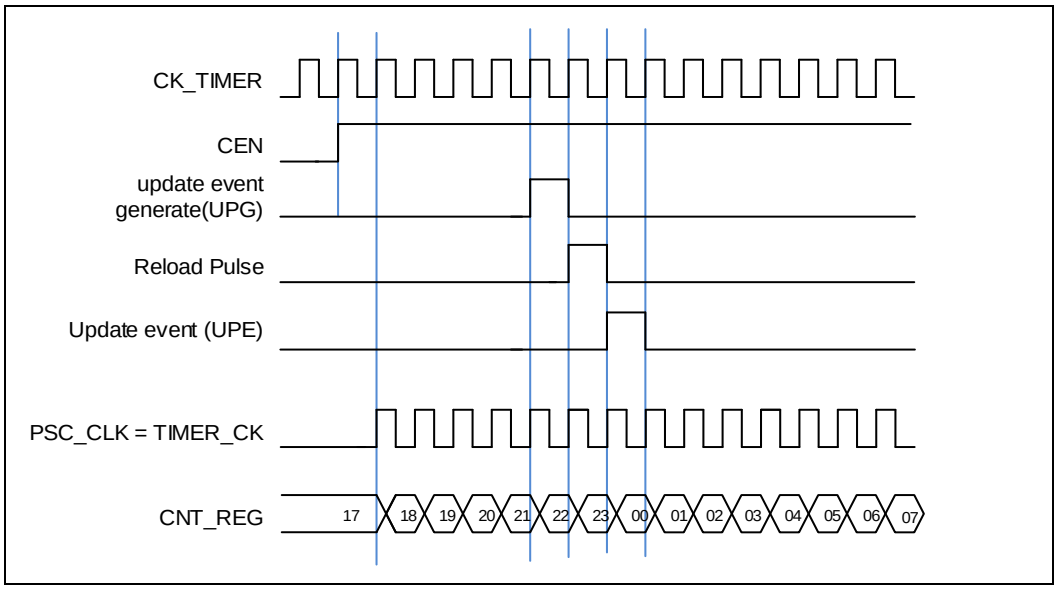
- SMC[2:0] = 3'b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when the SMC[2:0] = 3'b000. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If the SMC[2:0] in the TIMERx_SMCFG register are setting to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock sources selected by the TRGS[3:0] in the TIMERx_SMCFG register, more details will be introduced later. When the SMC[2:0] bits are set to 0x4, 0x5 or 0x6, the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 18-2. Timing chart of internal clock divided by 1



- SMC[2:0] = 3'b111 (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CHn/ TIMERx_MCHn（n=0..3）. This mode can be selected by setting SMC[2:0] to 0x7 and the TRGS[3:0] to 0x4~0x6 and 0x8~0xD.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/1/2/3. This mode can be selected by setting SMC[2:0] to 0x7 and the TRGS[3:0] to 0x0, 0x1, 0x2 or 0x3.

- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is setting the SMC[2:0] to 0x7 and the TRGS[3:0] to 0x7. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock

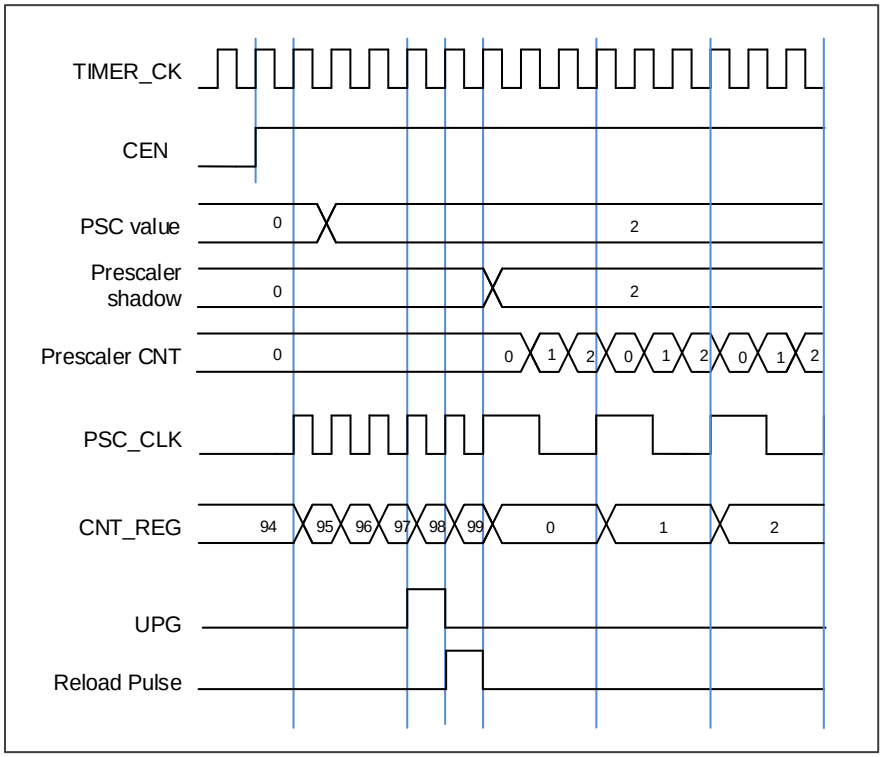
pulse on each ETI signal rising edge to clock the counter prescaler.

Note: The ETI pin can select from TIMER_ETIx (x=0...2) pins, and each advanced TIMER only can use one of them. Please refer to ***TIMER input source select register (SYSCFG_TIMERINSEL)*** for more details.

Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescaler factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 18-3. Timing chart of PSC value change from 0 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again and an overflow event will be generated. In addition, the update events will be generated after (TIMERx_CREP+1) times of overflow events. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter register, counter auto reload register, prescaler register) are updated.

Figure 18-4. Timing chart of up counting mode, PSC=0/2 and *Figure 18-5. Timing chart of up counting mode, change TIMERx_CAR on the go* show some examples of the counter behavior for different clock prescaler factors when TIMERx_CAR=0x63.

Figure 18-4. Timing chart of up counting mode, PSC=0/2

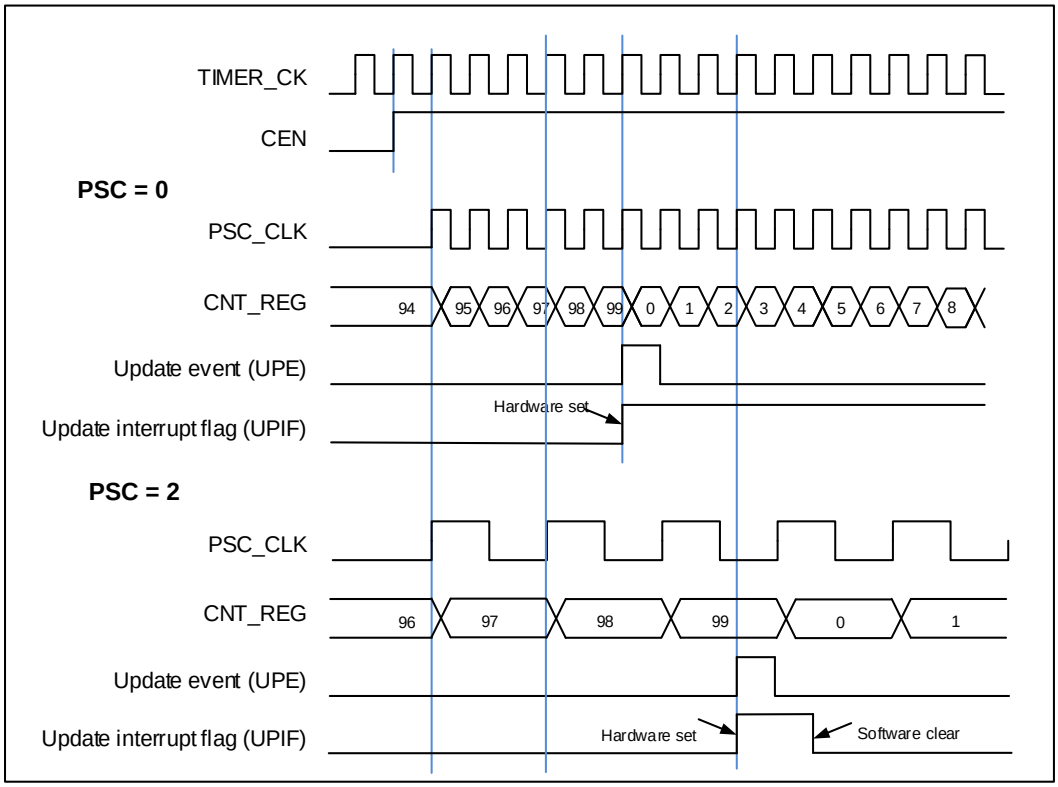
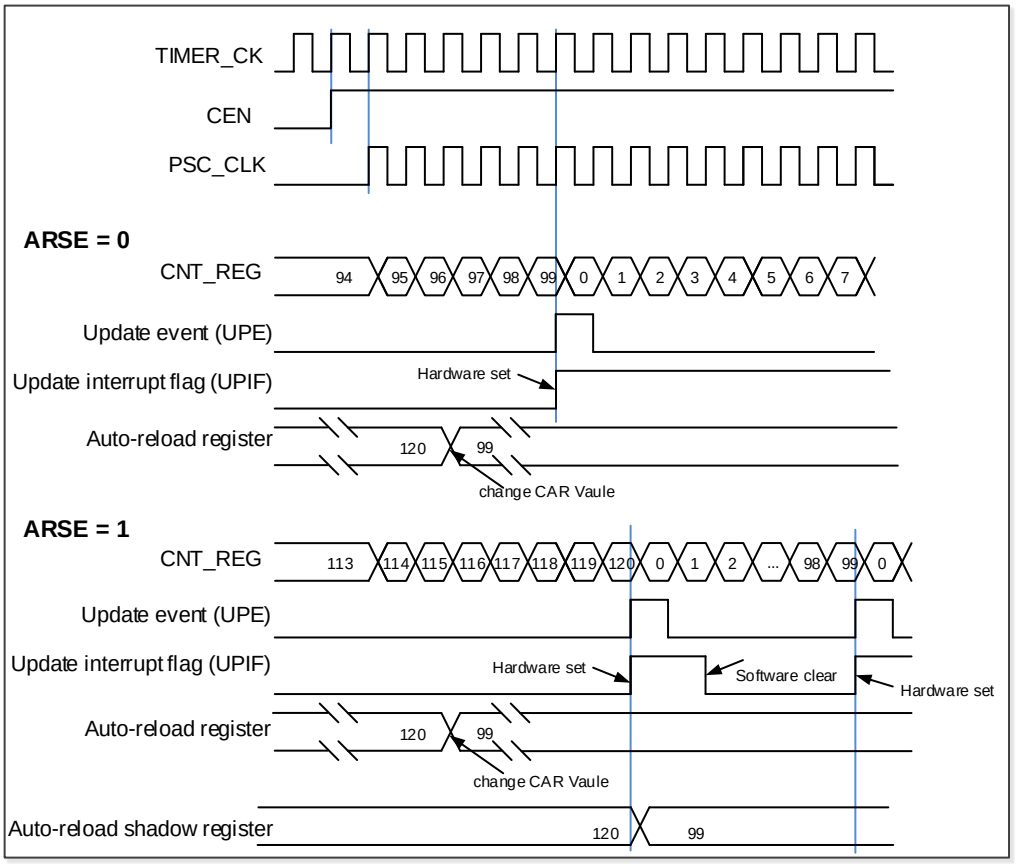


Figure 18-5. Timing chart of up counting mode, change TIMERx_CAR on the go



Counter down counting

In this mode, the counter counts down continuously from the counter-reload value, which is defined in the TIMERx_CAR register, to 0 in a count-down direction. Once the counter reaches to 0, the counter the counter will start counting down from the counter-reload value again and an underflow event will be generated. In addition, the update event will be generated after (TIMERx_CREP+1) times of underflow. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down-counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter register, counter auto reload register, prescaler register) are updated.

Figure 18-6. Timing chart of down counting mode, PSC=0/2 and *Figure 18-7. Timing chart of down counting mode, change TIMERx_CAR on the go* show some examples of

the counter behavior in different clock frequencies when TIMERx_CAR = 0x99.

Figure 18-6. Timing chart of down counting mode, PSC=0/2

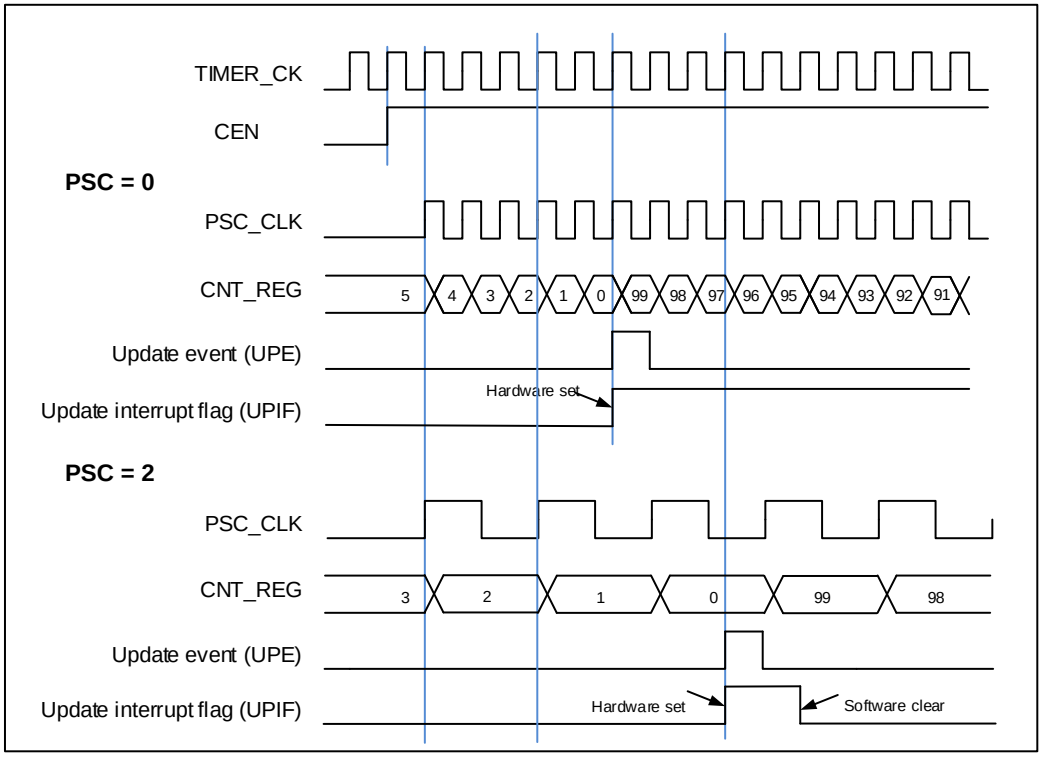
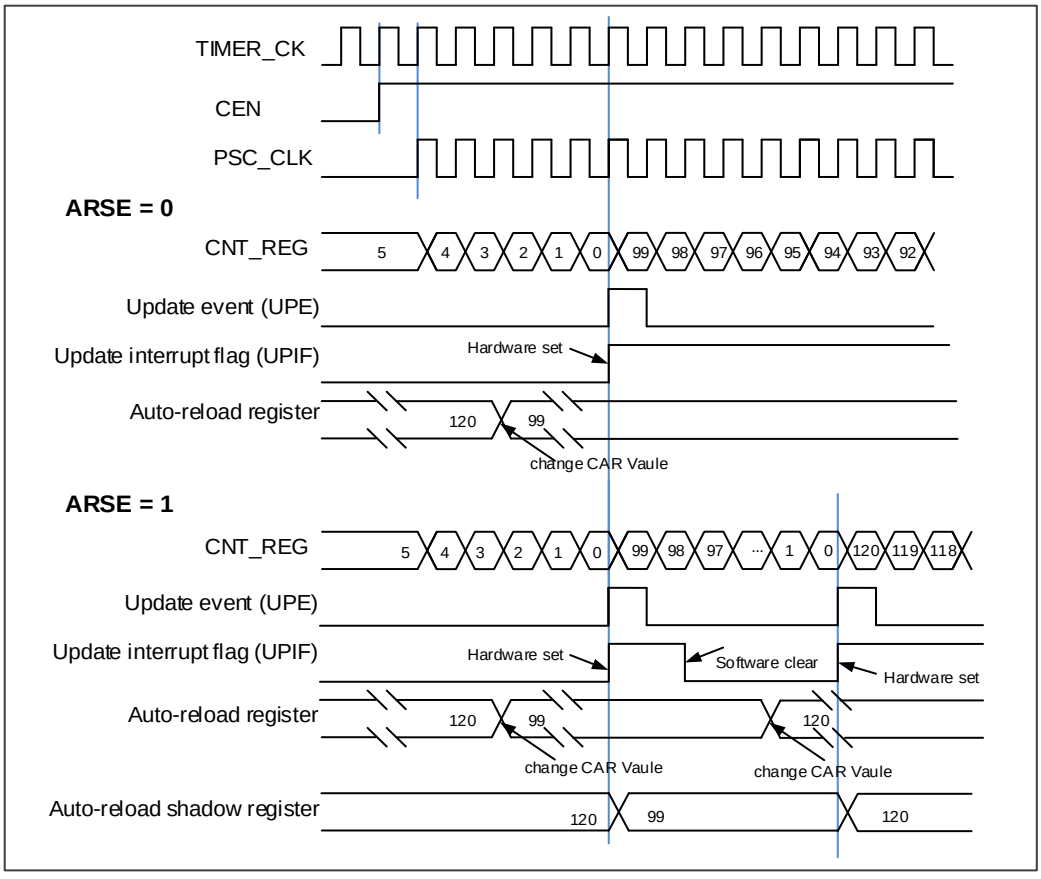


Figure 18-7. Timing chart of down counting mode, change TIMERx_CAR on the go



Counter center-aligned counting

In this mode, the counter counts up from 0 to the counter-reload value and then counts down to 0 alternatively. The Timer module generates an overflow event when the counter counts to the counter-reload value subtract 1 in the up-counting direction and generates an underflow event when the counter counts to 1 in the down-counting direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned mode.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

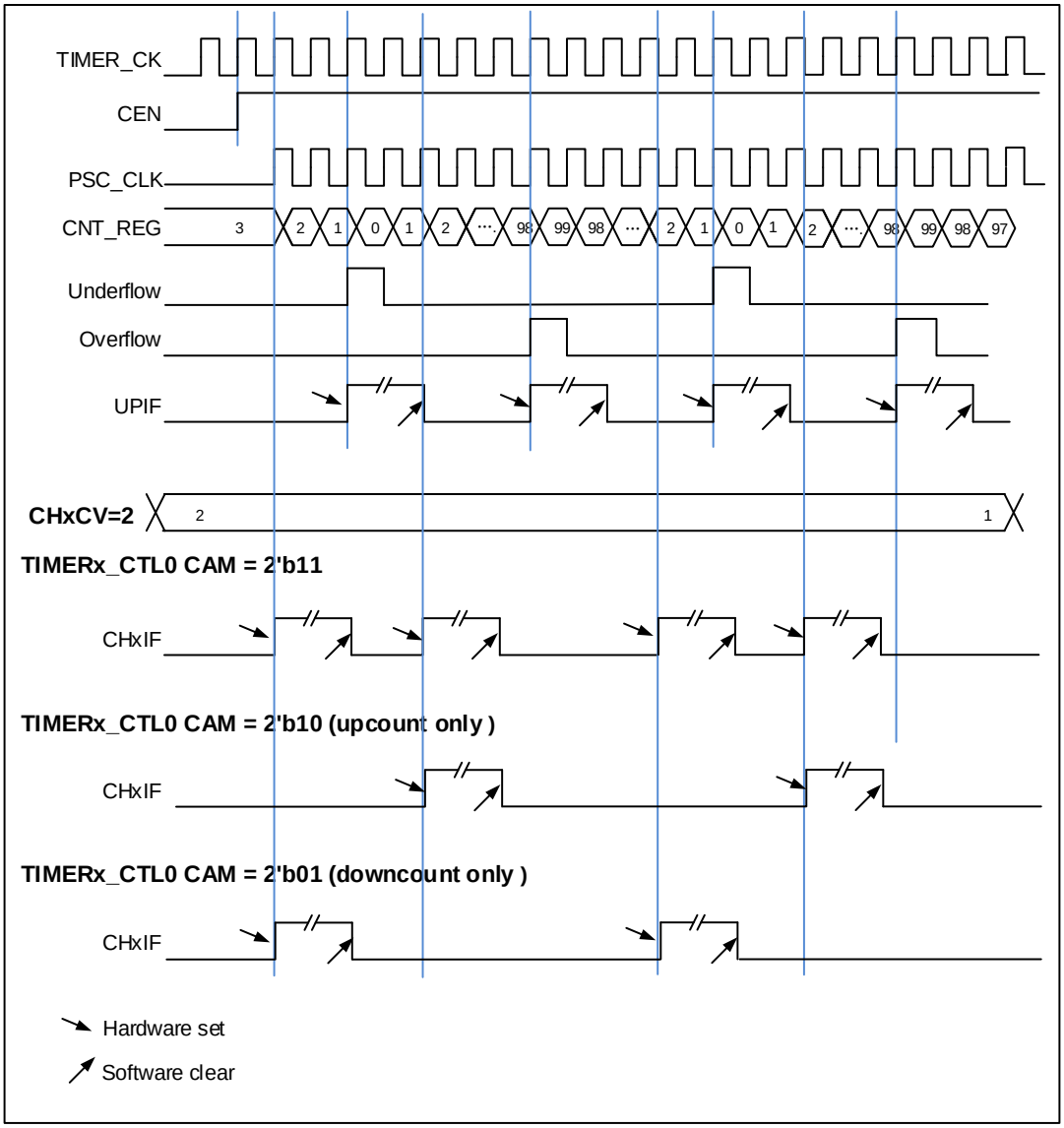
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM[1:0] in TIMERx_CTL0. The details refer to ***Figure 18-8. Timing chart of center-aligned counting mode***.

If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter register, counter auto-reload register, prescaler register) are updated.

Figure 18-8. Timing chart of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 18-8. Timing chart of center-aligned counting mode



Update event (from overflow/underflow) rate configuration

The rate of update events generation (from overflow and underflow events) can be configured by the TIMERx_CREP register. Counter repetition is used to generator update event or updates the timer registers only after a given number (N+1) of cycles of the counter, where N is CREP in TIMERx_CREP register. The repetition counter is decremented at each counter overflow (does not exist in down counting mode) and underflow (does not exist in up counting mode).

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generate an update event.

The new written CREP value will not take effect until the next update event. When the value of CREP is odd, and the counter is counting in center-aligned mode, the update event is generated (on overflow or underflow) depending on when the written CREP value takes effect. If an update event is generated by software after writing an odd number to CREP, the update

events will be generated on the underflow. If the next update event occurs on overflow after writing an odd number to CREP, then the subsequent update events will be generated on the overflow.

Figure 18-9. Repetition counter timing chart of center-aligned counting mode

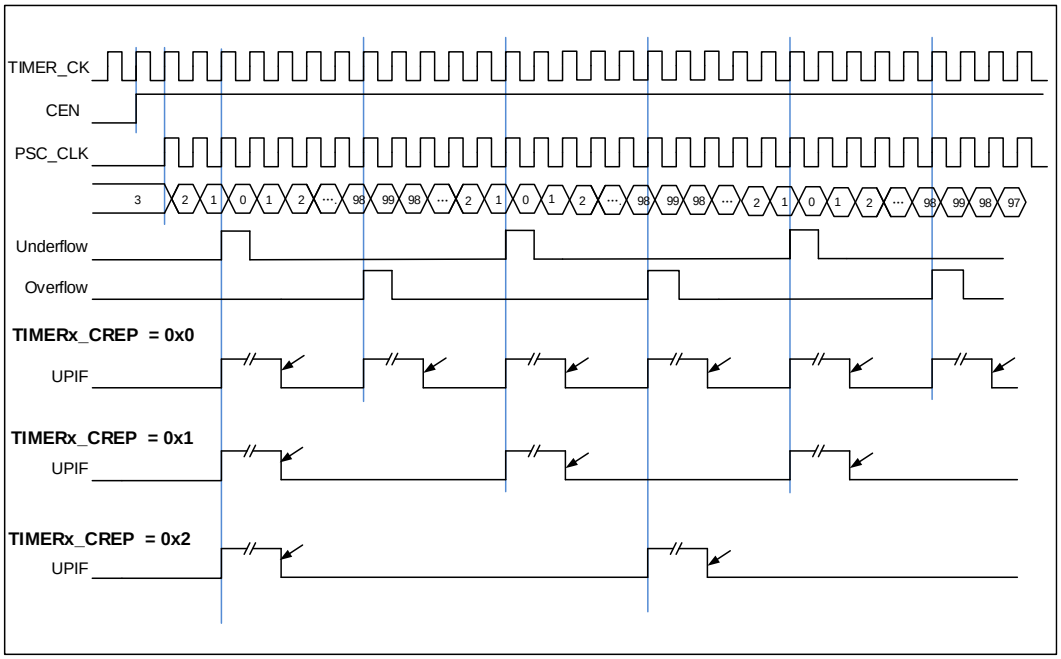


Figure 18-10. Repetition counter timing chart of up counting mode

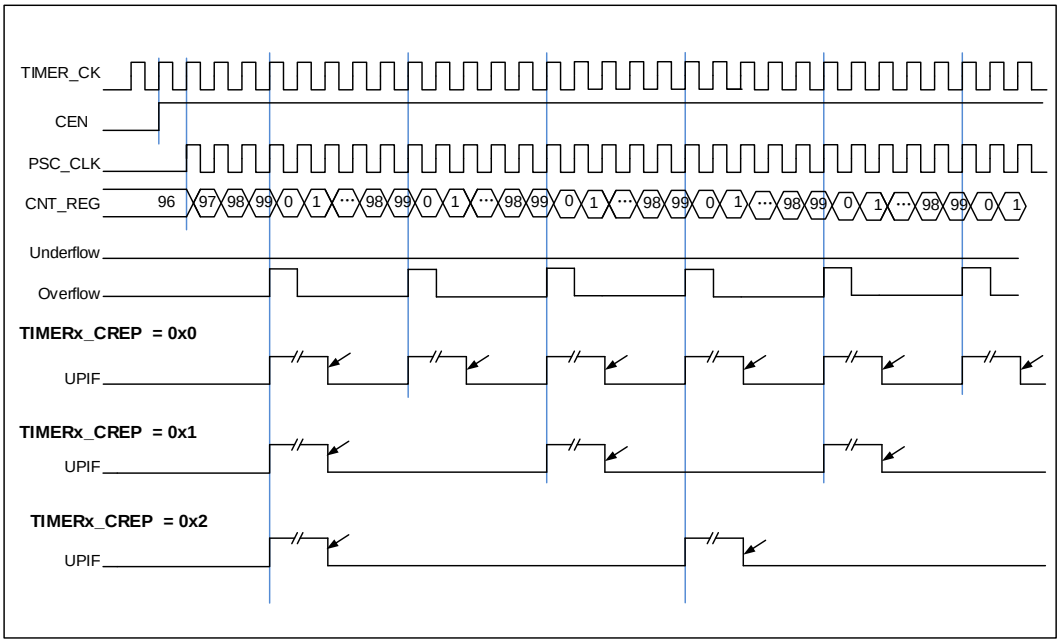
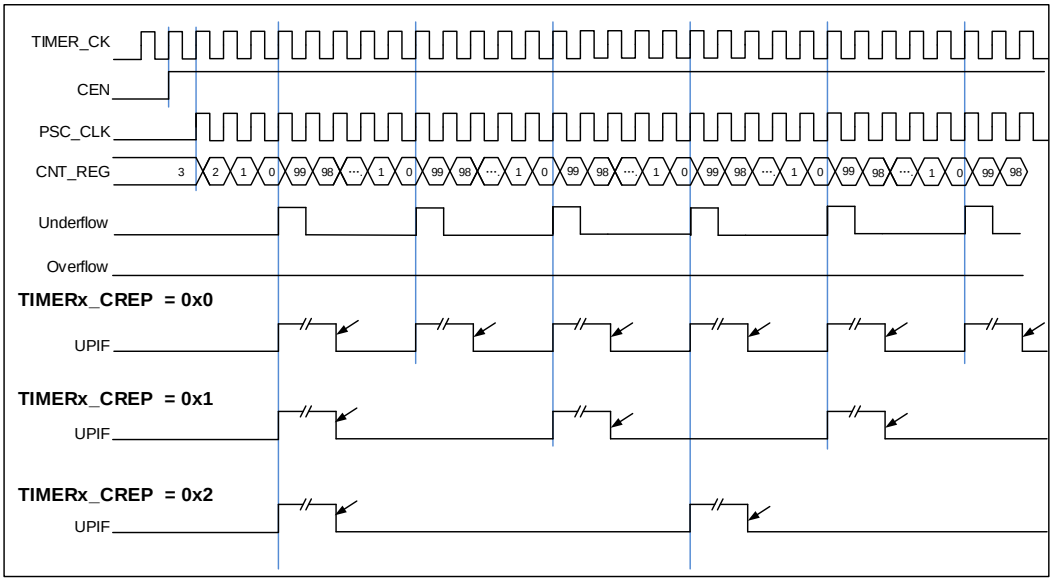


Figure 18-11. Repetition counter timing chart of down counting mode



Input capture and output compare channels

The advanced timer has eight independent channels which can be used as capture inputs or compare outputs. Each channel is built around a channel capture compare register including an input stage, a channel controller and an output stage.

When the channels are used for input, channel x and multi mode channel x can perform input capture independently; when the channels are used for comparison output, the channel x and multi mode channel x can output independent, mirrored, and complementary outputs.

- **Channel input capture function**

Channel input capture function allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV/ TIMERx_MCHxCV(x=0...3) registers, at the same time the CHxIF/ MCHxIF (x=0..3) bits are set and the channel interrupt is generated if it is enabled when CHxIE/ MCHxIE =1(x=0..3).

Figure 18-12. Channel 0 input capture principle

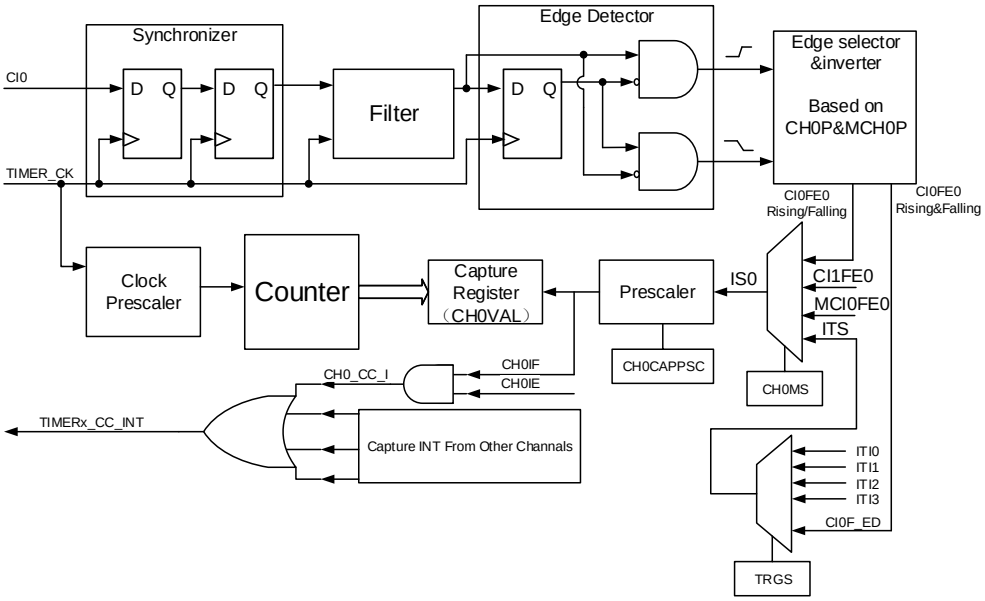
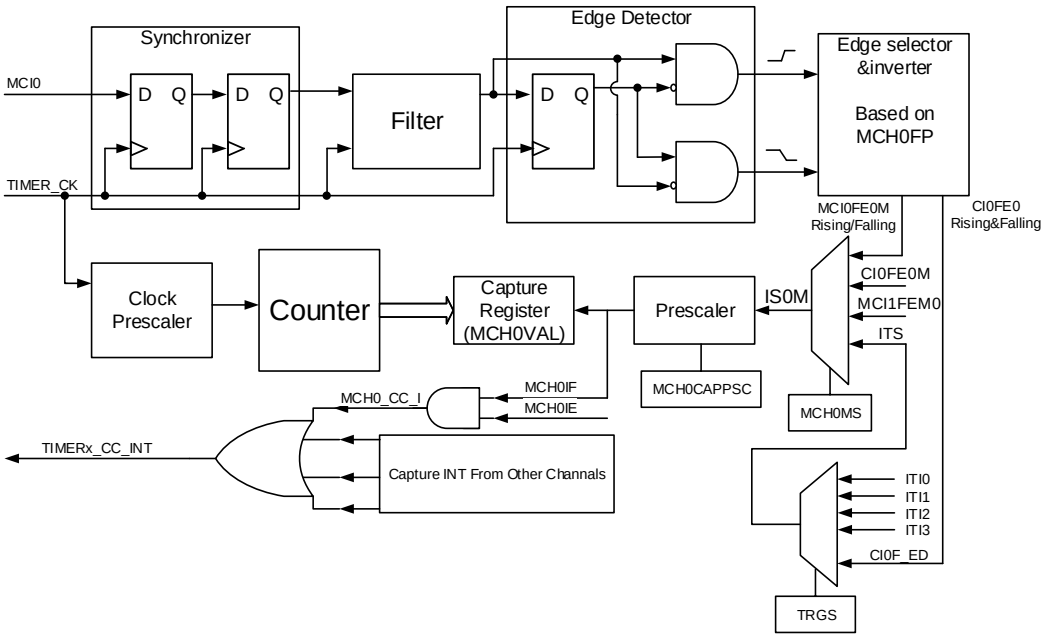


Figure 18-13. Multi mode channel 0 input capture principle



The input signals of channelx (CIx/ MCIx) can be the TIMERx_CHx/ TIMERx_MCHxCV signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals (just for CI0).

First, the input signal of channel (CIx/ MCIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP/ MCHxP or MCHxFP bits. The input capture signal can also be selected from the input signal of other channel or the internal

trigger signal by configuring CHxMS/ MCHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV/ TIMERx_MCHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT bit in TIMERx_CHCTL0 register and MCHxCAPFLT bit in TIMERx_MCHCTL0 register).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT or MCHxCAPFLT bit.

Step2: Edge selection (CHxP and MCHxP bits in TIMERx_CHCTL2 register, MCHxFP[1:0] bits in TIMERx_MCHCTL2 register).

Rising edge or falling edge, choose one by configuring CHxP and MCHxP bits or MCHxFP[1:0] bits.

Step3: Capture source selection (CHxMS bit in TIMERx_CHCTL0 register, MCHxMS bit in TIMERx_MCHCTL0 register).

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x000 or MCHxMS!=0x000) and TIMERx_CHxCV/ TIMERx_MCHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN and MCHxEN bits in TIMERx_CHCTL2).

Result: When the wanted input signal is captured, TIMERx_CHxCV/ TIMERx_MCHxCV will be set by counter's value and CHxIF/ MCHxIF bit is asserted. If the CHxIF/ MCHxIF bit is 1, the CHxOF/ MCHxOF bit will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN.

Direct generation: A DMA or interrupt request is generated by setting CHxG directly.

The channel input capture function can be also used for pulse width measurement from signals on the TIMERx_CHx and TIMERx_MCHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 3'b001 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 3'b010 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

- **Channel output compare function**

Figure 18-14. Channel output compare principle (when MCHxMSEL = 2'00, x=0, 1, 2, 3), ***Figure 18-15. Channel output compare principle (when MCHxMSEL = 2'01, x=0, 1, 2, 3)*** and ***Figure 18-16. Channel output compare principle (with complementary output when MCHxMSEL = 2'11, x=0,1,2,3)*** show the principle circuit of channels output compare function.

Figure 18-14. Channel output compare principle (when MCHxMSEL = 2'00, x=0, 1, 2, 3)

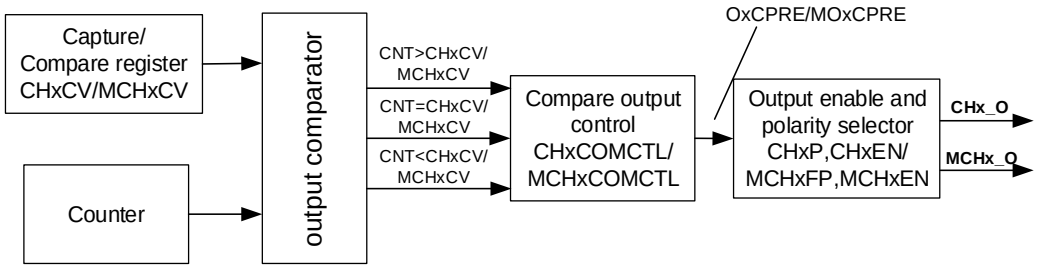


Figure 18-15. Channel output compare principle (when MCHxMSEL = 2'01, x=0, 1, 2, 3)

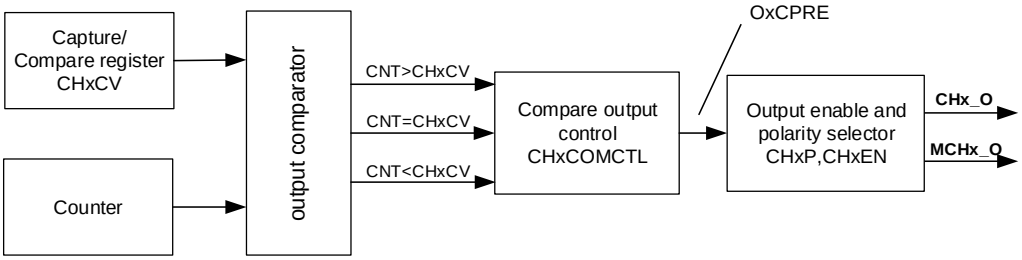
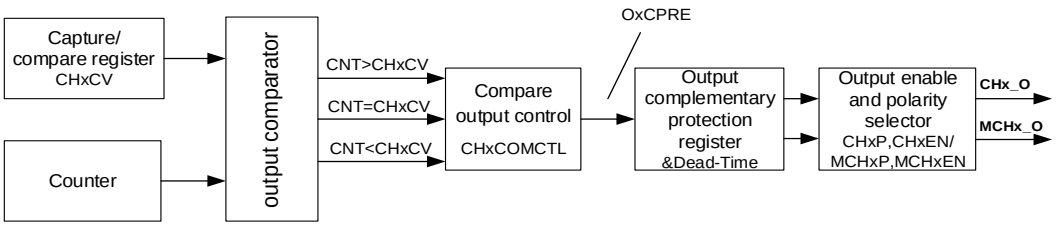


Figure 18-16. Channel output compare principle (with complementary output when MCHxMSEL = 2'11, x=0,1,2,3)



The relationship between the channel output signal CHx_O/MCHx_O and the OxCPRE/ MOxCPRE signal (more details refer to ***Channel output prepare signal***) is described as blew(the active level of OxCPRE is high and the active level of MOxCPRE is high).

- When MCHxMSEL=2'b00 (in TIMERx_CTL2 register), the MCHx_O output is independent from the CHx_O output. The output level of CHx_O depends on OxCPRE signal, CHxP bit and CHxEN bit (please refer to the TIMERx_CHCTL2 register for more details). The output level of MCHx_O depends on MOxCPRE signal, MCHxFP[1:0] bits and MCHxEN bit (please refer to the TIMERx_MCHCTL2 for more details). Please refer to ***Figure 18-14. Channel output compare principle (when MCHxMSEL = 2'00, x=0, 1, 2, 3)***.
- When MCHxMSEL=2'b01, the MCHx_O output is the same as the CHx_O output. The output level of CHx_O and MCHx_O depends on OxCPRE signal, CHxP bit and CHxEN bits. It is invalid for the configuration of MCHx_O. Please refer to ***Figure 18-15. Channel***

output compare principle (when MCHxMSEL = 2'01, x=0, 1, 2, 3).

- When MCHxMSEL=2'b11, the MCHx_O output is the inverse of the CHx_O output. The output level of CHx_O/MCHx_O depends on OxCPRE signal, CHxP/ MCHxP bits and CHxEN/MCHxEN bits. Please refer to *Figure 18-16. Channel output compare principle (with complementary output when MCHxMSEL = 2'11, x=0,1,2,3)*.

For examples (the MCHx_O output is independent from the CHx_O output):

1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxEN=1 (the output of CHx_O is enabled):
   If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;
   If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

2) Configure MCHxP=1 (the active level of MCHx_O is low, contrary to MOxCPRE), MCHxEN=1 (the output of MCHx_O is enabled):
   If the output of MOxCPRE is active(high) level, the output of CHx_O is active(low) level;
   If the output of MOxCPRE is inactive(low) level, the output of CHx_O is active(high) level.

When MCHxMSEL=2'b11 and CHx_O and MCHx_O are output at the same time, the specific outputs of CHx_O and MCHx_O are related to the relevant bits (ROS, IOS, POEN and DTCFG bits) in the TIMERx_CCHP register. Please refer to *Channel output complementary PWM* for more details.

In channel output compare function, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register of an output compare channel, the channel output can be set, cleared, or toggled based on CHxCOMCTL/ MCHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register, the CHxIF/ MCHxIF bit will be set and the channel interrupt is generated if CHxIE/ MCHxIE = 1. And the DMA request will be asserted, if CHxDEN/ MCHxDEN =1.

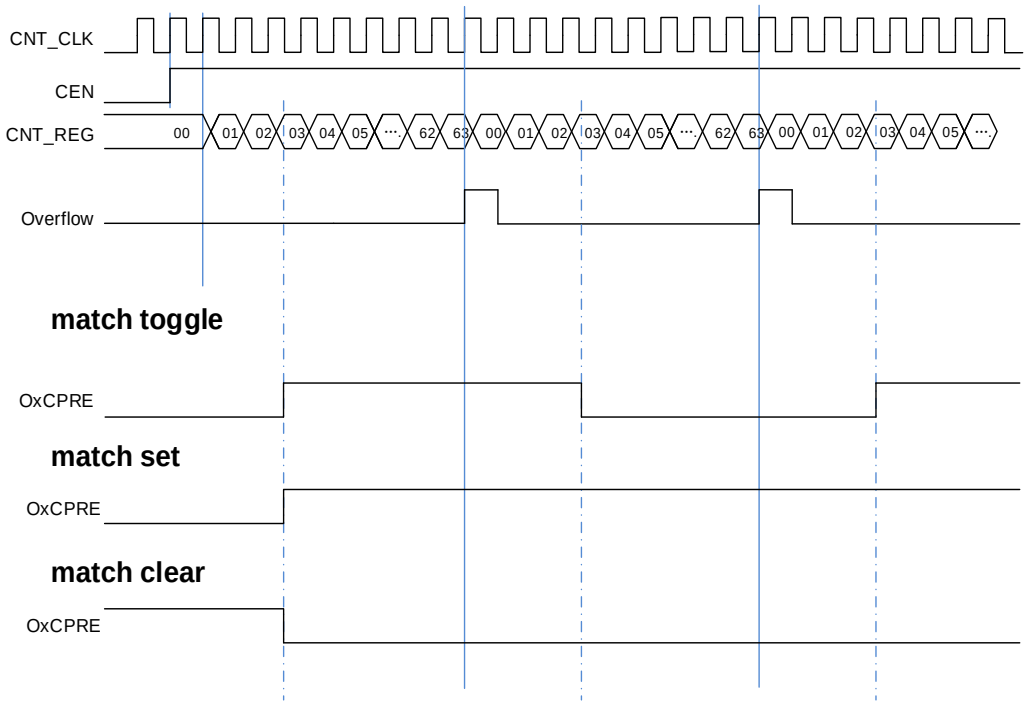
So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.
Step2: Compare mode configuration.
- Set the shadow enable mode by CHxCOMSEN/ MCHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL/ MCHxCOMCTL.
- Select the active polarity by CHxP/MCHxP/ MCHxFP.
- Enable the output by CHxEN/ MCHxEN.

Step3: Interrupt/DMA request enable configuration by CHxIE/ MCHxIE /CHxDEN/ MCHxDEN.
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.
The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed ongoing to meet the expected waveform.
Step5: Start the counter by configuring CEN to 1.

Figure 18-17. Output-compare under three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 18-17. Output-compare under three modes



Output PWM function

In the PWM output function (by setting the CHxCOMCTL/ MCHxCOMCTL bit to 3'b110 (PWM mode 0) or to 3'b 111(PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV/ TIMERx_MCHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV/ TIMERx_MCHxCV. ***Figure 18-18. EAPWM timechart*** shows the EAPWM output and interrupts waveform.

The CAPWM's period is determined by 2*TIMERx_CAR, and the duty cycle is determined by 2*TIMERx_CHxCV/ TIMERx_MCHxCV. ***Figure 18-19. CAPWM timechart*** shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV/ TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL/ MCHxCOMCTL =3'b110). And if the value of TIMERx_CHxCV/ TIMERx_MCHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 1 (CHxCOMCTL/ MCHxCOMCTL =3'b111).

Figure 18-18. EAPWM timechart

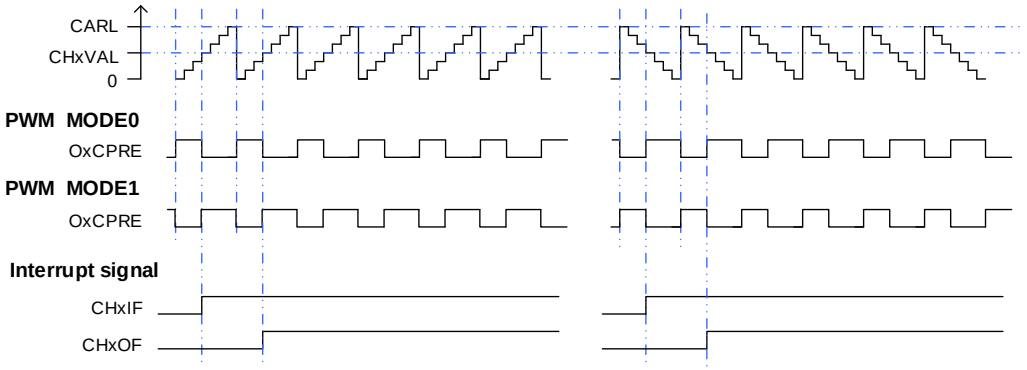
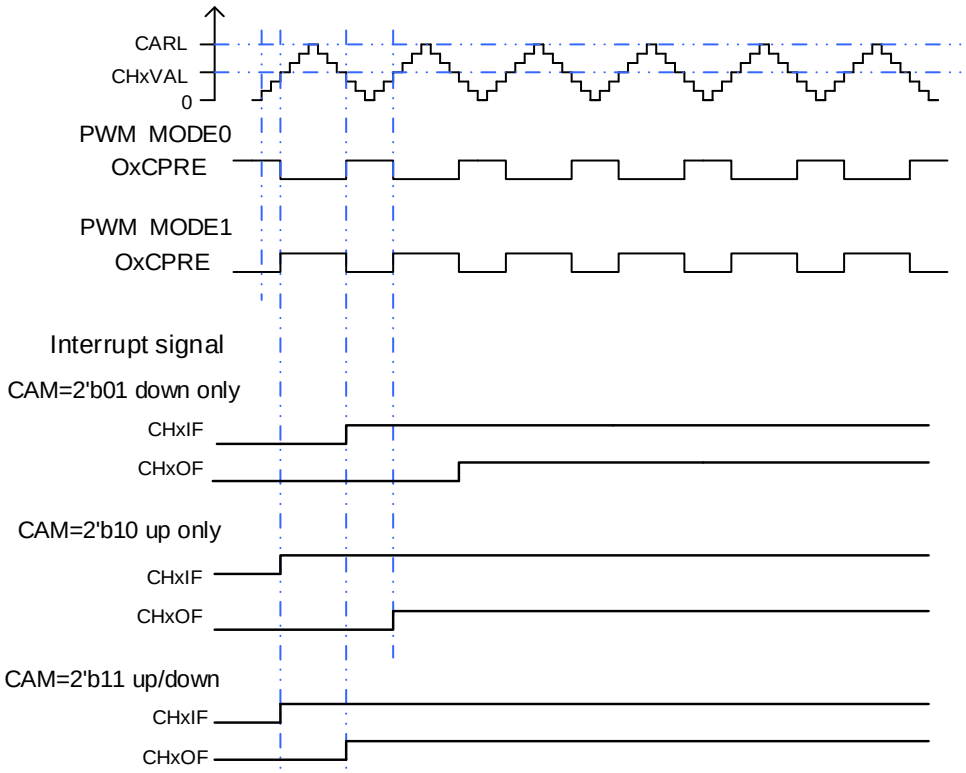


Figure 18-19. CAPWM timechart



Composite PWM mode

In the Composite PWM mode (CHxCPWMEN = 1'b1, CHxMS[2:0] = 3'b000 and CHxCOMCTL=3'b110 or 3'b111), the PWM signal output in channel x (x=0..3) is composited by CHxVAL and CHxCOMVAL_ADD bits.

If CHxCOMCTL = 3'b110 (PWM mode 0) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 3'b111 (PWM mode 1) and DIR = 1'b1 (Down counting mode), the channel x output is forced low when the counter matches the value of CHxVAL. It is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 3'b111 (PWM mode 1) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 3'b110 (PWM mode 0) and DIR = 1'b1 (down counting mode) the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when the counter matches the value of CHxCOMVAL_ADD.

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the ***Table 18-3.The Composite PWM pulse width***.

Table 18-3.The Composite PWM pulse width

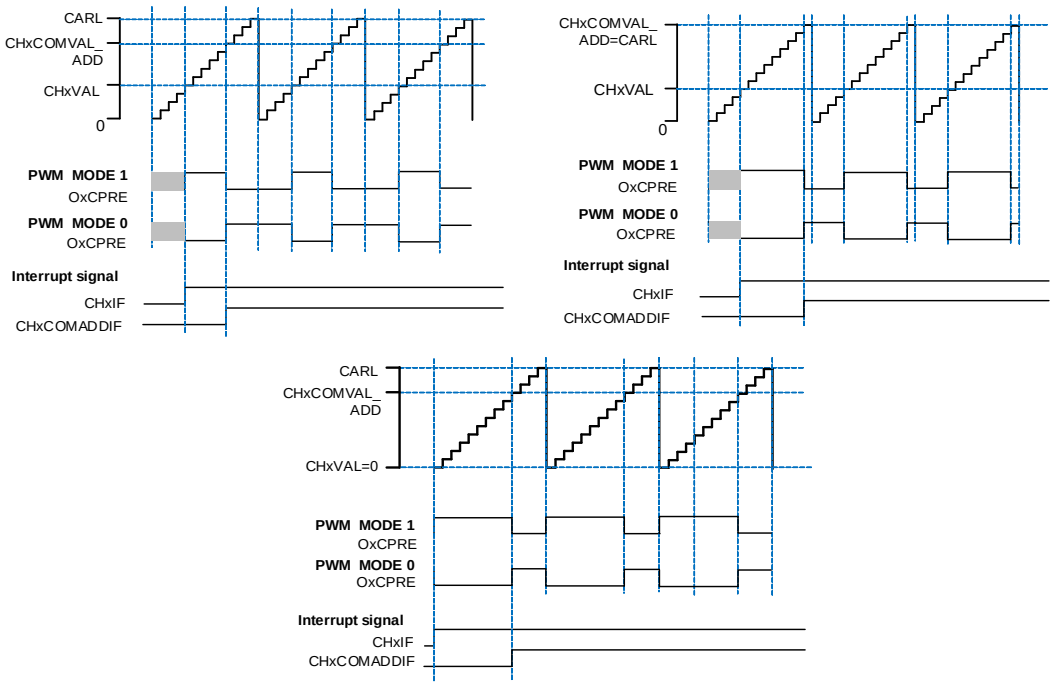
Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL – CHxCOMVAL_ADD)
	PWM mode 1	(CHxCOMVAL_ADD – CHxVAL)
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD – CHxVAL)
(CHxVAL = CHxCOMVAL_ADD ≤ CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down counting)	100%
	PWM mode 0 (down counting) or PWM mode 1 (up counting)	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when CHxCPWMEN=1) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into four situations:

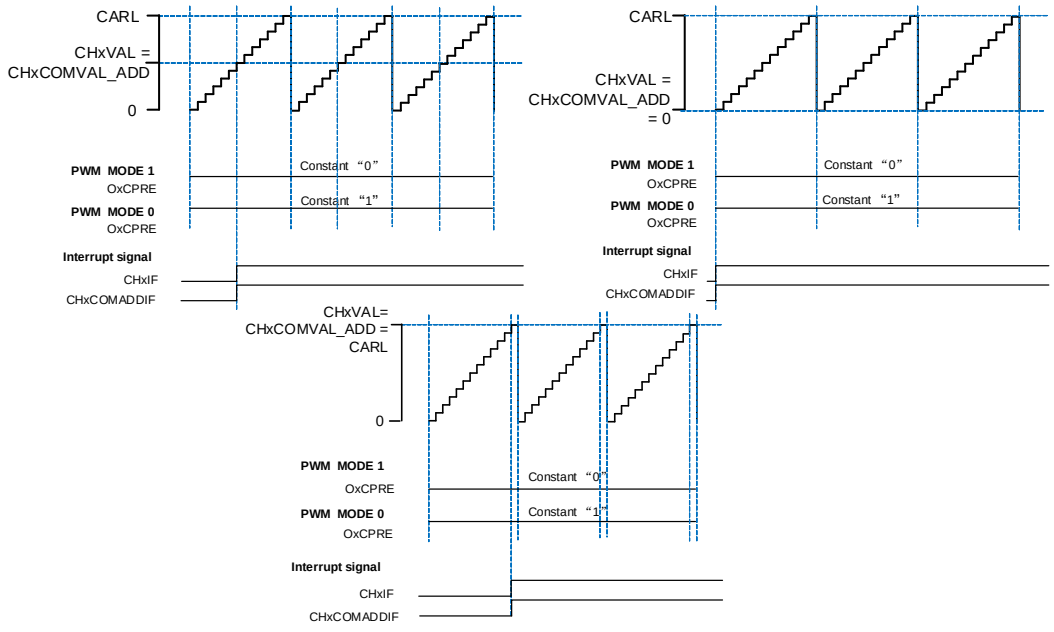
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 18-20. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



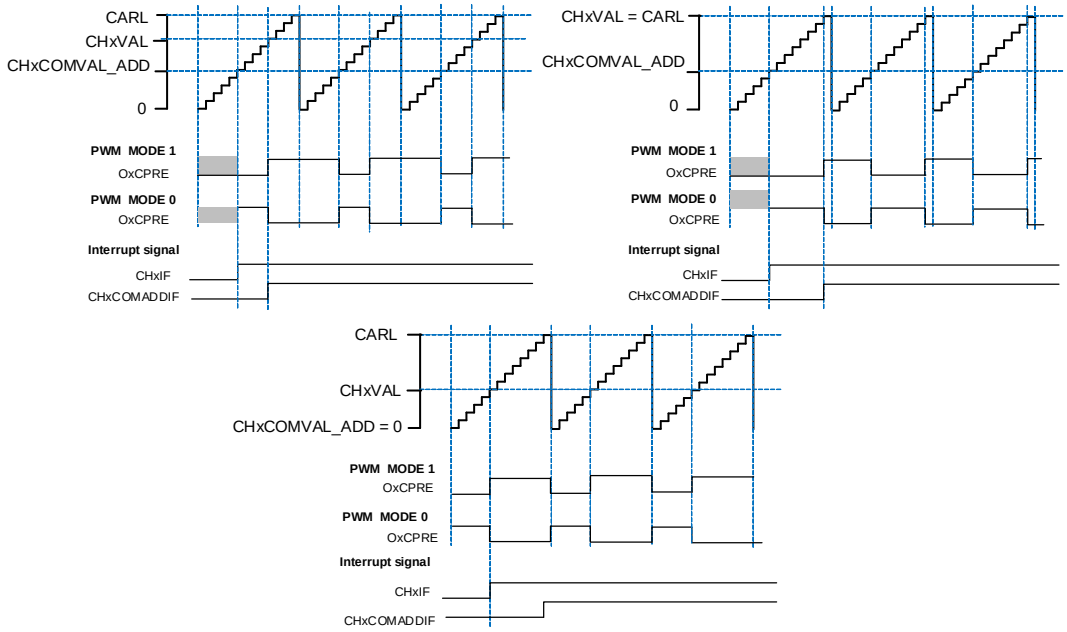
2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 18-21. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



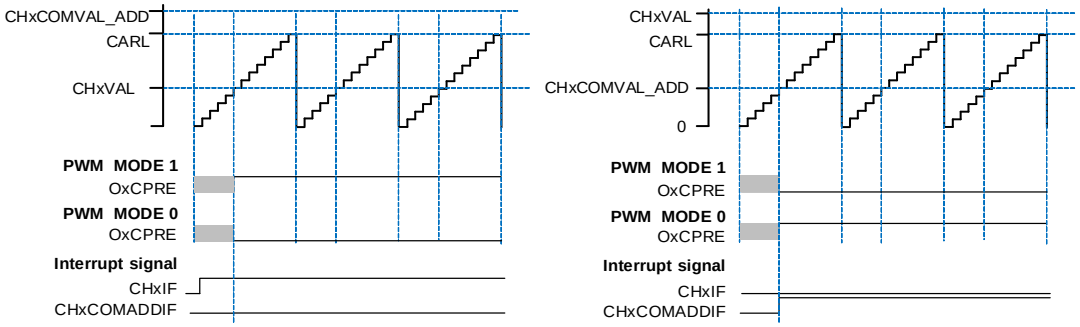
3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 18-22. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

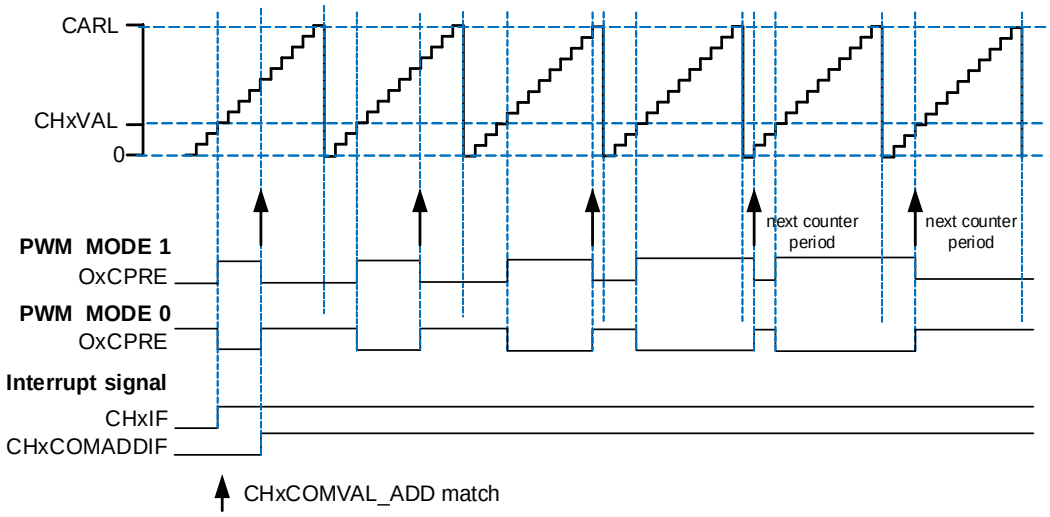
Figure 18-23. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



The composite PWM mode is intended to support the generation of PWM signals where the period is not modified while the signal is being generated, but the duty cycle will be varied. ***Figure 18-24. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD*** shows the PWM output and interrupts waveform.

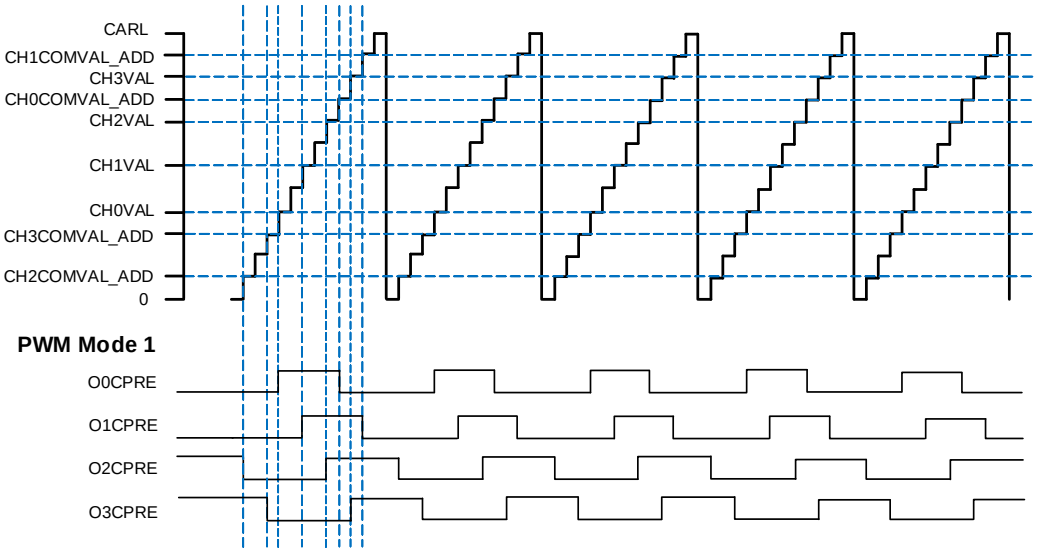
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 18-24. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD



If more than one channels are configured in composite PWM mode, it is possible to fix an offset for the channel x match edge of each pair with respect to other channels. This behavior is useful in the generation of lighting PWM control signals where it is desirable that edges are not coincident with each other pair to help eliminate noise generation. The CHxVAL register value is the shift of the PWM pulse with respect to the beginning of counter period.

Figure 18-25. Four Channels outputs in Composite PWM mode



Output match pulse select

Basing on that CHx_O(x=0...3) outputs are configured by CHxCOMCTL[2:0](x=0..3) bits when the match events occur, the output signal is configured by CHxOMPSEL[1:0](x=0..3) bit to be normal or a pulse.

When the match events occur, the CHxOMPSEL[1:0](x=0..3) bits are used to select the output of OxCPRE which drives CHx_O:

- CHxOMPSEL = 2'b00, the OxCPRE signal is output normally with the configuration of

CHxCOMCTL[2:0] bits;

- CHxOMPSEL = 2'b01, only the counter is counting up, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b10, only the counter is counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b11, both the counter is counting up and counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.

Figure 18-26. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00)

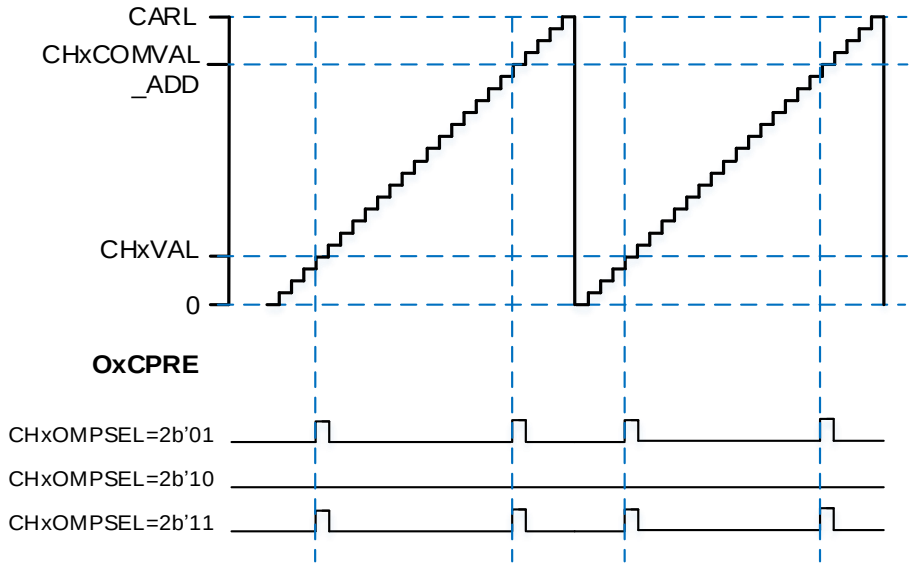
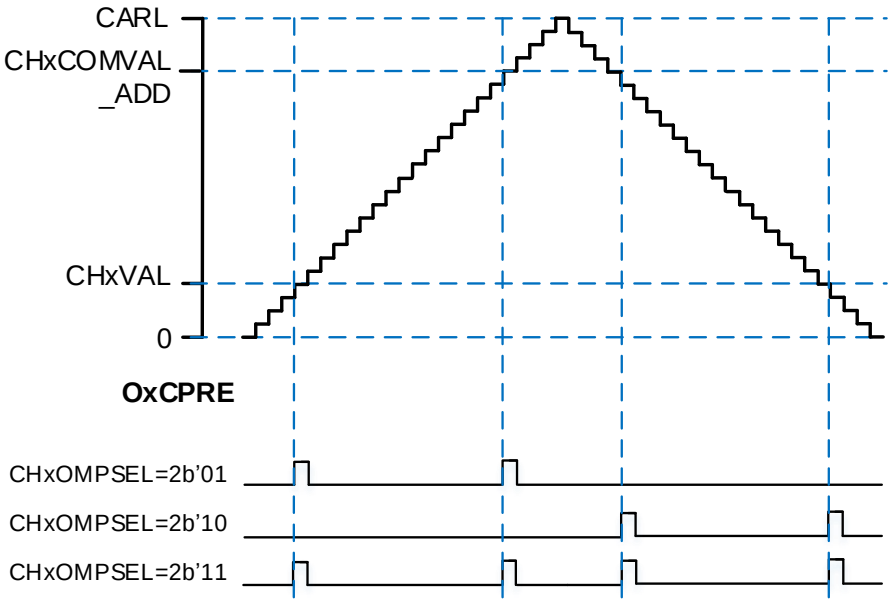


Figure 18-27. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00)



Channel output prepare signal

As is shown in ***Figure 18-14. Channel output compare principle (when MCHxMSEL = 2'00, x=0, 1, 2, 3)*** and ***Figure 18-16. Channel output compare principle (with complementary output when MCHxMSEL = 2'11, x=0,1,2,3)***, when TIMERx is configured in compare match output mode, a middle signal named OxCPRE or MOxCPRE (channel x output or multi mode channel x output prepare signal) will be generated before the channel outputs signal.

The OxCPRE and MOxCPRE signal have several types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit and the MOxCPRE signal type is defined by configuring the MCHxCOMCTL bit.

Take OxCPRE as an example for description below, these include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/ PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

Another special function of the OxCPRE signal is forced output which can be achieved by configuring the CHxCOMCTL field to 0x04/ 0x05. The output can be forced to an inactive/active level irrespective of the comparison condition between the values of the counter and the TIMERx_CHxCV.

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next update event occurs.

Channel output complementary PWM

The outputs of CHx_O and MCHx_O have three situations:

- MCHxMSEL=2'b00: The MCHx_O output is independent from the CHx_O output;
- MCHxMSEL=2'b01: The MCHx_O output is the same as the CHx_O output and the MCHxOMCTL bits are not used in the generation of the MCHx_O output;
- MCHxMSEL=2'b11: The outputs of MCHx_O and CHx_O are complementary and the MCHxOMCTL bits are not used in the generation of the MCHx_O output.

Function of complementary is for a pair of channels, CHx_O and MCHx_O, the two output signals cannot be active at the same time. The TIMERx has 4 pairs of channels, all the four pairs have this function. The complementary signals CHx_O and MCHx_O are controlled by a group of parameters: the CHxEN and MCHxEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP register(when CHx_O and MCHx_O channels has separated deadtime value and break function, please refer to ***Table 18-4.***

Complementary outputs controlled by parameters (MCHxMSEL =2'b11)), ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and MCHxP bits in the TIMERx_CHCTL2 register.

Table 18-4. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable(1).	
				1	CHx_O/ CHx_ON output "off-state" (2): the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. (3)	
			1	0		
				1		
		1	x	x	CHx_O/ CHx_ON output "off-state": the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.	
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O output disable.	
				1	CHx_O = LOW CHx_O output disable.	MCHx_O=OxCPRE⊕(4)MCHxP MCHx_O output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = LOW MCHx_O output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = (! OxCPRE) (5)⊕ MCHxP MCHx_O output enable.
	1		0	0	CHx_O = CHxP CHx_O output disable.	MCHx_O = MCHxP MCHx_O output disable.
				1	CHx_O = CHxP CHx_O output enable	MCHx_O=OxCPRE⊕MCHxP MCHx_O output enable
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable	MCHx_O = MCHxP MCHx_O output enable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable	MCHx_O = (! OxCPRE)⊕ MCHxP MCHx_O output enable.

Note:

(1) output disable: the CHx_O / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) "off-state": CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (! OxCPRE)：the complementary output of the OxCPRE signal.

Insertion dead time for complementary PWM

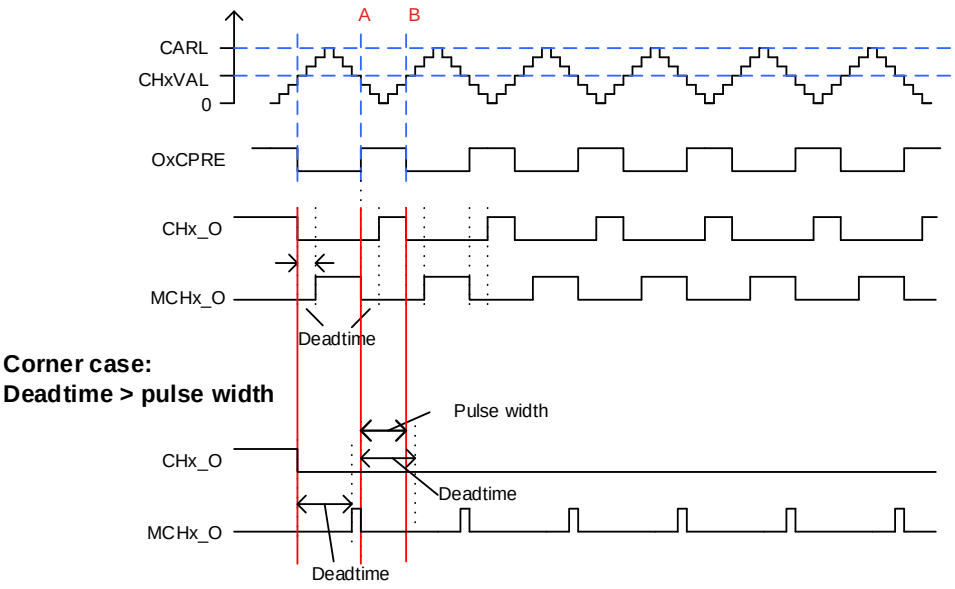
The dead time insertion is enabled when MCHxMSEL=2'b11 and both CHxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register (TIMERx_CCHP)*** for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event (TIMERx_CNT = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 18-28. Channel output complementary PWM with dead-time insertion***, CHx_O signal remains at the low level until the end of the dead time delay, while MCHx_O signal will be cleared at once. Similarly, at point B when the channelx match event (TIMERx_CNT = CHxVAL) occurs again, OxCPRE is cleared, and CHx_O signal will be cleared at once, while MCHx_O signal remains at the low level until the end of the dead time delay.

Sometimes, we can see corner cases about the dead time insertion. For example: the dead time delay is greater than or equal to the duty cycle of the CHx_O signal, then the CHx_O signal is always inactive (As shown in ***Figure 18-28. Channel output complementary PWM with dead-time insertion***).

Figure 18-28. Channel output complementary PWM with dead-time insertion



When CHx_O and MCHx_O channels has separated deadtime value, please refer to ***Separated dead time insertion and Break function***.

By configuring the DTIENCHx(x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of dead-time insertion function for each pair of channels. When the DTIENCHx(x=0..3) bit is "0", the corresponding channels CHx_O and CHx_ON will not be

inserted into the dead-time.

Break mode

In this function, CHx_O and MCHx_O are controlled by the POEN, OAEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

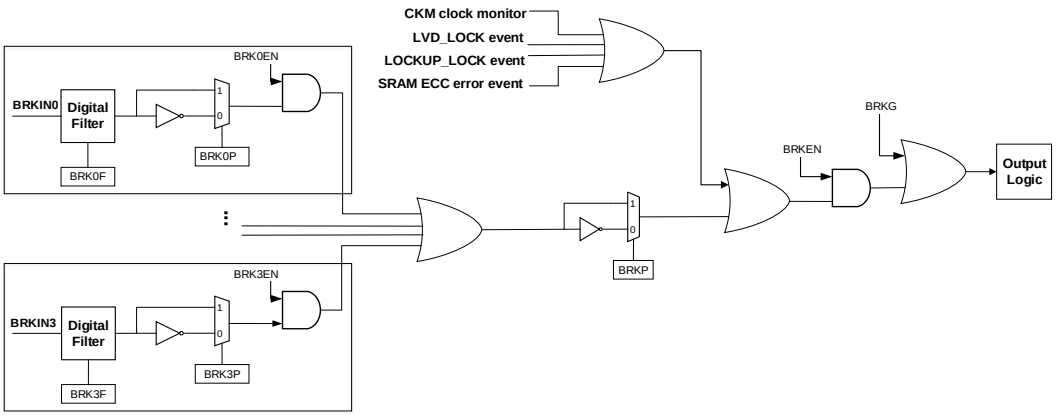
The MCHx_O output is the inverse of the CHx_O output when the MCHxMSEL=2'b11 (and the MCHxOMCTL bits are not used in the generation of the MCHx_O output). In this case, CHx_O and MCHx_O signals cannot be set to active level at the same time. The break sources are system events and other events. When the break event occurs, the outputs is force at a predefined level (either active or inactive) after a deadtime duration. The break function is enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is configured by the BRKP bit in TIMERx_CCHP register.

The break event is the result of logic ORed of all sources. The break function can handle two types of event sources:

- External sources: coming from BRKINx(x=0...3) inputs, and with digital filters and polarity selection;
- Internal sources: system sources (such as: HXTAL stuck event which is generated by Clock Monitor CKM in RCU, LVD lock event, Cortex®-M33 LOCKUP_LOCK event or SRAM ECC error event), and on-chip comparator events(configured in TRIGSEL module, input by BRKIN0 pin).

Break events can also be generated by software using BRKG bit in the TIMERx_SWEVG register.

Figure 18-29. Break function diagram



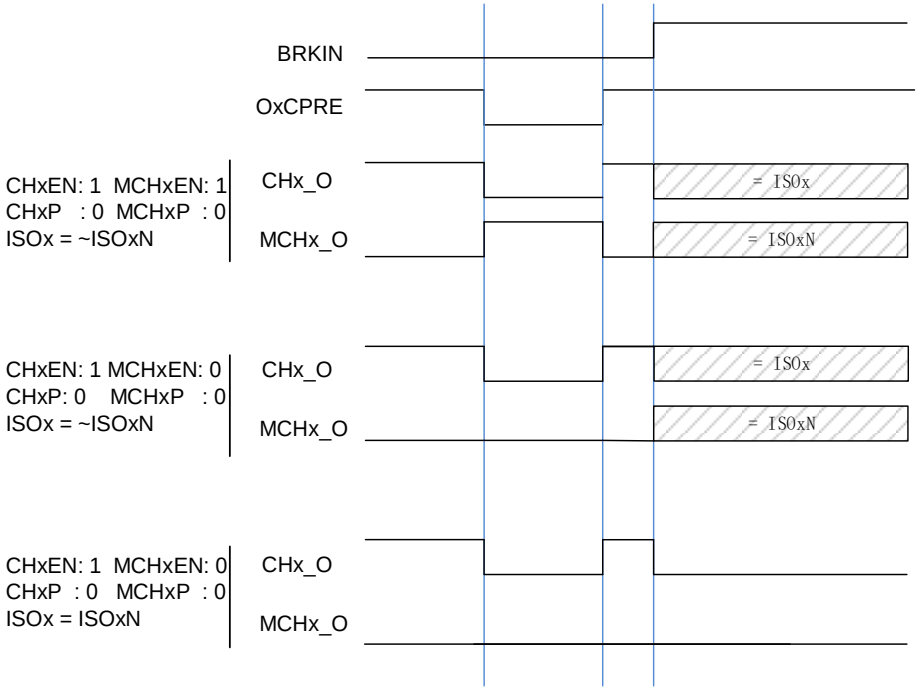
Refer to ***Figure 18-29. Break function diagram***, BRKINx(x=0..3) can from the TRIGSEL module or from the GPIO pins, which can select by ***TIMER input source select register (SYSCFG_TIMERINSEL)***.

When the MCHxMSEL = 2'b11 and a break occurs, the POEN bit is cleared asynchronously. As soon as POEN is 0, the level of the CHx_O and MCHx_O outputs are determined by the ISOx and ISOxN bits in the TIMERx_CTL1 register. If IOS = 0, the timer releases the enable

output, otherwise, the enable output remains high. The complementary outputs are first in the reset state, and then the dead time generator is reactivated to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead time.

When a break occurs, the BRKIF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

Figure 18-30. Output behavior of the channel in response to a break (the break high active)



When CHx_O and MCHx_O channels has separated break function, please refer to ***Separated dead time insertion and Break function***.

By configuring the BEKENCHx (x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of break function for each pair of channels. When the BEKENCHx(x=0..3) bit is “0” and a break event occurs, the corresponding channels CHx_O and MCHx_O will not be changed and the outputs is keeping.

Separated dead time insertion and Break function

The separated dead time insertion and break function for CHx_O and MCHx_O allows that each pair of channels has its own deadtime value and break function. In this function, CHx_O and MCHx_O are actually controlled by the IOS bit、ROS bit and DTCFG[7:0] bits in TIMERx_FCCHPy(y=0..3) register.

By configuring the FCCHPyEN (y=0...3) bits in the TIMERx_FCCHPy(y=0..3) registers can select whether each pair of channels uses the separated dead time insertion and break function. When the FCCHPyEN=0, the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active; When the FCCHPyEN=1, the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP0 register is active.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0FE0 and CI1FE1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact with each other to generate the counter value. Setting SMC=0x01, 0x02, or 0x03 to select that the counting direction of timer is determined only by the CI0FE0, only by the CI1FE1, or by the CI0FE0 and the CI1FE1. The DIR bit is modified during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in ***Table 18-5. Counting direction in different quadrature decoder mode***. The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 18-5. Counting direction in different quadrature decoder mode

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 SMC[2:0]=3'b001	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 SMC [2:0]=3'b010	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 SMC [2:0]=3'b011	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 18-31. Counter behavior with CI0FE0 polarity non-inverted in mode 2

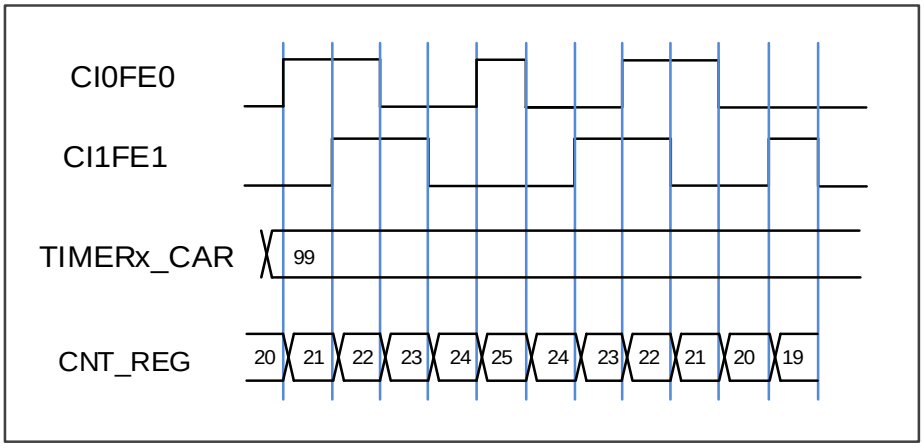
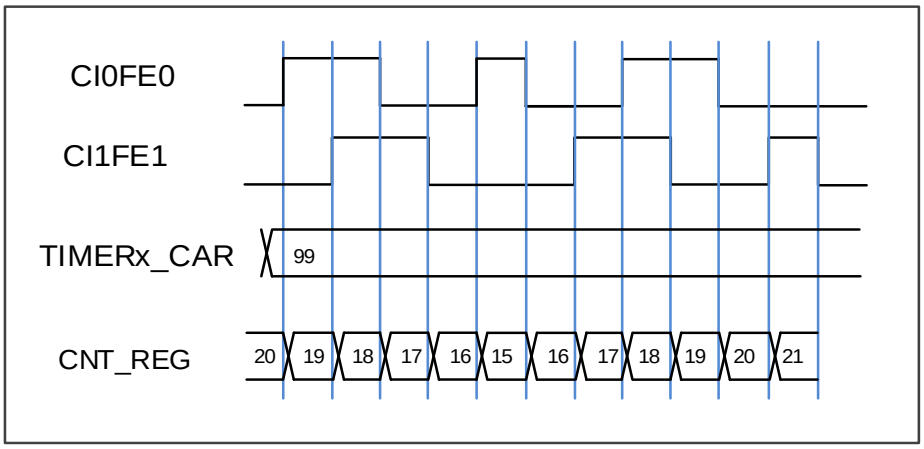


Figure 18-32. Counter behavior with CI0FE0 polarity inverted in mode 2



Hall sensor function

Hall sensor is generally used to control BLDC motor; the timers can support this function.

Figure 18-33. Hall sensor is used for BLDC motor shows how to connect the timer and the motor. And two timers are needed. TIMER_in(Advanced/General L0 TIMER）is used to accept three HALL sensor signals.

Each of the three input of HALL sensors provides a pulse which is applied to an input capture pin, then both the speed and position of rotor can be calculated by analyzing the hall sensor signals.

By the internal connection function (TRGO-ITIx), TIMER_in and TIMER_out can be connected. TIMER_out will generate PWM signals to control the speed of BLDC motor based on the ITIx. Then, the feedback circuit is finished, you can change the configuration to fit your request.

Because the advanced/general L0 TIMER has the input XOR function, they can be used as the TIMER_in timer. And the advanced timer has the functions of complementary output and dead time, so it can be used as the TIMER_out timer.

In addition, the interconnected timer can be seleted by TRIGSEL module, for example:

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

After appropriate interconnected timers are selected and wires are connected, the timers need to be configured. Some key settings are as follows:

- Enable XOR by setting TI0S, then, the change of each input signal will make the CI0 toggle. CH0VAL will record the current value of counter.
- Choose ITIx to trigger commutation by configuring CCUC and CCSE.
- Configure PWM parameters based on the requests.

Figure 18-33. Hall sensor is used for BLDC motor

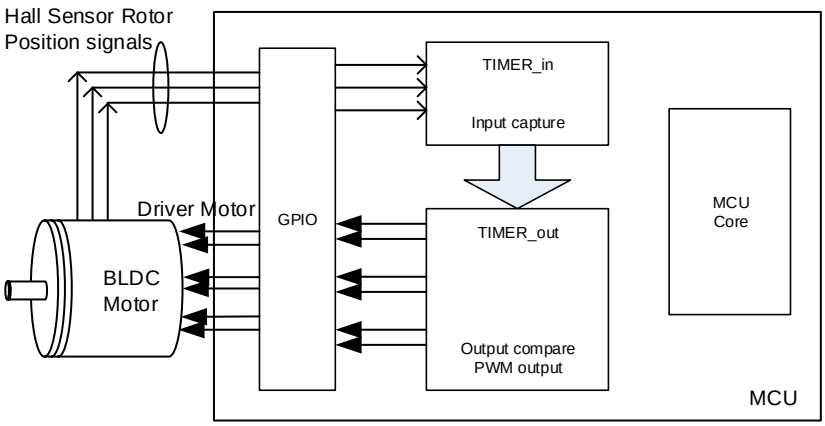
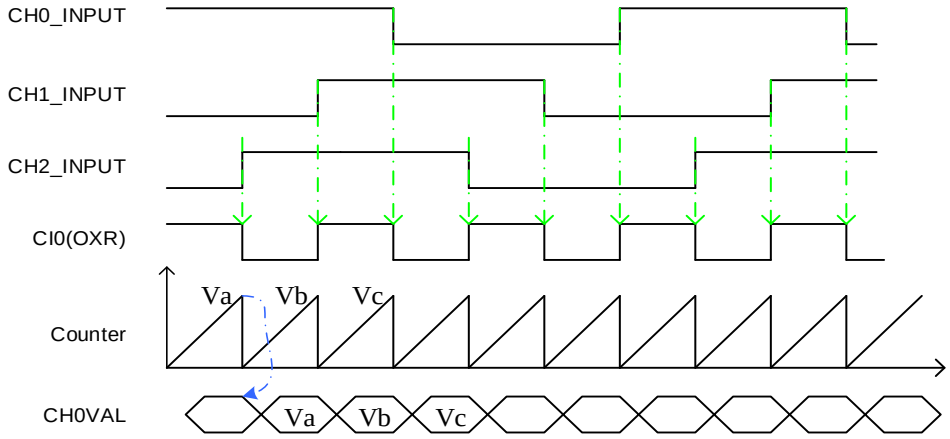
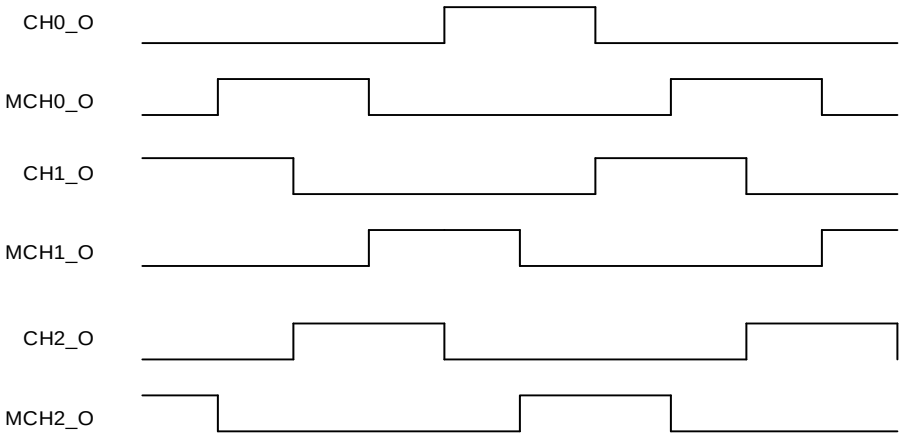


Figure 18-34. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



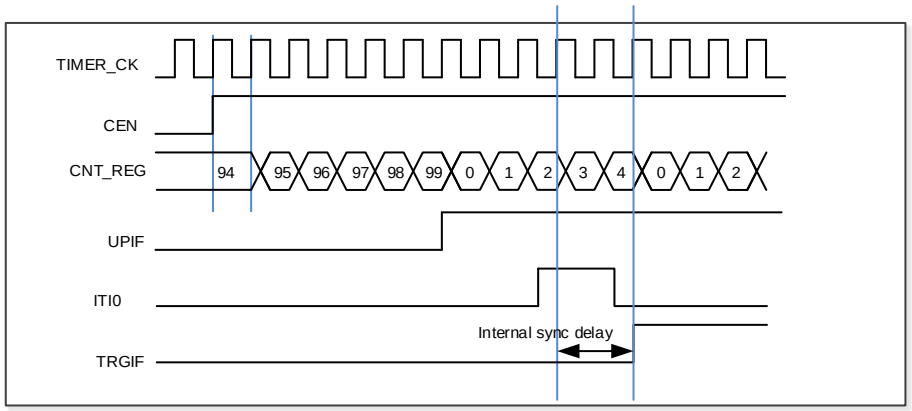
Advanced TIMER_out under output compare mode(PWM with Dead-time)

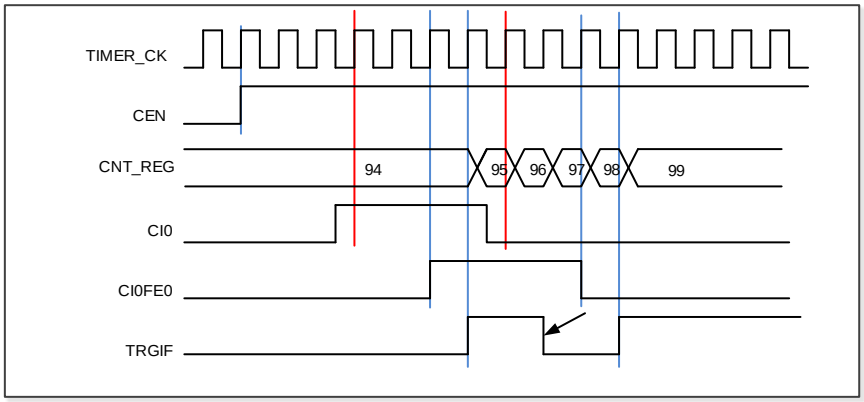
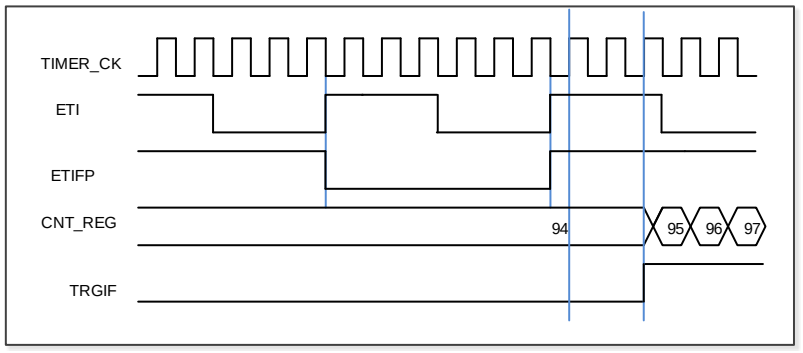


Master-slave management

The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode which is selected by the SMC[2:0] bits in the TIMERx_SMCFG register. The input trigger of these modes can be selected by the TRGS[3:0] bits in the TIMERx_SMCFG register.

Table 18-6. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	SMC[2:0] 3'b100 (restart mode) 3'b101 (pause mode) 3'b110 (event mode)	TRGS[3:0] 0000: ITI0 0001: ITI1 0010: ITI2 0011: ITI3 0100: CI0F_ED 0101: CI0FE0 0110: CI1FE1 0111: ETIFP(1) 1000: CI2FE2 1001: CI3FE3 1010: MCI0FEM0 1011: MCI1FEM1 1100: MCI2FEM2 1101: MCI3FEM3	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and MCHxP for the polarity selection and inversion. If ETIFP (the filtered output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	For the ITIx, no filter and prescaler can be used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of trigger input comes.	TRGS[3:0] = 3'b000 ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 18-35. Restart mode 			
Exam2	**Pause mode** The counter will be paused when the trigger input is low, and it will start when	TRGS[3:0]=3'b101 CI0FE0 is selected.	TI0S=0 (Non-xor) [MCH0P=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the	Filter is bypassed in this example.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	the trigger input is high.		rising edge only.	
	Figure 18-36. Pause mode 			
Exam3	**Event mode** The counter will start to count when a rising edge of trigger input comes.	TRGS[3:0] =3'b111 ETIFP is selected.	ETP = 0, the polarity of ETI does not change.	ETPSC = 1, ETI is divided by 2. ETFC = 0, ETI does not filter.
	Figure 18-37. Event mode 			

(1) The ETI pin can select from TIMER_ETIx(x=0...2) pins, and each advanced TIMER only can use one of them. Please refer to ***TIMER input source select register (SYSCFG_TIMERINSEL)*** for more details.

Single pulse mode

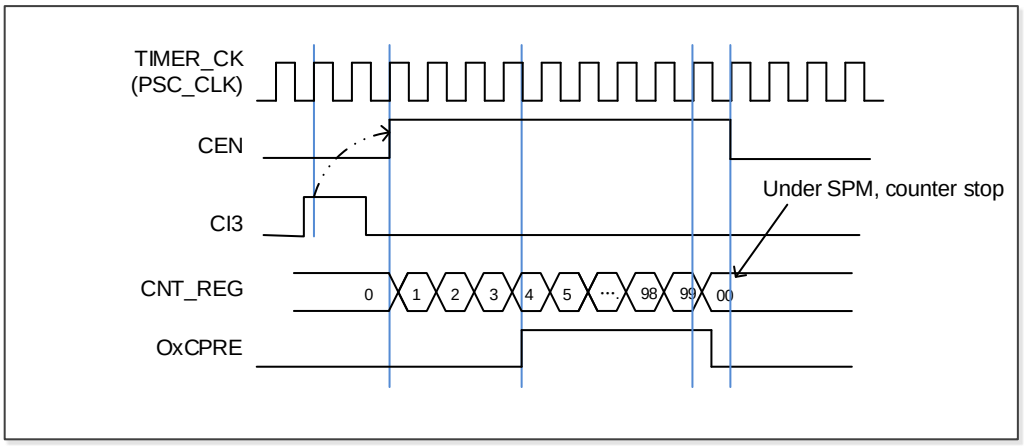
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE/ MOxCPRE signals will change to, as the compare match event occurs without taking the comparison result into account.

Single pulse mode is also applicable to composite PWM mode (CHxCPWMEN = 1'b1 and CHxMS[2:0] = 3'b000).

Figure 18-38. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x99

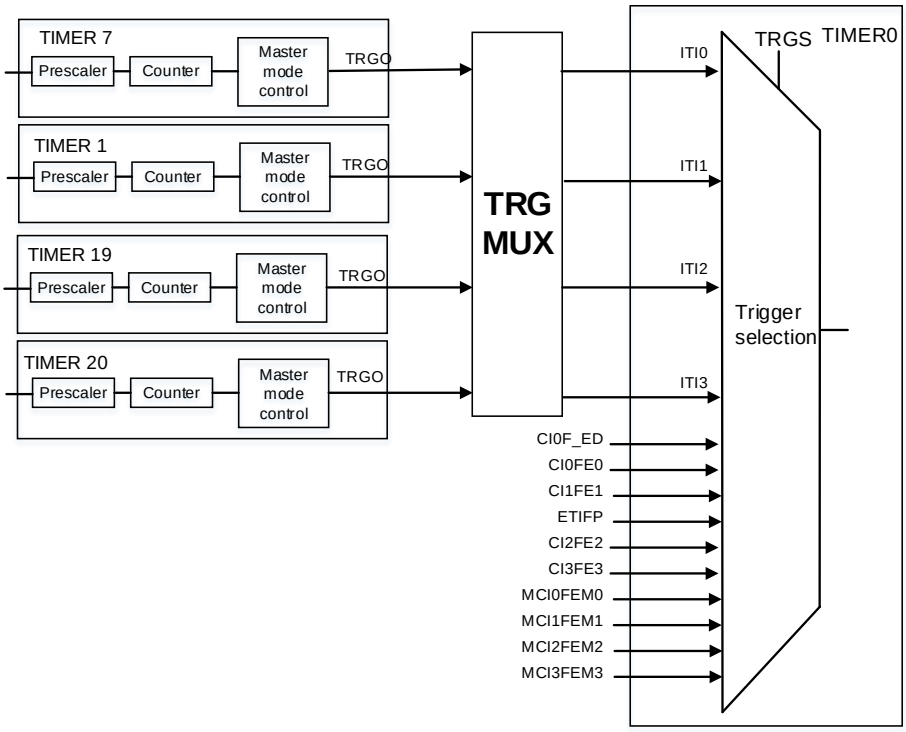


Timers interconnection

The timers can be internally connected for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the master mode while configuring another timer to be in the slave mode. The following figures show several examples of trigger selection for the master mode and slave mode.

Figure 18-39. TIMER0 master/slave mode example shows the TIMER0 trigger selection when it is configured in slave mode.

Figure 18-39. TIMER0 master/slave mode example



Other interconnection examples:

- TIMER1 as the prescaler for TIMER0

TIMER1 is configured as a prescaler for TIMER0. Refer to ***Figure 18-39. TIMER0 master/slave mode example*** for connections. Steps are shown as follows:

1. Configure TIMER1 in master mode and select its update event (UPE) as trigger output (MMC=3'b010 in the TIMER1_CTL1 register). Then TIMER1 drives a periodic signal on each counter overflow.
2. Configure TIMER1 period (TIMER1_CAR register).
3. Select TIMER1 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
4. Configure TIMER0 in external clock mode 0 (SMC=3'b111 in TIMERx_SMCFG register).
5. Start TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
6. Start TIMER1 by writing '1' to the CEN bit (TIMER1_CTL0 register).

- Start TIMER0 with TIMER1's enable signal

First, enable TIMER0 with the enable signal of TIMER1. Refer to ***Figure 18-40. Triggering TIMER0 with enable signal of TIMER1***. TIMER0 starts counting from its current value with the divided internal clock after being triggered by TIMER1 enable signal output.

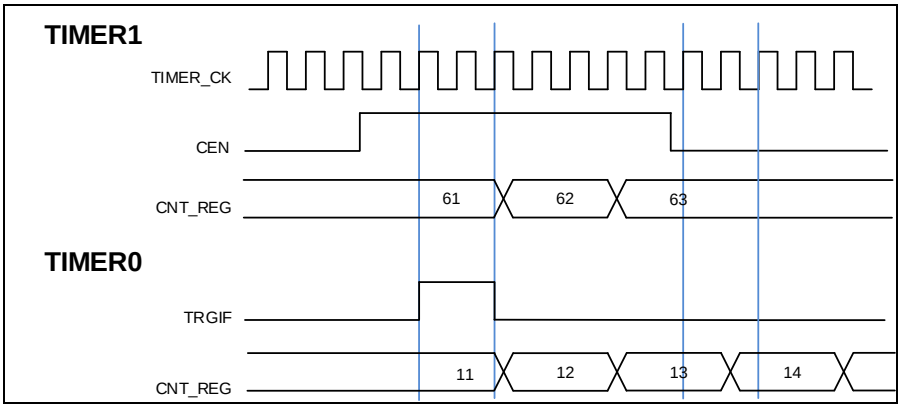
When TIMER0 receives the trigger signal, its CEN bit is set and the counter counts until TIMER0 is disabled. Both clock frequency of the counters is divided by 3 from TIMER_CK ($f_{PSC_CLK} = f_{TIMER_CK}$ /3). Steps are shown as follows:

1. Configure TIMER1 in master mode to send its enable signal as trigger output

(MMC=3'b001 in the TIMER1_CTL1 register).

2. Select TIMER1 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
3. Configure TIMER0 in event mode (SMC=3'b110 in TIMERx_SMCFG register).
4. Start TIMER1 by writing 1 to the CEN bit (TIMER1_CTL0 register).

Figure 18-40. Triggering TIMER0 with enable signal of TIMER1



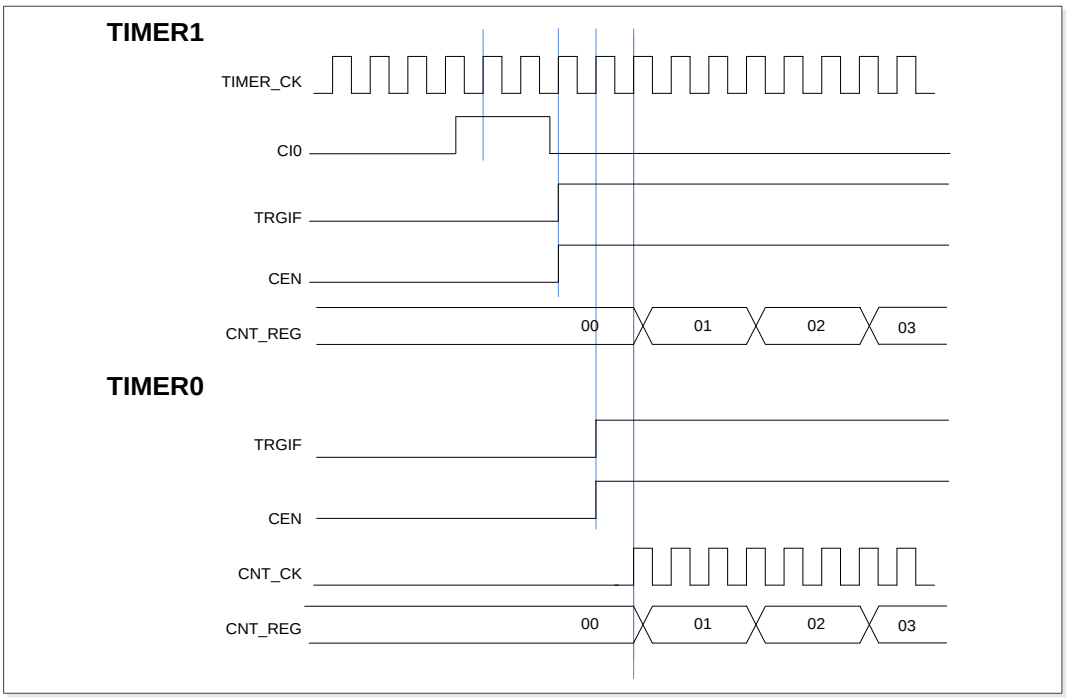
■ Using an external trigger to start two timers synchronously.

The start of TIMER0 is triggered by the enable signal of TIMER1, and TIMER1 is triggered by its CI0 input rising edge. To ensure that two timers start synchronously, TIMER1 must be configured in master/slave mode. Steps are shown as follows:

1. Configure TIMER1 in slave mode, and select CI0F_ED as the input trigger (TRGS=3'b100 in the TIMER1_SMCFG register).
2. Configure TIMER1 in event mode (SMC=3'b110 in the TIMER1_SMCFG register).
3. Configure TIMER1 in master/slave mode by writing MSM=1 (TIMER1_SMCFG register).
4. Select TIMER1 as TIMER0 input trigger source (TRGS=3'b010 in the TIMERx_SMCFG register).
5. Configure TIMER0 in event mode (SMC=3'b110 in the TIMER0_SMCFG register).

When the CI0 signal of TIMER1 generates a rising edge, two timer counters start counting synchronously with the internal clock and both TRGIF flags are set.

Figure 18-41. Triggering TIMER0 and TIMER1 with TIMER1's CI0 input



Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register is set to 1, the TIMERx counter stops.

18.1.5. Registers definition (TIMERx, x=0, 7, 19, 20)

TIMER0 base address: 0x4001 2C00

TIMER7 base address: 0x4001 3400

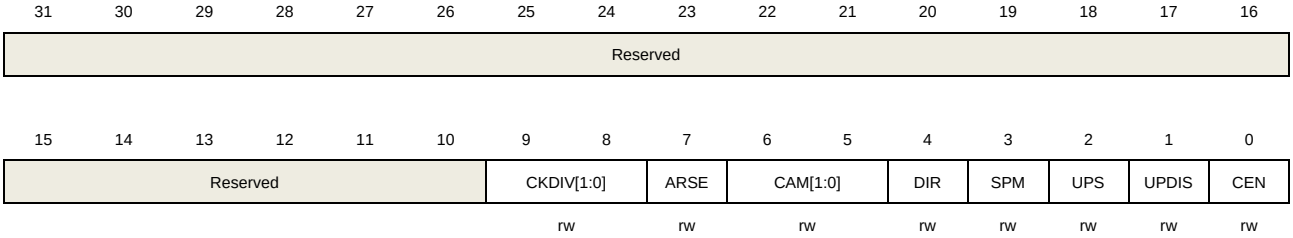
TIMER19 base address: 0x4001 5000

TIMER20 base address: 0x4001 5400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: f_{DTS}= f_{CK_TIMER} 01: f_{DTS}= f_{CK_TIMER} /2 10: f_{DTS}= f_{CK_TIMER} /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when counting down, the CHxF bit can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when counting up, the CHxF bit can be set. 11: Center-aligned and counting up/down assert mode. The counter counts in

		center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Both when counting up and counting down, the CHxF bit can be set. After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down If the timer work in center-aligned mode or decoder mode, this bit is read only.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. The counter continues after an update event. 1: Single pulse mode is enabled. The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate an update interrupt or a DMA request: – The UPG bit is set. – The counter generates an overflow or underflow event. – The restart mode generates an update event. 1: This event generates update interrupts or DMA requests: The counter generates an overflow or underflow event
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event: – The UPG bit is set. – The counter generates an overflow or underflow event. – The restart mode generates an update event. 1: Update event disable. **Note:** When this bit is set to 1, setting UPG bit or restart mode does not generate an update event, but the counter and prescaler are initialized.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISO3N	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC	Reserved	CCSE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ISO3N	Idle state of multi mode channel 3 complementary output. Refer to ISO0N bit.
14	ISO3	Idle state of channel 3 output. Refer to ISO0 bit.
13	ISO2N	Idle state of multi mode channel 2 complementary output. Refer to ISO0N bit
12	ISO2	Idle state of channel 2 output Refer to ISO0 bit
11	ISO1N	Idle state of multi mode channel 1 complementary output Refer to ISO0N bit
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of multi mode channel 0 complementary output 0: When POEN bit is reset, MCH0_O is set low. 1: When POEN bit is reset, MCH0_O is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN bit is reset, CH0_O is set low. 1: When POEN bit is reset, CH0_O is set high. The CH0_O output changes after a dead time if MCH0_O is implemented. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP register is 00.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent in master mode to slave timers for synchronization function. 000: When a counter reset event occurs, a TRGO trigger signal is output. The

		counter reset source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a counter start event occurs, a TRGO trigger signal is output. The counter start source: CEN control bit is set The trigger input in pause mode is high 010: When an update event occurs, a TRGO trigger signal is output. The update source depends on UPDIS bit and UPS bit. 011: When a capture or compare pulse event occurs in channel0, a TRGO trigger signal is output. 100: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O0CPRE. 101: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O1CPRE. 110: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O2CPRE. 111: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O3CPRE.
3	DMAS	DMA request source selection 0: When capture or compare event occurs, the DMA request of channel x is sent 1: When update event occurs, the DMA request of channel x is sent.
2	CCUC	Commutation control shadow register update control When the commutation control shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are enabled (CCSE=1), the update control of the shadow registers are shown as below: 0: The shadow registers update when CMTG bit is set. 1: The shadow registers update when CMTG bit is set or a rising edge of TRGI occurs. When a channel does not have a complementary output, this bit has no effect.
1	Reserved	Must be kept at reset value.
0	CCSE	Commutation control shadow enable 0: The shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are disabled. 1: The shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are enabled. After these bits have been written, they are updated when commutation event comes. When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRGS[3]	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			Reserved	SMC[2:0]		
rw	rw	rw		rw				rw	rw				rw		

Bits	Fields	Descriptions
31	TRGS[3]	Trigger selection. Refer to TRGS[2:0] description.
30:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal. 0: ETI is active at rising edge or high level. 1: ETI is active at falling edge or low level.
14	SMC1	Part of SMC is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled When the slave mode is configured as restart mode, pause mode or event mode, the timer can still work in the external clock 1 mode by setting this bit. But the TRGS bits must not be 3'b111 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in this register's SMC[2:0] bit field.
13:12	ETPSC[1:0]	The prescaler of external trigger The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: The prescaler is 2 10: The prescaler is 4 11: The prescaler is 8
11:8	ETFC[3:0]	External trigger filter control The external trigger can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the external trigger signal according to f_{SAMP} and record the number of times of the same level of the signal.

After reaching the filtering capacity configured by this bit-field, it is considered to be an effective level.

The filtering capability configuration is as follows:

EXTFC[3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS_CK}/2$
4'b0101	8	
4'b0110	6	$f_{DTS_CK}/4$
4'b0111	8	
4'b1000	6	$f_{DTS_CK}/8$
4'b1001	8	
4'b1010	5	$f_{DTS_CK}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS_CK}/32$
4'b1110	6	
4'b1111	8	

7 MSM

Master-slave mode

This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected.

0: Master-slave mode disabled

1: Master-slave mode enabled

6:4 TRGS[2:0]

Trigger selection

This bit-field specifies which signal is selected as the trigger input to synchronize the timers.

0000: Internal trigger input 0 (ITI0)

0001: Internal trigger input 1 (ITI1)

0010: Internal trigger input 2 (ITI2)

0011: Internal trigger input 3 (ITI3)

0100: CI0 edge flag (CI0F_ED)

0101: The filtered output of channel 0 input (CI0FE0)

0110: The filtered output of channel 1 input (CI1FE1)

0111: The filtered output of external trigger input (ETIFP)

1000：The filtered output of channel 2 input (CI2FE2)

1001：The filtered output of channel 3 input (CI3FE3)

1010：The filtered output of multi mode channel 0 input (MCI0FEM0)

1011：The filtered output of multi mode channel 1 input (MCI1FEM1)

1100：The filtered output of multi mode channel 2 input (MCI2FEM2)

1101：The filtered output of multi mode channel 3 input (MCI3FEM3)

		1110：Reserved 1111：Reserved These bits must not be changed when slave mode is enabled.
3	Reserved	Must be kept at reset value.
2:0	SMC[2:0]	Slave mode control 000: Disable slave mode. The slave mode is disabled. The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high. 001: Quadrature decoder mode 0. The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level. 010: Quadrature decoder mode 1. The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level. 011: Quadrature decoder mode 2. The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on the level of the other (CI1FE1 or CI0FE0). 100: Restart mode. The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 101: Pause mode. The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 110: Event mode. A rising edge of the trigger input enables the counter. 111: External clock mode 0. The counter counts on the rising edges of the selected trigger.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIE	CH2COMADDIE	CH1COMADDIE	CH0COMADDIE	MCH3DEN	MCH2DEN	MCH1DEN	MCH0DEN	MCH3IE	MCH2IE	MCH1IE	MCH0IE	Reserved			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3COMADDIE	Channel 3 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode (when CH3CPWMEN=1, CH3MS[2:0] = 3'b000 and CH3COMCTL=3'b110 or 3'b111).
30	CH2COMADDIE	Channel 2 additional compare interrupt enable 0: Disabled

		1: Enabled **Note:** This bit just used in composite PWM mode (when CH2CPWMEN=1, CH2MS[2:0] = 3'b000 and CH2COMCTL=3'b110 or 3'b111).
29	CH1COMADDIE	Channel 1 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode (when CH1CPWMEN=1, CH1MS[2:0] = 3'b000 and CH1COMCTL=3'b110 or 3'b111).
28	CH0COMADDIE	Channel 0 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode (when CH0CPWMEN=1, CH0MS[2:0] = 3'b000 and CH0COMCTL=3'b110 or 3'b111).
27	MCH3DEN	Multi mode channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2b'00).
26	MCH2DEN	Multi mode channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2b'00).
25	MCH1DEN	Multi mode channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH1SEL[1:0] = 2b'00).
24	MCH0DEN	Multi mode channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH0SEL[1:0] = 2b'00).
23	MCH3IE	Multi mode channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH3MSEL[1:0] = 2b'00).
22	MCH2IE	Multi mode channel 2 capture/compare interrupt enable 0: Disabled

		1: Enabled **Note:** This bit just used for channel input and output independent mode (when MCH2MSEL[1:0] = 2b'00).
21	MCH1IE	Multi mode channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH1SEL[1:0] = 2b'00).
20	MCH0IE	Multi mode channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH0SEL[1:0] = 2b'00).
19:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable 0: Disabled 1: Enabled
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled

6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	CMTIE	Commutation interrupt enable 0: Disabled 1: Enabled
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	Reserved			
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CH3OF	CH2OF	CH1OF	CH0OF	Reserved	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0	.	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31	CH3COMADDIF	Channel 3 additional compare interrupt flag. Refer to CH0COMADDIF description.
30	CH2COMADDIF	Channel 2 additional compare interrupt flag.

		Refer to CH0COMADDIF description.
29	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
28	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred **Note:** This flag just used in composite PWM mode (when CH0CPWMEN=1, CH0MS[2:0] = 3'b000 and CH0COMCTL=3'b110 or 3'b111).
27	MCH3OF	Multi mode channel 3 over capture flag Refer to MCH0OF description.
26	MCH2OF	Multi mode channel 2 over capture flag Refer to MCH0OF description.
25	MCH1OF	Multi mode channel 1 over capture flag Refer to MCH0OF description.
24	MCH0OF	Multi mode channel 0 over capture flag When multi mode channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while MCH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
23	MCH3IF	Multi mode channel 3 capture/compare interrupt flag Refer to MCH0IF description
22	MCH2IF	Multi mode channel 2 capture/compare interrupt flag Refer to MCH0IF description
21	MCH1IF	Multi mode channel 1 capture/compare interrupt flag Refer to MCH0IF description
20	MCH0IF	Multi mode channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If multi mode channel 0 is in input mode, this flag is set when a capture event occurs. If multi mode channel 0 is in output mode, this flag is set when a compare event occurs. If multi mode channel 0 is set to input mode, this bit will be reset by reading TIMERx_MCH0CV. 0: No multi mode channel 0 capture/compare interrupt occurred 1: Multi mode channel 0 capture/compare interrupt occurred
19:13	Reserved	Must be kept at reset value.

12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8	Reserved	Must be kept at reset value.
7	BRKIF	Break interrupt flag When the break input is inactive, the bit is set by hardware. When the break input is inactive, the bit can be cleared by software. 0: No active level break has been detected. 1: An active level has been detected.
6	TRGIF	Trigger interrupt flag This flag is set on trigger event and cleared by software. When in pause mode, both edges on trigger input generates a trigger event, otherwise, only an active edge on trigger input can generates a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when the commutation event of channel occurs, and cleared by software. 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4	CH3IF	Channel 3 capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs.

		If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDG	CH2COMADDG	CH1COMADDG	CH0COMADDG	Reserved				MCH3G	MCH2G	MCH1G	MCH0G	Reserved			
w	w	w	w					w	w	w	w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31	CH3COMADDG	Channel 3 additional compare event generation. Refer to CH0COMADDG description.
30	CH2COMADDG	Channel 2 additional compare event generation. Refer to CH0COMADDG description.
29	CH1COMADDG	Channel 1 additional compare event generation. Refer to CH0COMADDG description.
28	CH0COMADDG	Channel 0 additional compare event generation. This bit is set by software to generate a compare event in channel 0 additional, it is automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event 1: Generate a channel 0 additional compare event **Note:** This bit just used in composite PWM mode (when CH0CPWMEN=1, CH0MS[2:0] = 3'b000 and CH0COMCTL=3'b110 or 3'b111).
27:24	Reserved	Must be kept at reset value.
23	MCH3G	Multi mode channel 3 capture or compare event generation. Refer to MCH0G description.

22	MCH2G	Multi mode channel 2 capture or compare event generation. Refer to MCH0G description.
21	MCH1G	Multi mode channel 1 capture or compare event generation. Refer to MCH0G description.
20	MCH0G	Multi mode channel 0 capture or compare event generation. This bit is set by software to generate a capture or compare event in multi mode channel 0, it is automatically cleared by hardware. When this bit is set, the MCH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if multi mode channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_MCH0CV register, and the MCH0OF flag is set if the MCH0IF flag has been set. 0: No generate a multi mode channel 0 capture or compare event 1: Generate a multi mode channel 0 capture or compare event
19:8	Reserved	Must be kept at reset value.
7	BRKG	Break event generation This bit is set by software to generate an event and cleared by hardware automatically. When this bit is set, the POEN bit will be cleared and BRKIF flag will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a break event 1: Generate a break event
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, MCHxEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel commutation update event
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description

1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	CH1COMADDSEN	CH0COMADDSEN	Reserved											
		Reserved	Reserved												
rw	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	Reserved	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	Reserved	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register

		which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
27:16	Reserved	Must be kept at reset value.
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	Reserved	Must be kept at reset value.
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH1MS[2:0] bit-field is writable only when the channel is not active (When MCH1MSEL[1:0] = 2b'00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH1MSEL[1:0] = 2b'01 or 2b'11, the CH1EN and MCH1EN bits in TIMERx_CHCTL2 register are reset). 000: Channel 1 is programmed as output. 001: Channel 1 is programmed as input, IS1 is connected to CI1FE1. 010: Channel 1 is programmed as input, IS1 is connected to CI0FE1. 011: Channel 1 is programmed as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register). 100: Channel 1 is programmed as input, IS1 is connected to MCI1FE1. 101~111: Reserved.
7	CH0COMCEN	Channel 0 output compare clear enable When this bit is set, if the ETIFP signal is detected as high level, the O0CPRE signal will be cleared. 0: Channel 0 output compare clear disabled 1: Channel 0 output compare clear enabled
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field controls the behavior of O0CPRE which drives CH0_O. The active level of O0CPRE is high, while the active level of CH0_O depends on CH0P bit. **Note**: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, This bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O. The active level of O0CPRE is high, while the active

level of CH0_O and MCH0_O depends on CH0P and MCH0P bits.
000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT.
001: Set the channel output on match. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV.
010: Clear the channel output on match. O0CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH0CV.
011: Toggle on match. O0CPRE toggles when the counter is equals to the output compare register TIMERx_CH0CV.
100: Force low. O0CPRE is forced to low level.
101: Force high. O0CPRE is forced to high level.
110: PWM mode 0. When counting up, O0CPRE is active when the counter is smaller than TIMERx_CH0CV, otherwise it is inactive. When counting down, O0CPRE is inactive when the counter is larger than TIMERx_CH0CV, otherwise it is active.
111: PWM mode 1. When counting up, O0CPRE is inactive when the counter is smaller than TIMERx_CH0CV, otherwise it is active. When counting down, O0CPRE is active when the counter is larger than TIMERx_CH0CV, otherwise it is inactive.
NOTE: In the composite PWM mode (CH0CPWMEN = 1'b1 and CH0MS = 3'b000), the PWM signal output in channel 0 is composited by TIMERx_CH0CV and TIMERx_CH0COMV_ADD. Please refer to ***Composite PWM mode*** for more details.
If configured in PWM mode, the O0CPRE level changes only when the output compare mode is adjusted from "Timing" mode to "PWM" mode or the comparison result changes.
This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000 (compare mode).

3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0MS bit-field is 000.
2	Reserved	Must be kept at reset value.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2b'00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH0MSEL[1:0] = 2b'01 or 2b'11, the CH0EN and MCH0EN bits in

TIMERx_CHCTL2 register are reset).
000: Channel 0 is programmed as output.
001: Channel 0 is programmed as input, IS0 is connected to CI0FE0.
010: Channel 0 is programmed as input, IS0 is connected to CI1FE0.
011: Channel 0 is programmed as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).
100: Channel 0 is programmed as input, IS0 is connected to MCI0FE0.
101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Same as output compare mode.
30	CH0MS[2]	Channel 0 I/O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control The CI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

CH0CAPFLT [3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	

		4’b1010	5	f_{DTS}/16
		4’b1011	6	
		4’b1100	8	
		4’b1101	5	f_{DTS}/32
		4’b1110	6	
		4’b1111	8	

3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, input capture occurs on every channel input edge. 01: The input capture occurs on every 2 channel input edges 10: The input capture occurs on every 4 channel input edges 11: The input capture occurs on every 8 channel input edges
1:0	CH0MS[1:0]	Channel 0 I/O mode selection Same as output compare mode.

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS[2]	CH2MS[2]	CH3COMADDSEN	CH2COMADDSEN	Reserved											
		Reserved	Reserved												
rw	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	Reserved	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	Reserved	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Refer to CH3MS[1:0]description.
30	CH2MS[2]	Channel 2 I/O mode selection Refer to CH2MS[1:0] description.
29	CH3COMADDSEN	Channel 3 additional compare output shadow enable Refer to CH2COMADDSEN description.
28	CH2COMADDSEN	Channel 2 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2COMV_ADD register

		which updates at each update event will be enabled. 0: Channel 2 additional compare shadow disabled 1: Channel 2 additional compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2MS bit-field is 000.
27:16	Reserved	Must be kept at reset value.
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH0COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH0COMCTL description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH2COMSEN description
10	Reserved	Must be kept at reset value.
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[2:0] bit-field is writable only when the channel is not active (When MCH3MSEL[1:0] = 2b'00, the CH3EN bit in TIMERx_CHCTL2 register is reset; when MCH3MSEL[1:0] = 2b'01 or 2b'11, the CH3EN and MCH3EN bits in TIMERx_CHCTL2 register are reset). 00: Channel 3 is programmed as output. 01: Channel 3 is programmed as input, IS3 is connected to CI3FE3. 10: Channel 3 is programmed as input, IS3 is connected to CI2FE3. 11: Channel 3 is programmed as input, IS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register). 100: Channel 3 is programmed as input, IS3 is connected to MCI3FE3. 101~111: Reserved.
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the O2CPRE signal will be cleared. 0: Channel 2 output compare clear disabled 1: Channel 2 output compare clear enabled
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field specifies the compare output mode of the the output prepare signal O2CPRE. In addition, the high level of O2CPRE is the active level, and CH2_O and CH2_ON channels polarity depends on CH2P and CH2NP bits. This bit-field controls the behavior of O2CPRE which drives CH2_O. The active level of O2CPRE is high, while the active level of CH2_O depends on CH2P bit.

Note: When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2b'11, This bit-field controls the behavior of O2CPRE which drives CH2_O and MCH2_O. The active level of O2CPRE is high, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits.

000: Timing mode. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT.

001: Set the channel output on match. O2CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH2CV.

010: Clear the channel output on match. O2CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH2CV.

011: Toggle on match. O2CPRE toggles when the counter is equals to the output compare register TIMERx_CH2CV.

100: Force low. O2CPRE is forced to low level.

101: Force high. O2CPRE is forced to high level.

110: PWM mode 0. When counting up, O2CPRE is active when the counter is smaller than TIMERx_CH2CV, otherwise it is inactive. When counting down, O2CPRE is inactive when the counter is larger than TIMERx_CH2CV, otherwise it is active.

111: PWM mode 1. When counting up, O2CPRE is inactive when the counter is smaller than TIMERx_CH2CV, otherwise it is active. When counting down, O2CPRE is active when the counter is larger than TIMERx_CH2CV, otherwise it is inactive.

NOTE: In the composite PWM mode (CH2CPWMEN = 1'b1 and CH2MS = 3'b000), the PWM signal output in channel 2 is composited by TIMERx_CH2CV and TIMERx_CH2COMV_ADD. Please refer to ***Composite PWM mode*** for more details.

If configured in PWM mode, the O2CPRE level changes only when the output compare mode is adjusted from "Timing" mode to "PWM" mode or the comparison result changes.

This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2MS bit-field is 000(compare mode).

3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disabled 1: Channel 2 output compare shadow enabled The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2MS bit-field is 000.
2	Reserved	Must be kept at reset value.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection

This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active(When MCH2MSEL[1:0] = 2b'00, the CH2EN bit in TIMERx_CHCTL2 register is reset; when MCH2MSEL[1:0] = 2b'01 or 2b'11, the CH2EN and MCH2EN bits in TIMERx_CHCTL2 register are reset).
00: Channel 2 is programmed as output.
01: Channel 2 is programmed as input, IS2 is connected to CI2FE2.
10: Channel 2 is programmed as input, IS2 is connected to CI3FE2.
11: Channel 2 is programmed as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).
100: Channel 2 is programmed as input, IS2 is connected to MCI2FE2.
101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I/O mode selection Same as output compare mode.
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control The CI2 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI2 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

CH2CAPFLT [3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$

		4’b0101	8	
		4’b0110	6	$f_{DTS}/4$
		4’b0111	8	
		4’b1000	6	$f_{DTS}/8$
		4’b1001	8	
		4’b1010	5	$f_{DTS}/16$
		4’b1011	6	
		4’b1100	8	
		4’b1101	5	$f_{DTS}/32$
		4’b1110	6	
		4’b1111	8	

3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: The input capture occurs on every 2 channel input edges 10: The input capture occurs on every 4 channel input edges 11: The input capture occurs on every 8 channel input edges
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode.

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3P	MCH3EN	CH3P	CH3EN	MCH2P	MCH2EN	CH2P	CH2EN	MCH1P	MCH1EN	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	MCH3P	Multi mode channel 3 capture/compare polarity Refer to MCH0P description.
14	MCH3EN	Multi mode channel 3 capture/compare enable Refer to MCH0EN description.
13	CH3P	Channel 3 capture/compare polarity Refer to CH0P description.

12	CH3EN	Channel 3 capture/compare enable Refer to CH0EN description.
11	MCH2P	Multi mode channel 2 output polarity Refer to MCH0P description.
10	MCH2EN	Multi mode channel 2 output enable Refer to MCH0EN description.
9	CH2P	Channel 2 capture/compare polarity Refer to CH0P description.
8	CH2EN	Channel 2 capture/compare enable Refer to CH0EN description.
7	MCH1P	Multi mode channel 1 output polarity Refer to MCH0P description.
6	MCH1EN	Multi mode channel 1 output enable Refer to MCH0EN description.
5	CH1P	Channel 1 capture/compare polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare enable Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, this bit specifies the MCH0_O output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.
2	MCH0EN	Multi mode channel 0 capture/compare enable When multi mode channel 0 is configured in output mode, setting this bit enables MCH0_O signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits specified the channel 0 input

		signal's polarity. [MCH0P, CH0P] will select the active trigger or capture polarity for channel 0 input signals. 00: channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted/both channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter. **Note:** When the timer is configured in input capture mode, this register must be configured a non-zero value (such as 0xFFFF) which is larger than user expected value.

Counter repetition register (TIMERx_CREP)

Address offset: 0x30
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CREP[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP[7:0]	Counter repetition value This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	POEN	Primary output enable The bit can be set to 1 by: - Write 1 to this bit - If OAEN is set to 1, this bit is set to 1 at the next update event. The bit can be cleared to 0 by: - Write 0 to this bit - Valid fault input. When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and MCHx_O) if the corresponding enable bits (CHxEN, MCHxEN in TIMERx_CHCTL2 register) have been set. 0: Disable channel outputs (CHxO or CHxON). 1: Enabled channel outputs (CHxO or CHxON). **Note:** This bit is only valid when CHxMS=2'b00
14	OAEN	Output automatic enable 0: POEN bit can only be set by software. 1: POEN can be set at the next update event, if the break input is not active. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

13	BRKP	BRKINx(x = 0..3) input signals polarity This bit specifies the polarity of the BRKINx(x = 0..3) input signals. 0: BRKINx(x = 0..3) input active low 1: BRKINx(x = 0..3) input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
12	BRKEN	BRKINx(x = 0..3) input signals enable This bit can be set to enable the BRKINx(x = 0..3) and CKM clock failure event inputs. 0: Break inputs disabled 1: Break inputs enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
11	ROS	Run mode “off-state” enable When POEN bit is set (Run mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. Please refer to ***Table 18-4. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***. 0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled. 1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode “off-state” enable When POEN bit is reset (Idle mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. Please refer to ***Table 18-4. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***. 0: “off-state” disabled. If the CHxEN/CHxNEN bits are both reset, the channels are output disabled. 1: “off-state” enabled. No matter the CHxEN/CHxNEN bits, the channels are “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx/ISOxN bits in TIMERx_CTL1 register, the BRKEN/ BRKP/OAEN/DTCFG bits in TIMERx_CCHP register, the BRKxP/BRKxEN(x = 0..3) bits in TIMERx_BRKCFG register and the OAEN/DTCFG bits in TIMERx_FCCHPx (x = 0..3) register, are writing protected. 10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP/MCHxP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode),

the ROS/IOS bits in TIMERx_CCHP register and the ROS/IOS bits in TIMERx_FCCHPx (x = 0..3) register are writing protected.

11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN bits in TIMERx_CHCTL0/1 and TIMERx_MCHCTL0/1 registers (if the related channel is configured in output) are writing protected.

This bit-field can be written only once after the system reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.

7:0 DTCFG[7:0] Dead time configuration

The relationship between DTVAL value and the duration of dead-time is as follow:

DTCFG[7:5]	The duration of dead-time
3'b0xx	DTCFG[7:0] * t_{DTS_CK}
3'b10x	(64+ DTCFG[5:0]) * t_{DTS_CK} *2
3'b110	(32+ DTCFG[4:0]) * t_{DTS_CK} *8
3'b111	(32+ DTCFG[4:0]) * t_{DTS_CK} *16

Note:

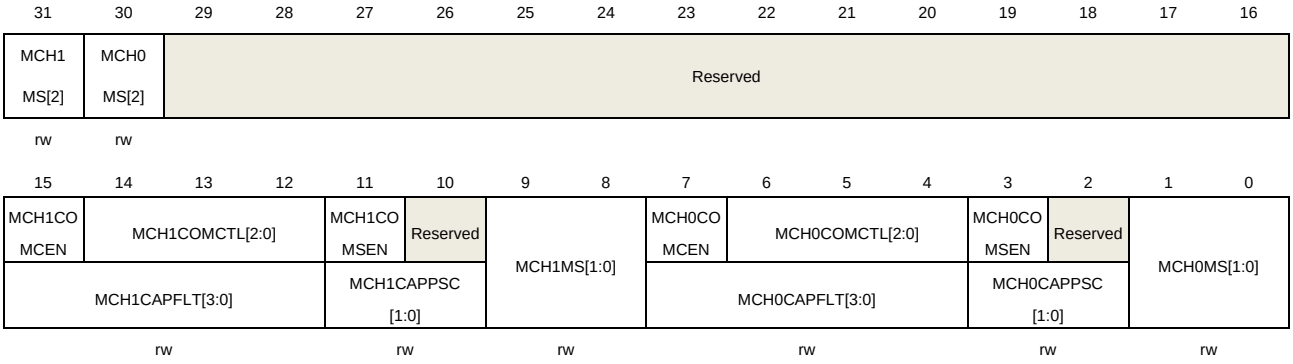
1. t_{DTS_CK} is the period of DTS_CK which is configured by CKDIV[1:0] in TIMERx_CTL0.
2. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

Multi mode channel control register 0 (TIMERx_MCHCTL0)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Output compare mode:

Bits	Fields	Descriptions
31	MCH1MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.

29:16	Reserved	Must be kept at reset value.
15	MCH1COMCEN	Multi mode channel 1 output compare clear enable. Refer to MCH0COMCEN description.
14:12	MCH1COMCTL [2:0]	Multi mode channel 1 compare output control. Refer to MCH0COMCTL description.
11	MCH1COMSEN	Multi mode channel 1 output compare shadow enable Refer to MCH0COMSEN description.
10	Reserved	Must be kept at reset value.
9:8	MCH1MS[1:0]	Multi mode channel 1 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (the MCH1EN bit in TIMERx_CHCTL2 register is reset) 000: Multi mode channel 1 is programmed as output. 001: Multi mode channel 1 is programmed as input, MIS1 is connected to MCI1FEM1. 010: Multi mode channel 1 is programmed as input, MIS1 is connected to MCI0FEM1. 011: Multi mode channel 1 is programmed as input, MIS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register). 100: Multi mode channel 1 is programmed as input, MIS1 is connected to CI1FEM1. 101~111: Reserved.
7	MCH0COMCEN	Multi mode channel 0 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the MO0CPRE signal will be cleared. 0: Multi mode channel 0 output compare clear disabled. 1: Multi mode channel 0 output compare clear enabled.
6:4	MCH0COMCTL [2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'00, this bit-field controls the behavior of MO0CPRE which drives MCH0_O. The active level of MO0CPRE is high, while the active level of MCH0_O depends on MCH0FP[1:0] bits. **Note**: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'11, the CH0COMCTL[2:0] bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 000: Timing mode. The MO0CPRE signal keeps stable, independent of the comparison between the register TIMERx_MCH0CV and the counter TIMERx_CNT. 001: Set the channel output on match. MO0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_MCH0CV.

		010: Clear the channel output on match. MO0CPRE signal is forced low when the counter is equals to the output compare register TIMERx_MCH0CV. 011: Toggle on match. MO0CPRE toggles when the counter is equals to the output compare register TIMERx_MCH0CV. 100: Force low. MO0CPRE is forced to low level. 101: Force high. MO0CPRE is forced to high level. 110: PWM mode 0. When counting up, MO0CPRE is active as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is inactive. When counting down, MO0CPRE is inactive as long as the counter is larger than TIMERx_MCH0CV, otherwise it is active. 111: PWM mode 1. When counting up, MO0CPRE is inactive as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is active. When counting down, MO0CPRE is active as long as the counter is larger than TIMERx_MCH0CV, otherwise it is inactive. If configured in PWM mode, the MO0CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH0NMS bit-field is 00(compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and MCH0MS bit-field is 00.
2	Reserved	Must be kept at reset value.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH0EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 0 is programmed as output. 001: Multi mode channel 0 is programmed as input, MIS0 is connected to MCI0FEM0. 010: Multi mode channel 0 is programmed as input, MIS0 is connected to MCI1FEM0. 011: Multi mode channel 0 is programmed as input, MIS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register). 100: Multi mode channel 0 is programmed as input, MIS0 is connected to CI0FEM0.

101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	MCH1MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH1MS[1:0]description.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15:12	MCH1CAPFLT[3:0]	Multi mode channel 1 input capture filter control. Refer to MCH0CAPFLT description.
11:10	MCH1CAPPSC[1:0]	Multi mode channel 1 input capture prescaler. Refer to MCH0CAPPSC description.
9:8	MCH1MS[1:0]	Multi mode channel 1 I/O mode selection. Same as output compare mode.
7:4	MCH0CAPFLT[3:0]	Multi mode channel 0 input capture filter control. The MCI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the MCI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

MCH0CAPFLT**[3:0]**	**Times**	**f_{SAMP}**
4’b0000	Filter disabled.	
4’b0001	2	f_{TIMER_CK}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2	MCH0CAPPSC[1:0]	Multi mode channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when MCH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disable, capture occurs on every active edge of the input signal 01: The capture input prescaler factor is 2. 10: The capture input prescaler factor is 4. 11: The capture input prescaler factor is 8.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection Same as output compare mode

Multi mode channel control register 1 (TIMERx_MCHCTL1)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCH3MS[2]	MCH2MS[2]	Reserved													
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3COMCEN	MCH3COMCTL[2:0]			MCH3COMSEN	Reserved	MCH3MS[1:0]		MCH2COMCEN	MCH2COMCTL[2:0]			MCH2COMSEN	Reserved	MCH2MS[1:0]	
MCH3CAPFLT[3:0]				MCH3CAPPSC[1:0]				MCH2CAPFLT[3:0]				MCH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31	MCH3MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH3MS[1:0]description.
30	MCH2MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH2MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15	MCH3COMCEN	Multi mode channel 3 output compare clear enable Refer to MCH2COMCEN description
14:12	MCH3COMCTL[2:0]	Multi mode channel 3 compare output control Refer to MCH2COMCTL description
11	MCH3COMSEN	Multi mode channel 3 output compare shadow enable Refer to MCH2COMSEN description
10	Reserved	Must be kept at reset value.
9:8	MCH3MS[1:0]	Multi mode channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection.

This bit-field is writable only when the channel is not active (MCH3EN bit in TIMERx_CHCTL2 register is reset).
000: Multi mode channel 3 is programmed as output.
01: Multi mode channel 3 is programmed as input, MIS3 is connected to MCI3FEM3.
010: Multi mode channel 3 is programmed as input, MIS3 is connected to MCI2FEM3.
011: Multi mode channel 3 is programmed as input, MIS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).
100: Multi mode channel 3 is programmed as input, MIS3 is connected to CI3FEM3.
101~111: Reserved.

7	MCH2COMCEN	Multi mode channel 2 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the MO2CPRE signal will be cleared. 0: Multi mode channel 2 output compare clear disabled 1: Multi mode channel 2 output compare clear enabled
6:4	MCH2COMCTL[2:0]	Multi mode channel 2 compare output control When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2b'00, this bit-field controls the behavior of MO2CPRE which drives MCH2_O. The active level of MO2CPRE is high, while the active level of MCH2_O depends on MCH2FP[1:0] bits. **Note**: When multi mode channel 2 is configured in output mode, and the MCH2MSEL[1:0] = 2b'11, the CH2COMCTL[2:0] bit-field controls the behavior of O2CPRE which drives CH2_O and MCH2_O, while the active level of CH2_O and MCH2_O depends on CH2P and MCH2P bits. 000: Timing mode. The MO2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_MCH2CV and the counter TIMERx_CNT. 001: Set the channel output on match. MO2CPRE signal is forced high when the counter is equals to the output compare register TIMERx_MCH2CV. 010: Clear the channel output on match. MO2CPRE signal is forced low when the counter is equals to the output compare register TIMERx_MCH2CV. 011: Toggle on match. MO2CPRE toggles when the counter is equals to the output compare register TIMERx_MCH2CV. 100: Force low. MO2CPRE is forced to low level. 101: Force high. MO2CPRE is forced to high level. 110: PWM mode 0. When counting up, MO2CPRE is active as long as the counter is smaller than TIMERx_MCH2CV, otherwise it is inactive. When counting down, MO2CPRE is inactive as long as the counter is larger than TIMERx_MCH2CV, otherwise it is active. 111: PWM mode 1. When counting up, MO2CPRE is inactive as long as the counter is smaller than TIMERx_MCH2CV, otherwise it is active. When counting down, MO2CPRE is active as long as the counter is larger than TIMERx_MCH2CV,

		otherwise it is inactive. If configured in PWM mode, the MO2CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2NMS bit-field is 00(compare mode).
3	MCH2COMSEN	Multi mode channel 2 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH2CV register, which updates at each update event will be enabled. 0: Multi mode channel 2 output compare shadow disabled 1: Multi mode channel 2 output compare shadow enabled The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 and CH2NMS bit-field is 00.
2	Reserved	Must be kept at reset value.
1:0	MCH2MS[1:0]	Multi mode channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH2EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 2 is programmed as output. 001: Multi mode channel 2 is programmed as input, MIS2 is connected to MCI2FEM2. 010: Multi mode channel 2 is programmed as input, MIS2 is connected to MCI3FEM2. 011: Multi mode channel 2 is programmed as input, MIS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register). 100: Multi mode channel 2 is programmed as input, MIS2 is connected to CI2FEM2. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	MCH3MS[2]	Multi mode channel 1 I/O mode selection Refer to MCH3MS[1:0]description.
30	MCH2MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH2MS[1:0] description.
29:16	Reserved	Must be kept at reset value.
15:12	MCH3CAPFLT[3:0]	Multi mode channel 3 input capture filter control. Refer to MCH2CAPFLT description.
11:10	MCH3CAPPSC[1:0]	Multi mode channel 3 input capture prescaler.

Refer to MCH2CAPPSC description.

9:8 MCH3MS[1:0] Multi mode channel 3 I/O mode selection.
Same as output compare mode.

7:4 MCH2CAPFLT[3:0] Multi mode channel 2 input capture filter control.
The MCI2 input signal can be filtered by digital filter and this bit-field configure the filtering capability.
Basic principle of digital filter: continuously sample the MCI2 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level.
The filtering capability configuration is as follows:

MCH2CAPFLT[3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{TIMER_CK}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 MCH2CAPPSC[1:0] Multi mode channel 2 input capture prescaler.
This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when MCH2EN bit in TIMERx_CHCTL2 register is cleared.
00: Prescaler disable, capture occurs on every active edge of the input signal
01: The capture input prescaler factor is 2.
10: The capture input prescaler factor is 4.
11: The capture input prescaler factor is 8.

1:0 MCH2MS[1:0] Multi mode channel 2 I/O mode selection
Same as output compare mode.

Multi mode channel control register 2 (TIMERx_MCHCTL2)

Address offset: 0x50
Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

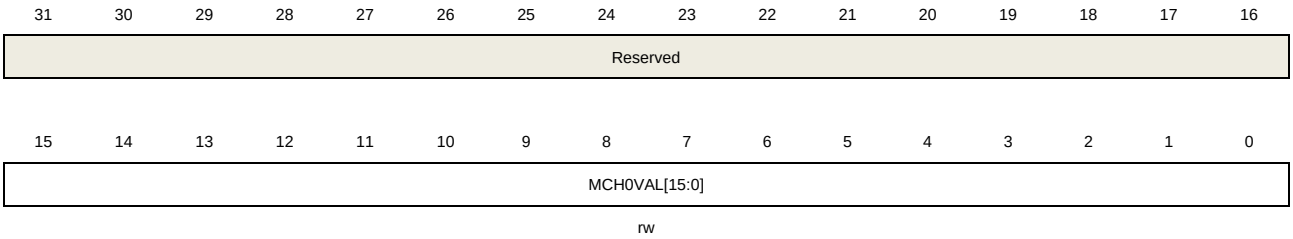
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MCH3FP[1:0]		MCH2FP[1:0]		MCH1FP[1:0]		MCH0FP[1:0]	
								rw		rw		rw		rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:6	MCH3FP[1:0]	Multi mode channel 3 capture/compare free polarity Refer to MCH0FP[1:0] description.
5:4	MCH2FP[1:0]	Multi mode channel 2 capture/compare free polarity Refer to MCH0FP[1:0] description.
3:2	MCH1FP[1:0]	Multi mode channel 1 capture/compare free polarity Refer to MCH0FP[1:0] description.
1:0	MCH0FP[1:0]	Multi mode channel 0 capture/compare free polarity When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2b'00, these bits specified the multi mode channel 0 output signal polarity. 00: Multi mode channel 0 active high 01: Multi mode channel 0 active low 10: Reserved. 11: Reserved. When multi mode channel 0 is configured in input mode, these bits specified the multi mode channel 0 input signal's polarity. MCH0FP[1:0] will select the active trigger or capture polarity for multi mode channel 0 input signals. 00: Multi mode channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And multi mode channel 0 input signal will not be inverted. 01: Multi mode channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And multi mode channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted/both multi mode channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.

Multi mode channel 0 capture/compare value register (TIMERx_MCH0CV)

Address offset: 0x54
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



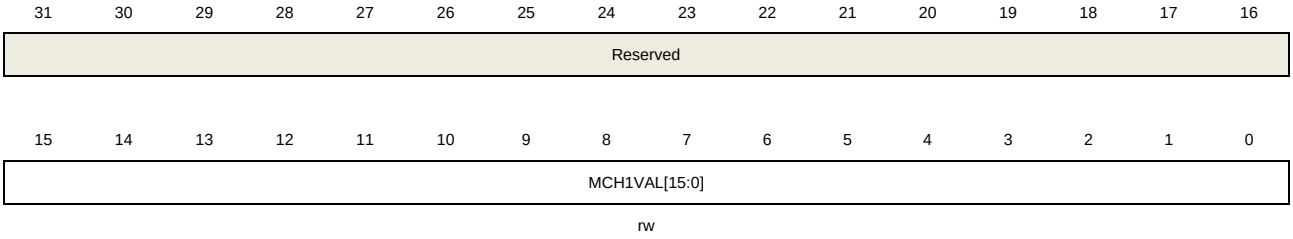
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH0VAL[15:0]	Capture/compare value of multi mode channel 0. When multi mode channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Multi mode channel 1 capture/compare value register (TIMERx_MCH1CV)

Address offset: 0x58

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



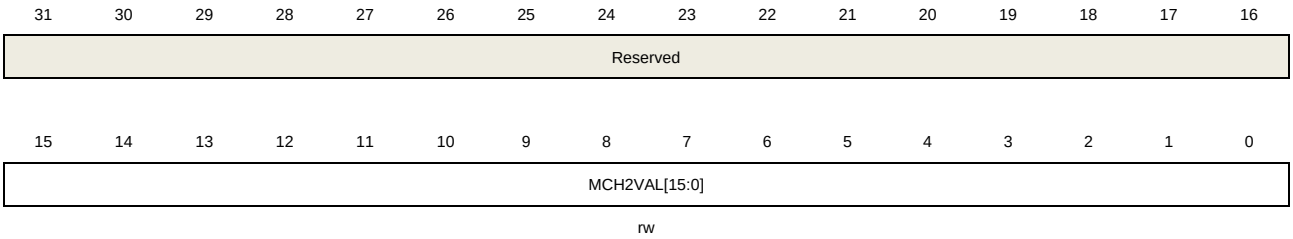
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH1VAL[15:0]	Capture/compare value of multi mode channel 1. When multi mode channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Multi mode channel 2 capture/compare value register (TIMERx_MCH2CV)

Address offset: 0x5C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



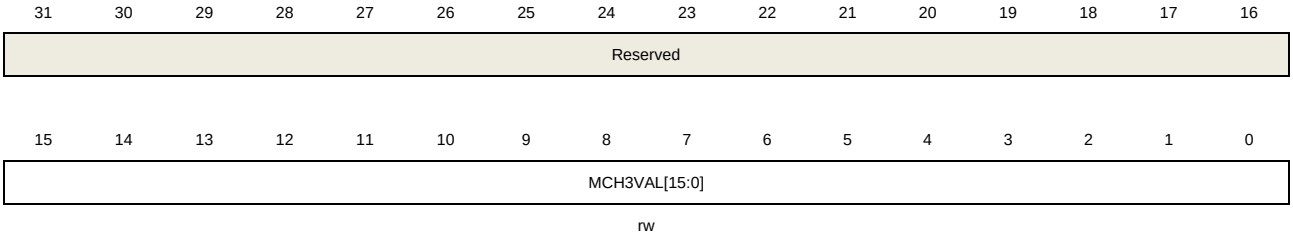
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH2VAL[15:0]	Capture/compare value of multi mode channel 2. When multi mode channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Multi mode channel 3 capture/compare value register (TIMERx_MCH3CV)

Address offset: 0x60

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH3VAL[15:0]	Capture/compare value of channel 3. When multi mode channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 0 additional compare value register (TIMERx_CH0COMV_ADD)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0COMVAL_ADD [15:0]	Additional compare value of channel 0 When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1).

Channel 1 additional compare value register (TIMERx_CH1COMV_ADD)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH1CPWMEN=1).

Channel 2 additional compare value register (TIMERx_CH2COMV_ADD)

Address offset: 0x6C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2COMVAL_ADD [15:0]	Additional compare value of channel 2 When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH2CPWMEN=1).

Channel 3 additional compare value register (TIMERx_CH3COMV_ADD)

Address offset: 0x70

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3COMVAL_ADD [15:0]	Additional compare value of channel 3 When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH3CPWMEN=1).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0FF0 00FF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3C PWMEN	CH2C PWMEN	CH1C PWMEN	CH0C PWMEN	MCH3MSEL[1:0]		MCH2MSEL[1:0]		MCH1MSEL[1:0]		MCH0MSEL[1:0]		Reserved			
rw	rw	rw	rw	rw		rw		rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		BRKEN CH3	BRKEN CH2	BRKEN CH1	BRKEN CH0	DTIEN CH3	DTIEN CH2	DTIEN CH1	DTIEN CH0
rw		rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3CPWMEN	Channel 3 composite PWM mode enable 0: Disabled 1: Enabled
30	CH2CPWMEN	Channel 2 composite PWM mode enable 0: Disabled 1: Enabled
29	CH1CPWMEN	Channel 1 composite PWM mode enable 0: Disabled 1: Enabled
28	CH0CPWMEN	Channel 0 composite PWM mode enable 0: Disabled 1: Enabled
27:26	MCH3MSEL[1:0]	Multi mode channel 3 mode select 00: Independent mode, MCH3 is independent of CH3 01: Mirrored mode, just used for output, the MCH3 output is the same as CH3 output 10: Reserved 11: Complementary mode, only the CH3 is valid for input, and the outputs of MCH3 and CH3 are complementary
25:24	MCH2MSEL[1:0]	Multi mode channel 2 mode select 00: Independent mode, MCH2 is independent of CH2 01: Mirrored mode, just used for output, the MCH2 output is the same as CH2 output 10: Reserved 11: Complementary mode, only the CH2 is valid for input, and the outputs of MCH2 and CH2 are complementary
23:22	MCH1MSEL[1:0]	Multi mode channel 1 mode select 00: Independent mode, MCH1 is independent of CH1 01: Mirrored mode, just used for output, the MCH1 output is the same as CH1 output 10: Reserved 11: Complementary mode, only the CH1 is valid for input, and the outputs of MCH1 and CH1 are complementary
21:20	MCH0MSEL[1:0]	Multi mode channel 0 mode select 00: Independent mode, MCH0 is independent of CH0 01: Mirrored mode, just used for output, the MCH0 output is the same as CH0 output 10: Reserved 11: Complementary mode, only the CH0 is valid for input, and the outputs of MCH0

		and CH0 are complementary
19:16	Reserved	Must be kept at reset value.
15:14	CH3OMPSEL[1:0]	Channel 3 output match pulse select When the match events occur, this bit is used to select the output of O3CPRE which drives CH3_O. 00: The O3CPRE signal is output normally with the configuration of CH3COMCTL[2:0] bits. 01: Only the counter is counting up, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle. 10: Only the counter is counting down, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle. 11: Both the counter is counting up and counting down, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
13:12	CH2OMPSEL[1:0]	Channel 2 output match pulse select When the match events occurs, this bit is used to select the output of O2CPRE which drives CH2_O. 00: The O2CPRE signal is output normal with the configuration of CH2COMCTL[2:0] bits. 01: Only when the counter is counting up, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
11:10	CH1OMPSEL[1:0]	Channel 1 output match pulse select When the match events occur, this bit is used to select the output of O1CPRE which drives CH1_O. 00: The O1CPRE signal is output normal with the configuration of CH1COMCTL[2:0] bits. 01: Only when the counter is counting up, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
9:8	CH0OMPSEL[1:0]	Channel 0 output match pulse select When the match events occur, this bit is used to select the output of O0CPRE which drives CH0_O. 00: The O0CPRE signal is output normal with the configuration of

		CH0COMCTL[2:0] bits. 01: Only when the counter is counting up, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
7	BRKENCH3	Break control enable for channel 3 0: Disabled 1: Enabled
6	BRKENCH2	Break control enable for channel 2 0: Disabled 1: Enabled
5	BRKENCH1	Break control enable for channel 1 0: Disabled 1: Enabled
4	BRKENCH0	Break control enable for channel 0 0: Disabled 1: Enabled
3	DTIENCH3	Dead time inserted enable for channel 3 Enables the deadtime insertion in the outputs of MCH3_O and CH3_O. 0: Disabled 1: Enabled
2	DTIENCH2	Dead time inserted enable for channel 2 Enables the deadtime insertion in the outputs of MCH2_O and CH2_O. 0: Disabled 1: Enabled
1	DTIENCH1	Dead time inserted enable for channel 1 Enables the deadtime insertion in the outputs of MCH1_O and CH1_O. 0: Disabled 1: Enabled
0	DTIENCH0	Dead time inserted enable for channel 0 Enables the deadtime insertion in the outputs of MCH0_O and CH0_O. 0: Disabled 1: Enabled

Break configuration register (TIMERx_BRKCFG)

Address offset: 0x78

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRK3P	BRK3EN	BRK2P	BRK2EN	BRK1P	BRK1EN	BRK0P	BRK0EN	Reserved							
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRK3F[3:0]				BRK2F[3:0]				BRK1F[3:0]				BRK0F[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31	BRK3P	BRKIN3 input signal polarity This bit specifies the polarity of the BRKIN3 input signal. 0: BRKIN3 input active low 1: BRKIN3 input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
30	BRK3EN	BRKIN3 input signal enable This bit can be set to enable the BRKIN3 input. 0: Break inputs disabled 1: Break inputs enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
29	BRK2P	BRKIN2 input signal polarity This bit specifies the polarity of the BRKIN2 input signal. 0: BRKIN2 input active low 1: BRKIN2 input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
28	BRK2EN	BRKIN2 input signal enable This bit can be set to enable the BRKIN2 input. 0: Break inputs disabled 1: Break inputs enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
27	BRK1P	BRKIN1 input signal polarity This bit specifies the polarity of the BRKIN1 input signal. 0: BRKIN1 input active low 1: BRKIN1 input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
26	BRK1EN	BRKIN1 input signal enable

This bit can be set to enable the BRKIN1 input.
0: Break inputs disabled
1: Break inputs enabled
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

25 BRK0P
BRKIN0 input signal polarity
This bit specifies the polarity of the BRKIN0 input signal.
0: BRKIN0 input active low
1: BRKIN0 input active high
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

24 BRK0EN
BRKIN0 input signal enable
This bit can be set to enable the BRKIN0 input.
0: Break inputs disabled
1: Break inputs enabled
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

23:16 Reserved
Must be kept at reset value.

15:12 BRK3F[3:0]
BRKIN3 input signal filter
An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BRKIN3 input signal and the length of the digital filter applied to BRKIN3.
0000: Filter disabled. BRKIN3 act asynchronously, N=1
0001: $f_{SAMP} = f_{CK_TIMER}$, N=2
0010: $f_{SAMP} = f_{CK_TIMER}$, N=4
0011: $f_{SAMP} = f_{CK_TIMER}$, N=8
0100: $f_{SAMP} = f_{DTS}/2$, N=6
0101: $f_{SAMP} = f_{DTS}/2$, N=8
0110: $f_{SAMP} = f_{DTS}/4$, N=6
0111: $f_{SAMP} = f_{DTS}/4$, N=8
1000: $f_{SAMP} = f_{DTS}/8$, N=6
1001: $f_{SAMP} = f_{DTS}/8$, N=8
1010: $f_{SAMP} = f_{DTS}/16$, N=5
1011: $f_{SAMP} = f_{DTS}/16$, N=6
1100: $f_{SAMP} = f_{DTS}/16$, N=8
1101: $f_{SAMP} = f_{DTS}/32$, N=5
1110: $f_{SAMP} = f_{DTS}/32$, N=6
1111: $f_{SAMP} = f_{DTS}/32$, N=8
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

11:8 BRK2F[3:0]
BRKIN2 input signal filter
An event counter is used in the digital filter, in which a transition on the output occurs

after N input events. This bit-field specifies the frequency used to sample BRKIN2 input signal and the length of the digital filter applied to BRKIN2.
0000: Filter disabled. BRKIN2 act asynchronously, N=1
0001: $f_{SAMP} = f_{CK_TIMER}$, N=2
0010: $f_{SAMP} = f_{CK_TIMER}$, N=4
0011: $f_{SAMP} = f_{CK_TIMER}$, N=8
0100: $f_{SAMP} = f_{DTS}/2$, N=6
0101: $f_{SAMP} = f_{DTS}/2$, N=8
0110: $f_{SAMP} = f_{DTS}/4$, N=6
0111: $f_{SAMP} = f_{DTS}/4$, N=8
1000: $f_{SAMP} = f_{DTS}/8$, N=6
1001: $f_{SAMP} = f_{DTS}/8$, N=8
1010: $f_{SAMP} = f_{DTS}/16$, N=5
1011: $f_{SAMP} = f_{DTS}/16$, N=6
1100: $f_{SAMP} = f_{DTS}/16$, N=8
1101: $f_{SAMP} = f_{DTS}/32$, N=5
1110: $f_{SAMP} = f_{DTS}/32$, N=6
1111: $f_{SAMP} = f_{DTS}/32$, N=8
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

7:4	BRK1F[3:0]	BRKIN1 input signal filter An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BRKIN1 input signal and the length of the digital filter applied to BRKIN1. 0000: Filter disabled. BRKIN1 act asynchronously, N=1 0001: $f_{SAMP} = f_{CK_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK_TIMER}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8 This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
3:0	BRK0F[3:0]	BRKIN0 input signal filter

An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BRKIN0 input signal and the length of the digital filter applied to BRKIN0.

0000: Filter disabled. BRKIN0 act asynchronously, N=1

0001: $f_{SAMP} = f_{CK_TIMER}$, N=2

0010: $f_{SAMP} = f_{CK_TIMER}$, N=4

0011: $f_{SAMP} = f_{CK_TIMER}$, N=8

0100: $f_{SAMP} = f_{DTS}/2$, N=6

0101: $f_{SAMP} = f_{DTS}/2$, N=8

0110: $f_{SAMP} = f_{DTS}/4$, N=6

0111: $f_{SAMP} = f_{DTS}/4$, N=8

1000: $f_{SAMP} = f_{DTS}/8$, N=6

1001: $f_{SAMP} = f_{DTS}/8$, N=8

1010: $f_{SAMP} = f_{DTS}/16$, N=5

1011: $f_{SAMP} = f_{DTS}/16$, N=6

1100: $f_{SAMP} = f_{DTS}/16$, N=8

1101: $f_{SAMP} = f_{DTS}/32$, N=5

1110: $f_{SAMP} = f_{DTS}/32$, N=6

1111: $f_{SAMP} = f_{DTS}/32$, N=8

This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

Free complementary channel protection register 0 (TIMERx_FCCHP0)

Address offset: 0x7C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This register is used to configure the outputs of CH0_O/MCH0_O.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP0EN	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ROS	IOS	Reserved		DTCFG[7:0]							
				rw	rw			rw							

Bits	Fields	Descriptions
31	FCCHP0EN	Free complementary channel protection register 0 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP0 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH0_O/MCH0_O) are disabled. 1: When POEN bit is set, the channel output signals (CH0_O/MCH0_O) are enabled, with relationship to CH0EN/MCH0EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH0_O/MCH0_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH0_O/MCH0_O) are enabled, with relationship to CH0EN/ MCH0EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

Free complementary channel protection register 1 (TIMERx_FCCHP1)

Address offset: 0x80
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).
This register is used to configure the outputs of CH1_O/MCH1_O.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCCHP1EN	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	ROS	IOS	Reserved	DTCFG[7:0]
	rw	rw		rw

Bits	Fields	Descriptions
31	FCCHP1EN	Free complementary channel protection register 1 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP1 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH1_O/MCH1_O) are disabled. 1: When POEN bit is set, the channel output signals (CH1_O/MCH1_O) are enabled, with relationship to CH1EN/ MCH1EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH1_O/MCH1_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH1_O/MCH1_O) are enabled, with relationship to CH1EN/ MCH1EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

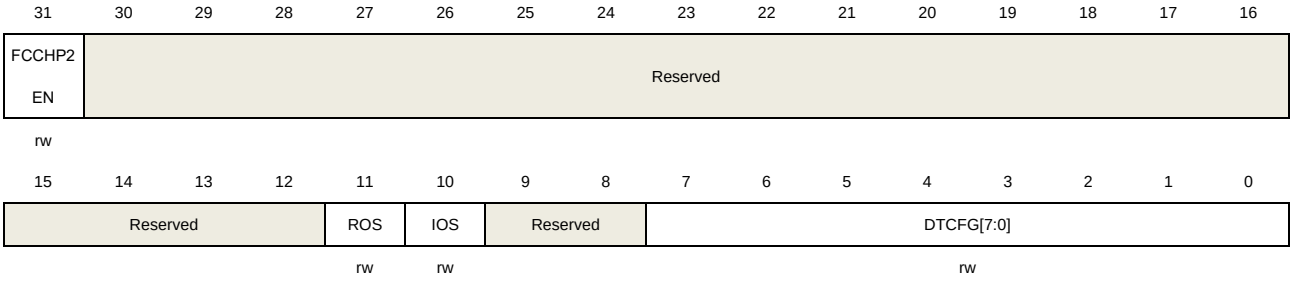
Free complementary channel protection register 2 (TIMERx_FCCHP2)

Address offset: 0x84

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This register is used to configure the outputs of CH2_O/MCH2_O.



Bits	Fields	Descriptions
31	FCCHP2EN	Free complementary channel protection register 2 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP2 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH2_O/MCH2_O) are disabled. 1: When POEN bit is set, the channel output signals (CH2_O/MCH2_O) are enabled, with relationship to CH2EN/ MCH2EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH2_O/MCH2_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH2_O/MCH2_O) are enabled, with relationship to CH2EN/ MCH2EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure

This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow:

DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}.

DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2.

DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8.

DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16.

This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

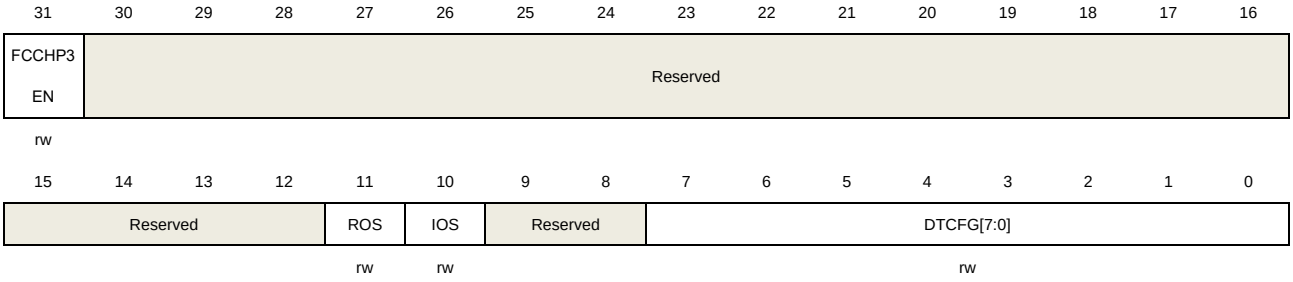
Free complementary channel protection register 3 (TIMERx_FCCHP3)

Address offset: 0x88

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This register is used to configure the outputs of CH3_O/MCH3_O.



Bits	Fields	Descriptions
31	FCCHP3EN	Free complementary channel protection register 0 enable 0: the ROS、IOS and DTCFG[7:0] bits in TIMERx_CCHP register is active 1: the ROS、IOS and DTCFG[7:0] bits in TIMERx_FCCHP3 register is active This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.
30:12	Reserved	Must be kept at reset value.
11	ROS	Run mode off-state configure When POEN bit is set, this bit specifies the output state for the channels which has a complementary output and has been configured in output mode. 0: When POEN bit is set, the channel output signals (CH3_O/MCH3_O) are disabled. 1: When POEN bit is set, the channel output signals (CH3_O/MCH3_O) are enabled, with relationship to CH3EN/ MCH3EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode off-state configure When POEN bit is reset, this bit specifies the output state for the channels which

		has been configured in output mode. 0: When POEN bit is reset, the channel output signals (CH3_O/MCH3_O) are disabled. 1: When POEN bit is reset, the channel output signals (CH3_O/MCH3_O) are enabled, with relationship to CH3EN/ MCH3EN bits in TIMERx_CHCTL2 register. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	Reserved	Must be kept at reset value.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DMATC[5:0]						Reserved		DMATA[5:0]					
		rw								rw					

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This filed defines the number(n) of the register that DMA will access(R/W), n = (DMATC [5:0] +1). DMATC [5:0] is from 6'b000000 to 6'b100010.
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to

the TIMERx_DMATB register will be (start address + 0x4).
In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CHVSEL	OUTSEL
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored.

		0: No effect.
0	OUTSEL	The output value selection bit This bit-field is set and reset by software. 1: If POEN bit and IOS bit are 0, the output is disabled. 0: No effect.

18.2. General level0 timer (TIMERx, x=1)

18.2.1. Overview

The general level0 timer module (TIMER1) is a four-channel timer that supports input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level0 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level0 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison.

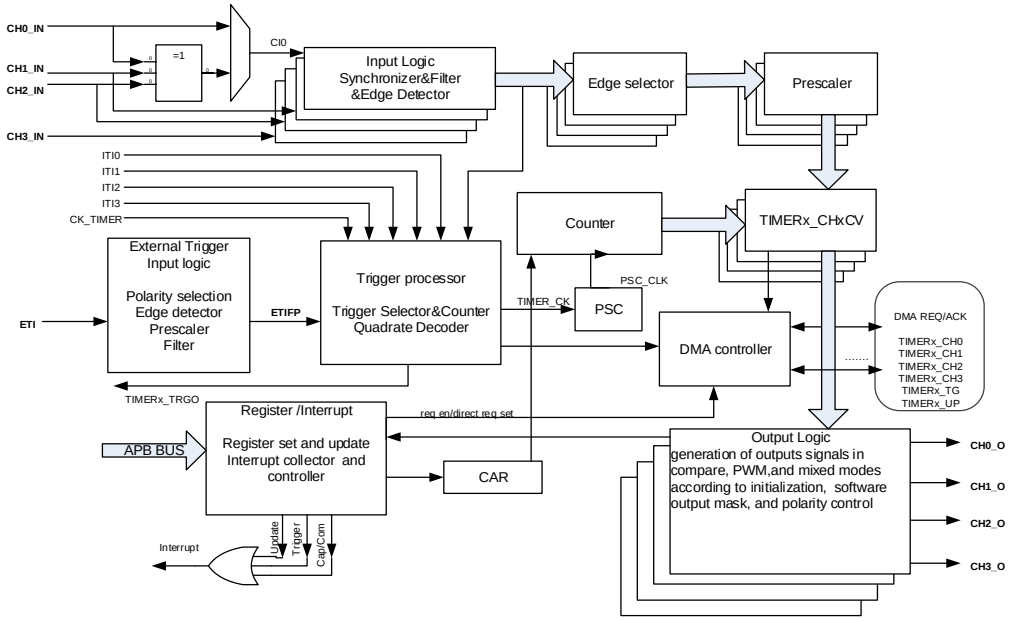
18.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode and single pulse mode.
- Auto reload function.
- Interrupt output or DMA request: update event, trigger event and compare/capture event.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

18.2.3. Block diagram

Figure 18-42. General Level 0 timer block diagram provides details on the internal configuration of the general level 0 timer.

Figure 18-42. General Level 0 timer block diagram



18.2.4. Function overview

Clock source configuration

The general level0 TIMER has the capability of being clocked by either the CK_TIMER or an alternate clock source controlled by SMC (TIMERx_SMCFG bit[2:0]).

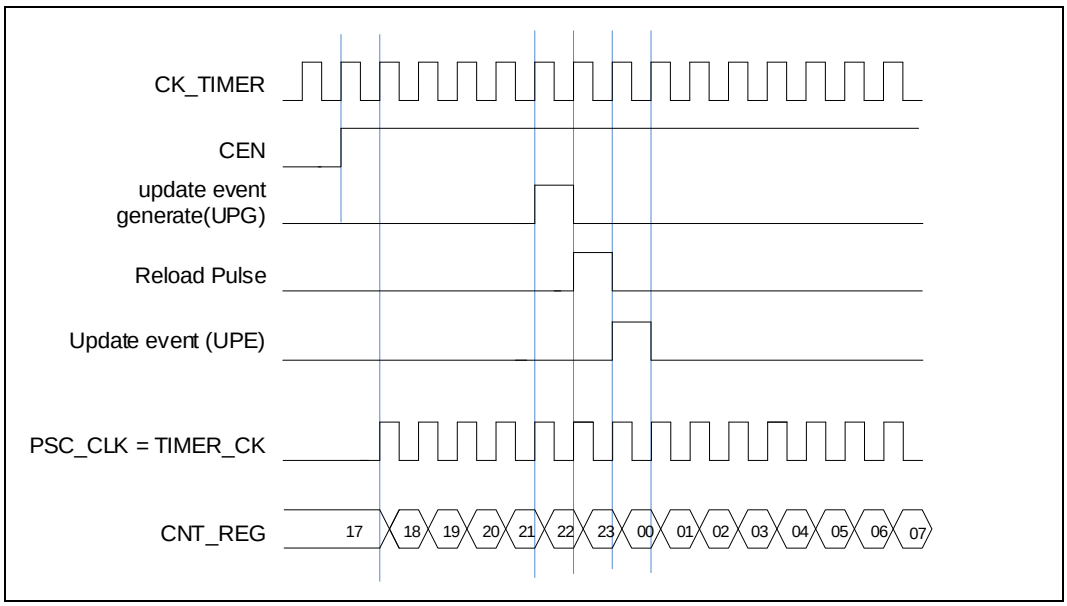
- SMC[2:0] = 3'b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when the SMC[2:0] = 3'b000. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If the SMC[2:0] in the TIMERx_SMCFG register are setting to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock sources selected by the TRGS[2:0] in the TIMERx_SMCFG register, more details will be introduced later. When the SMC[2:0] bits are set to 0x4, 0x5 or 0x6, the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 18-43. Timing chart of internal clock divided by 1



- SMC[2:0] = 3'b111 (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting SMC[2:0] to 0x7 and the TRGS[2:0] to 0x4, 0x5 or 0x6.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/1/2/3. This mode can be selected by setting SMC[2:0] to 0x7 and the TRGS[2:0] to 0x0, 0x1, 0x2 or 0x3.

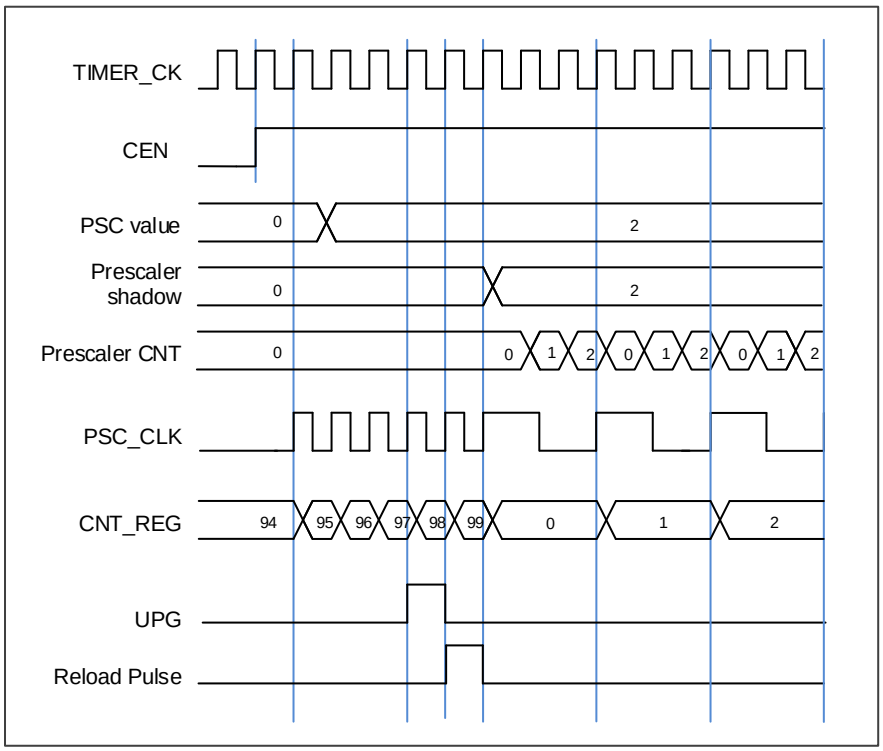
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is setting the SMC[2:0] to 0x7 and the TRGS[2:0] to 0x7. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescaler factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 18-44. Timing chart of PSC value change from 0 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again. The update event is generated at each counter overflow. The counting direction bit DIR in the TIMERx_CTL1 register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter auto reload register, prescaler register) are updated.

Figure 18-45. Timing chart of up counting mode, PSC=0/2 and ***Figure 18-46. Timing chart of up counting, change TIMERx_CAR on the go*** show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 18-45. Timing chart of up counting mode, PSC=0/2

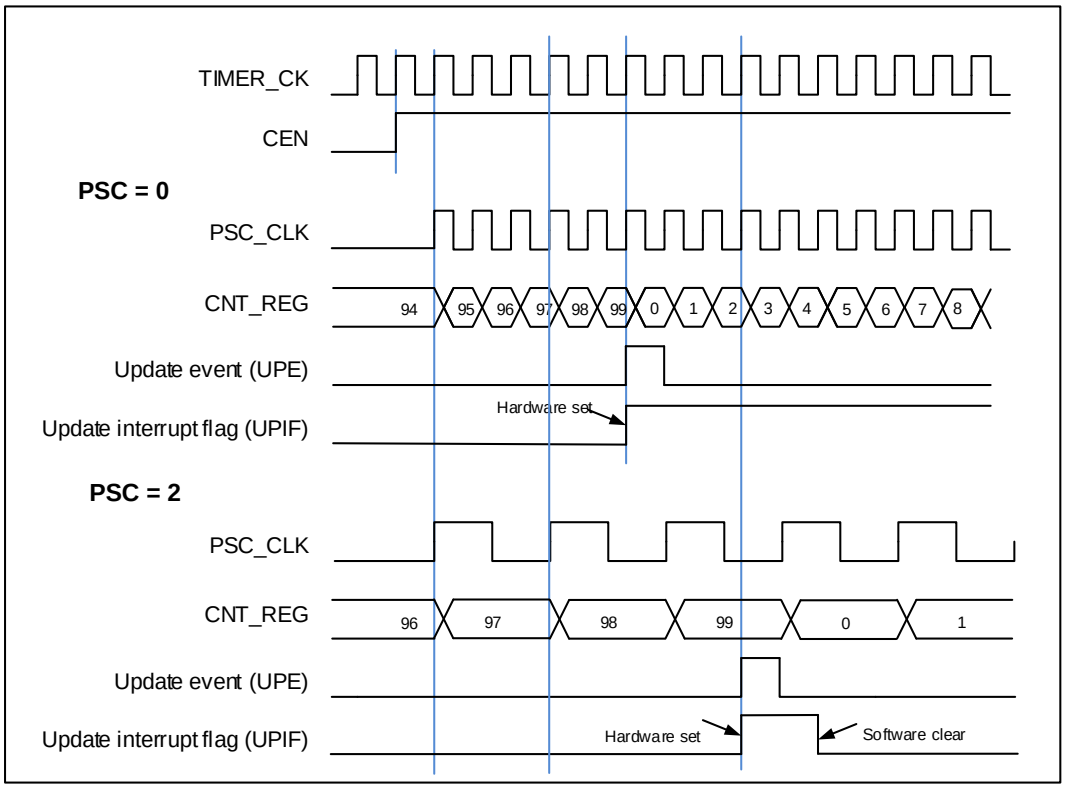
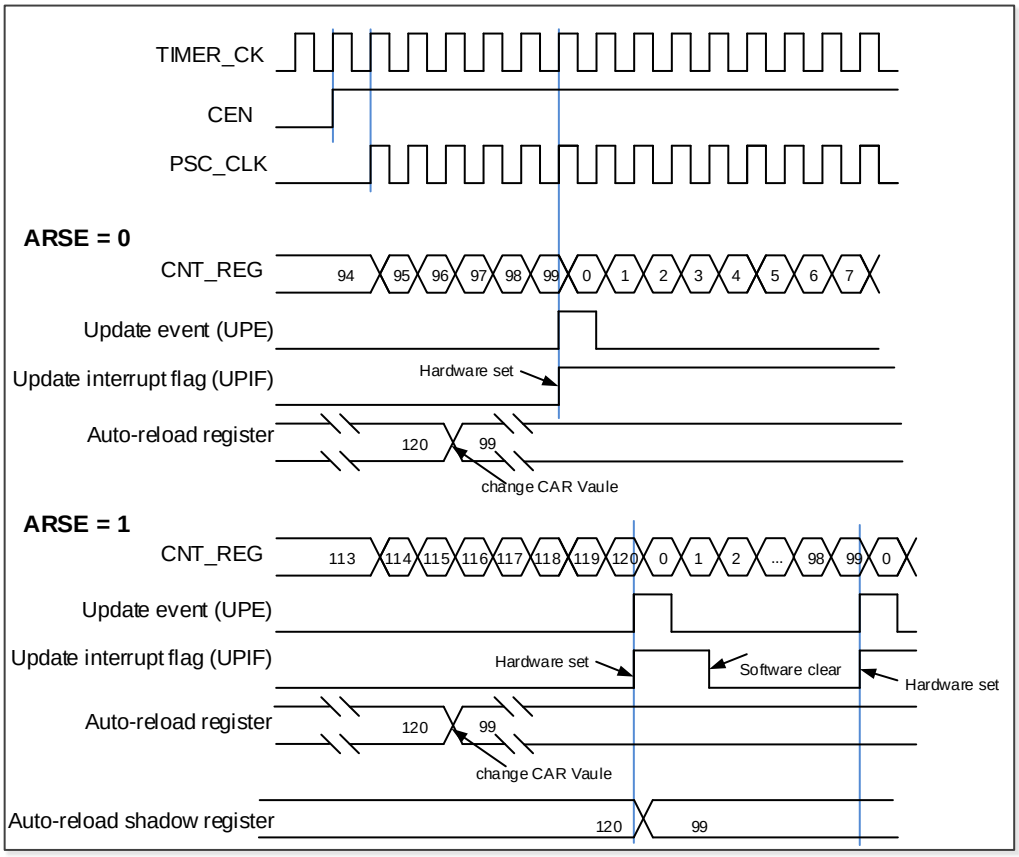


Figure 18-46. Timing chart of up counting, change TIMERx_CAR on the go



Counter down counting

In this mode, the counter counts down continuously from the counter-reload value, which is defined in the TIMERx_CAR register, to 0 in a count-down direction. Once the counter reaches to 0, the counter will start counting down from the counter-reload value. The update event is generated at each counter underflow. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down-counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter autoreload register, prescaler register) are updated.

Figure 18-47. Timing chart of down counting mode, PSC=0/2 and *Figure 18-48. Timing chart of down counting mode, change TIMERx_CAR on the go* show some examples of the counter behavior for different clock frequencies when TIMERx_CAR = 0x99.

Figure 18-47. Timing chart of down counting mode, PSC=0/2

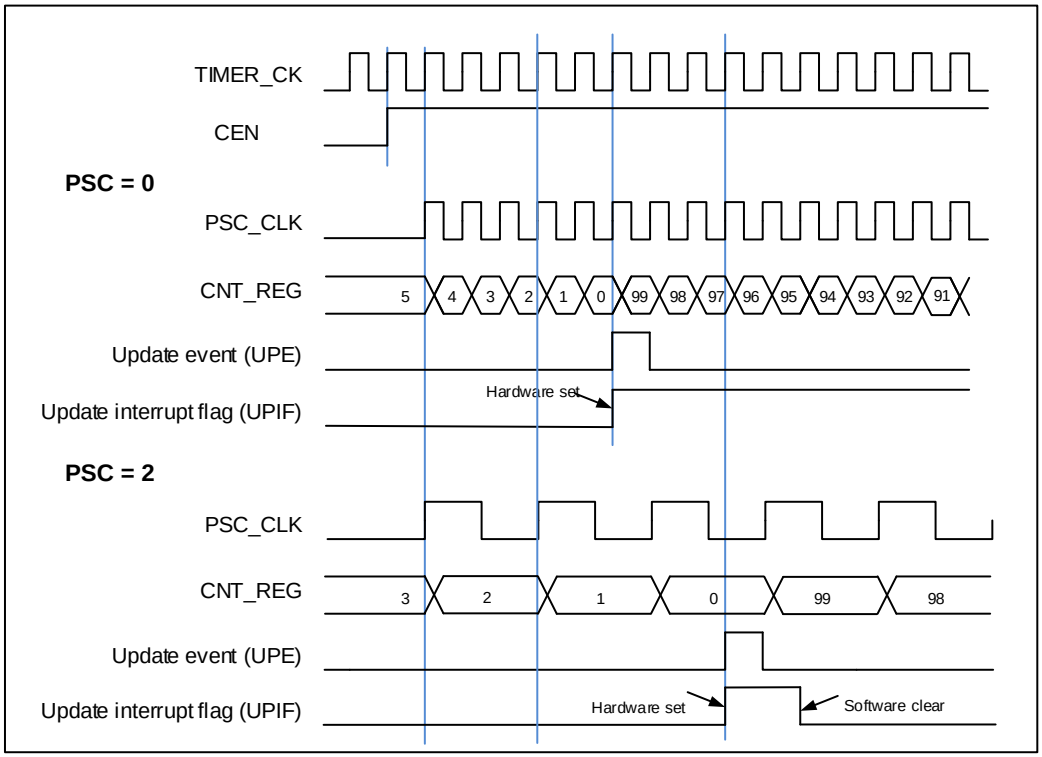
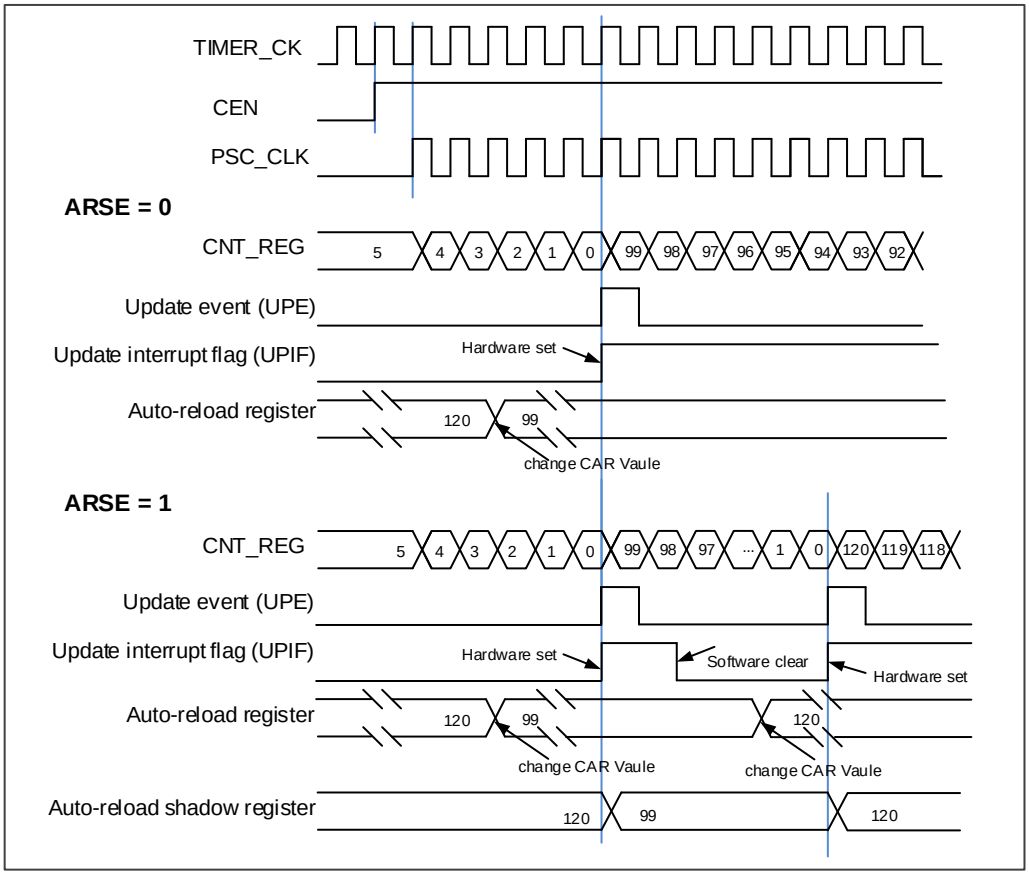


Figure 18-48. Timing chart of down counting mode, change TIMERx_CAR on the go



Counter center-aligned counting

In the center-aligned counting mode, the counter counts up from 0 to the counter-reload value and then counts down to 0 alternatively. The TIMER module generates an overflow event when the counter counts to the counter-reload value subtract 1 in the up-counting mode and generates an underflow event when the counter counts to 1 in the down-counting mode. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned mode.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

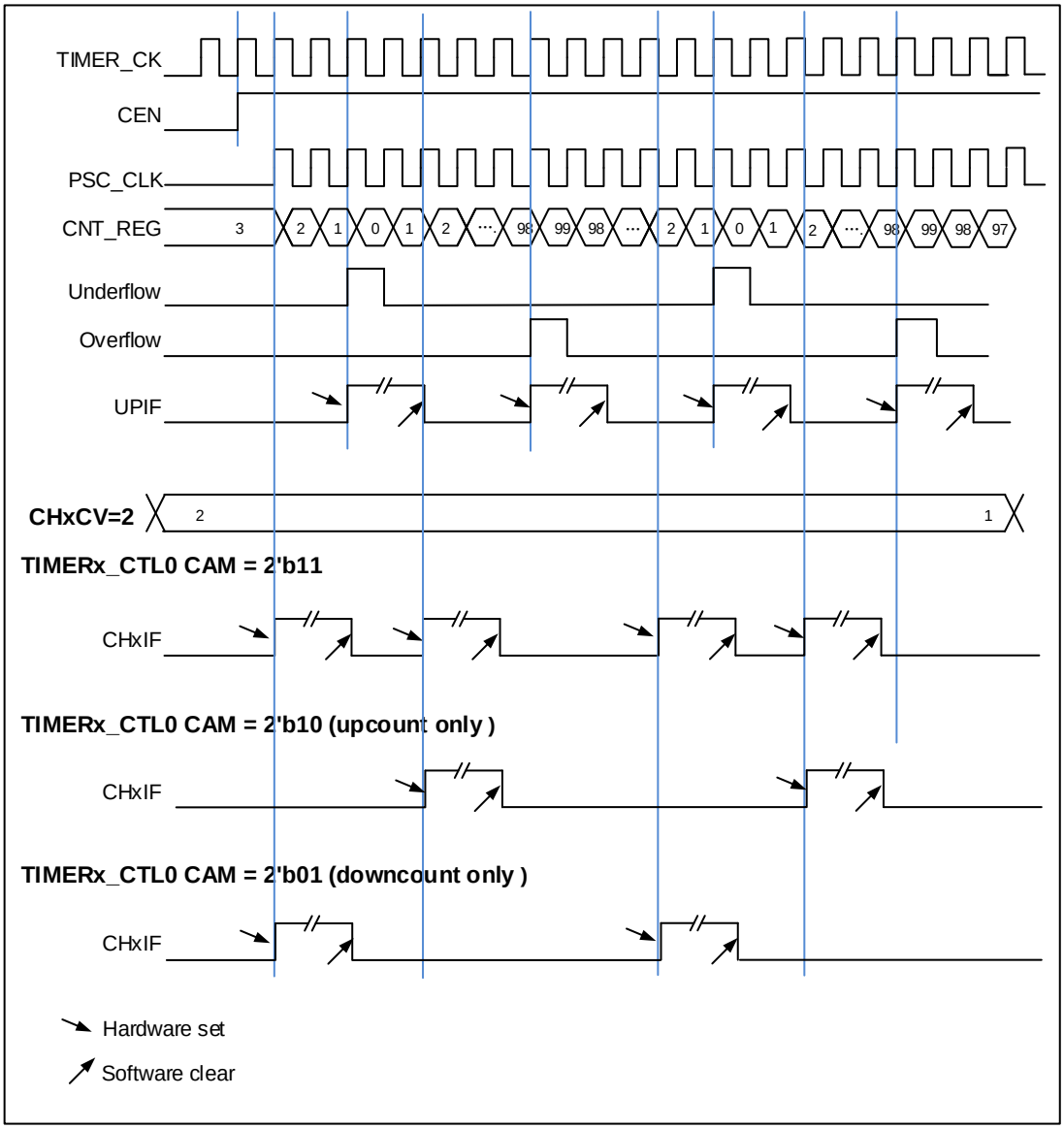
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 18-49. Timing chart of center-aligned counting mode***.

If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter autoreload register, prescaler register) are updated.

Figure 18-49. Timing chart of center-aligned counting mode shows the example of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0.

Figure 18-49. Timing chart of center-aligned counting mode



Input capture and output compare channels

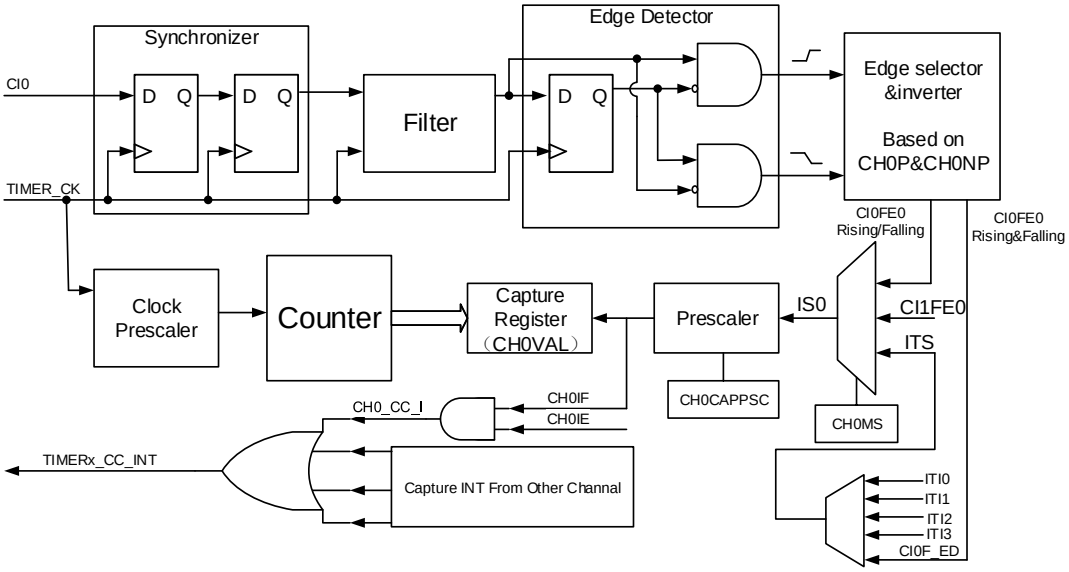
The general level0 Timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

- **Channel input capture function**

Channel input capture function allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the CHxIF bit is set and the channel interrupt is

generated if it is enabled when CHxIE=1.

Figure 18-50. Channels input capture principle



The input signals of channelx (CIx) can be the TIMERx_CHx signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals (just for CI0). First, the input signal of channel (CIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP bit. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT in TIMERx_CHCTL0).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT.

Step2: Edge selection (CHxP/CHxNP in TIMERx_CHCTL2).

Rising edge or falling edge, choose one by configuring CHxP/CHxNP bits.

Step3: Capture source selection (CHxMS in TIMERx_CHCTL0)

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN in TIMERx_DMAINTEN)

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN in TIMERx_CHCTL2)

Result: When the wanted input signal is captured, TIMERx_CHxCV will be set by counter's value and CHxIF is asserted. If the CHxIF is 1, the CHxOF will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN in TIMERx_DMAINTEN.

Direct generation: A DMA or interrupt request is generated by setting CHxG directly.

The channel input capture function can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

- **Channel output compare function**

Figure 18-51. Channel output compare principle (x=0,1,2,3)

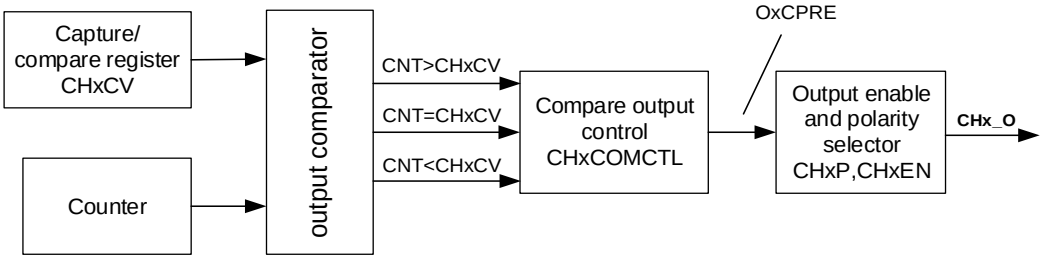


Figure 18-51. Channel output compare principle (x=0,1,2,3) shows the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output prepare signal*) is described as blew: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE signal, CHxP bit and CH0P bit (please refer to the TIMERx_CHCTL2 register for more details).For example, configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxEN=1 (the output of CHx_O is enabled),

If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

In channel output compare function, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV register, the CHxIF bit will be set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be asserted, if CHxDEN=1.

So, the process can be divided into several steps as below:

Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP.
- Enable the output by CHxEN.

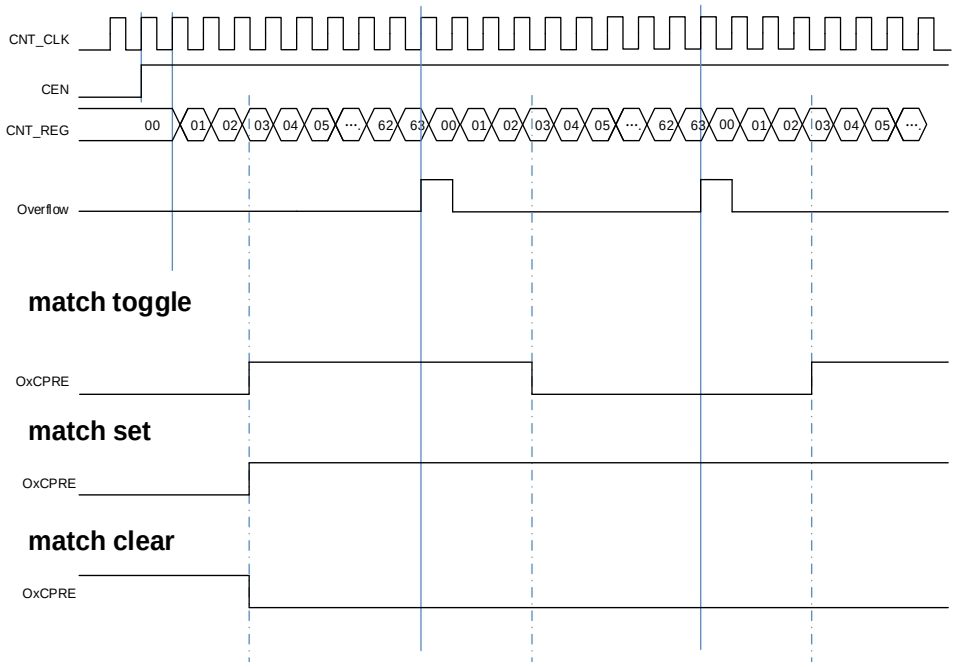
Step3: Interrupt/DMA-request enables configuration by CHxIE/CHxDEN.

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV. The TIMERx_CHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 18-52. Output-compare under three modes shows the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 18-52. Output-compare under three modes



Output PWM function

In the PWM output mode (by setting the CHxCOMCTL bit to 3'b110 (PWM mode 0) or to 3'b 111(PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV. *Figure 18-53. EAPWM timechart* shows the EAPWM output and interrupts waveform.

The CAPWM period is determined by 2*TIMERx_CAR, and duty cycle is determined by 2*TIMERx_CHxCV. *Figure 18-54. CAPWM timechart* shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV is greater than the value of

TIMERx_CAR, the output will be always active in PWM mode 0 (CHxCOMCTL=3'b110). And if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 1 (CHxCOMCTL=3'b111).

Figure 18-53. EAPWM timechart

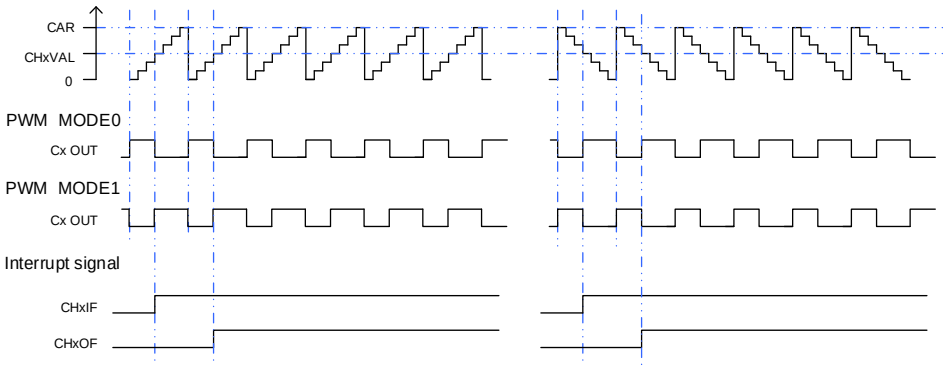
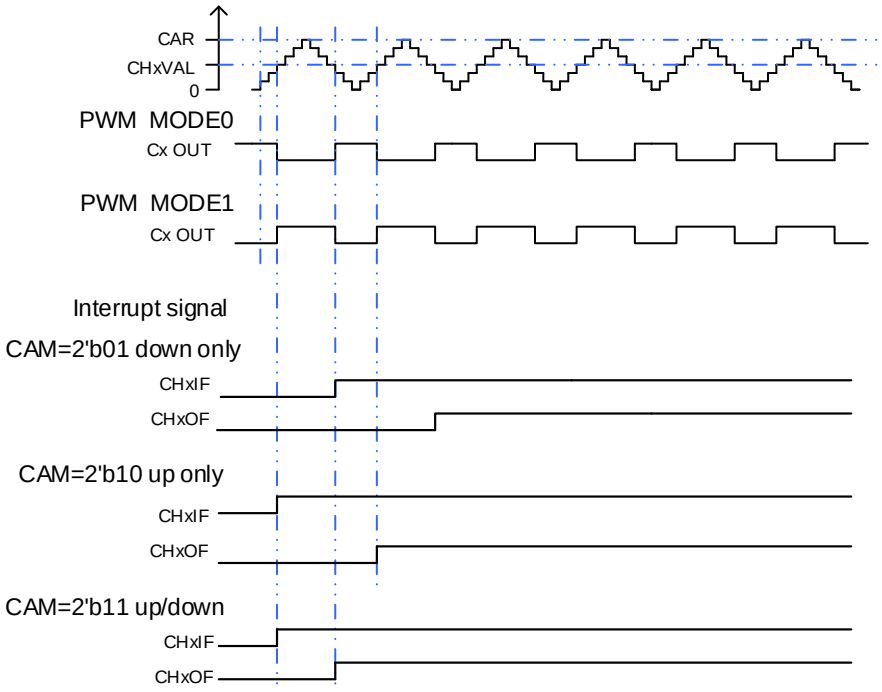


Figure 18-54. CAPWM timechart



Channel output prepare signal

As is shown in ***Figure 18-51. Channel output compare principle (x=0,1,2,3)***, when TIMERx is configured in compare match output mode,a middle signal which is OxCPRE signal (Channel x output prepare signal) will be generated before the channel outputs signal. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit. The OxCPRE signal has several types of output function. These include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring

the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

Another special function of the OxCPRE signal is a forced output which can be achieved by configuring the CHxCOMCTL field to 0x04/0x05. The output can be forced to an inactive/active level irrespective of the comparison condition between the values of the counter and the TIMERx_CHxCV.

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next update event occurs.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0FE0 and CI1FE1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact with each other to generate the counter value. Setting SMC=0x01, 0x02, or 0x03 to select that the counting direction of timer is determined only by the CI0FE0, only by the CI1FE1, or by the CI0FE0 and the CI1FE1. The DIR bit is modified during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in ***Table 18-7. Counting direction in different quadrature decoder mode***. The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 18-7. Counting direction in different quadrature decoder mode

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 SMC[2:0]=3'b001	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 SMC [2:0]=3'b010	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 SMC [2:0]=3'b011	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 18-55. Counter behavior with CI0FE0 polarity non-inverted in mode 2

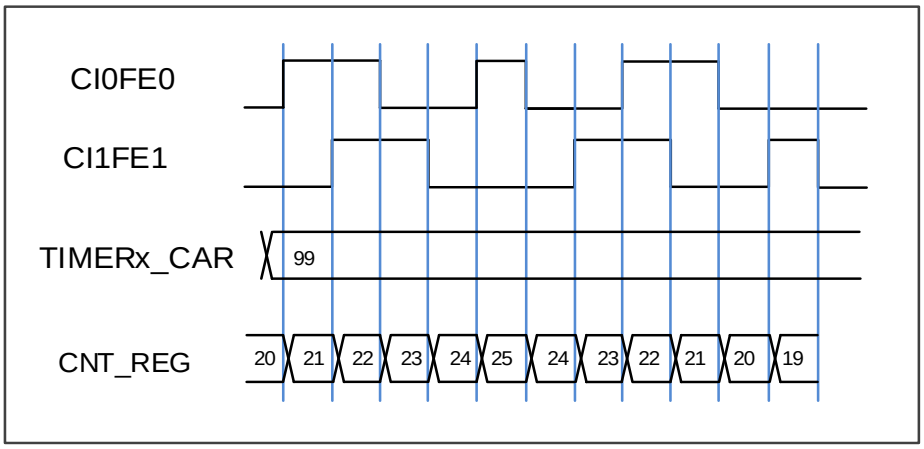
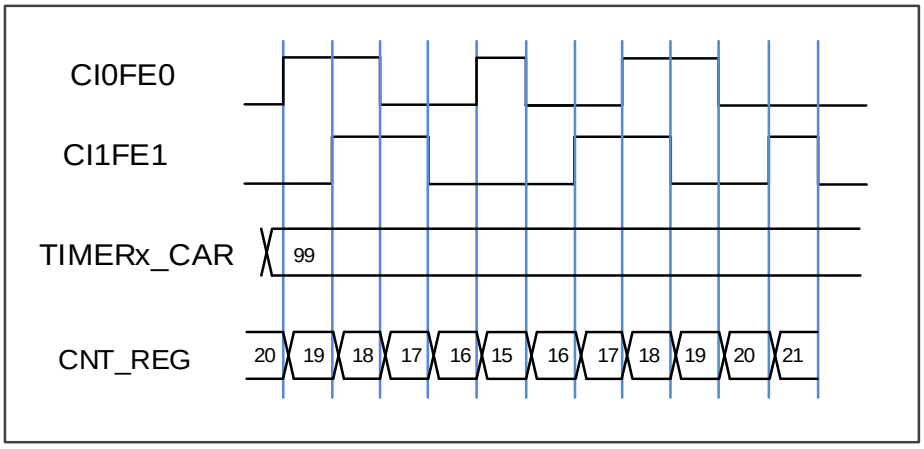


Figure 18-56. Counter behavior with CI0FE0 polarity inverted in mode 2



Hall sensor function

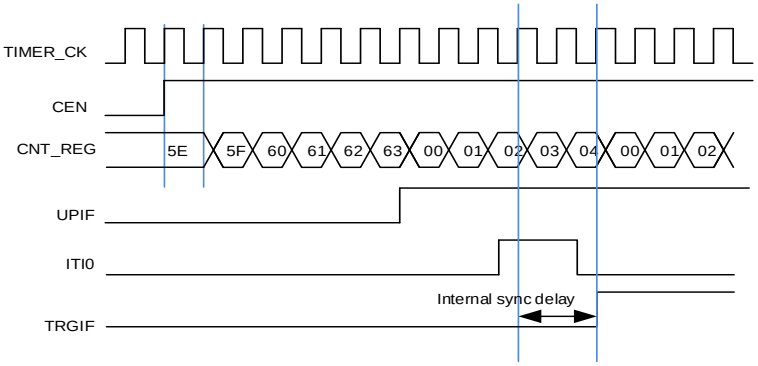
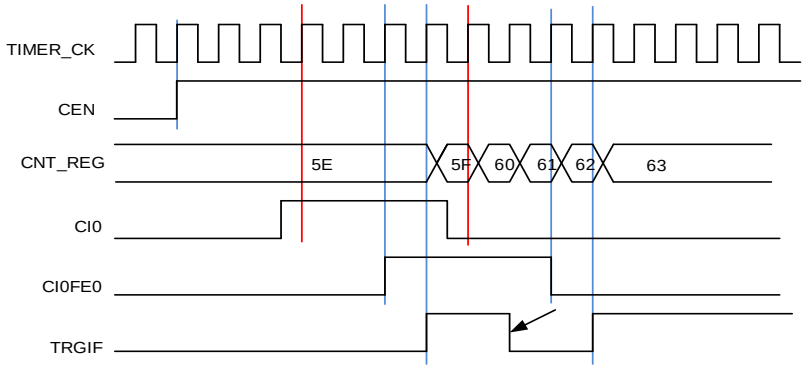
Refer to ***Advanced timer (TIMERx, x=0, 7, 19, 20)Hall sensor function***.

Master-slave management

The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode which is selected by the SMC[2:0] bits in the TIMERx_SMCFG register. The input trigger of these modes can be selected by the TRGS[2:0] bits in the TIMERx_SMCFG register.

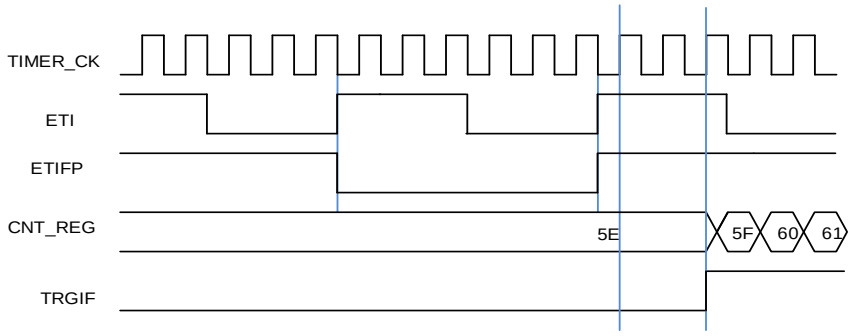
Table 18-8. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	SMC[2:0] 3'b100 (restart mode) 3'b101 (pause mode) 3'b110 (event mode)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion. If ETIFP (the filtered	For the ITIx, no filter and prescaler can be used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used.

<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td></td><td></td><td>110: CI1FE1
111: ETIFP(1)</td><td>output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.</td><td>For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.</td></tr>
<tr><td rowspan="2">Exam1</td><td>Restart mode
The counter will be cleared and restart when a rising edge of trigger input comes.</td><td>TRGS[2:0] =3’b000
ITI0 is selected.</td><td>For ITI0, no polarity selector can be used.</td><td>For the ITI0, no filter and prescaler can be used.</td></tr>
<tr><td colspan="4">Figure 18-57. Restart mode

</td></tr>
<tr><td rowspan="2">Exam2</td><td>Pause mode
The counter will be paused when the trigger input is low, and it will start when the trigger input is high.</td><td>TRGS[2:0] =3’b101
CI0FE0 is selected.</td><td>TI0S = 0 (Non-xor) [CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.</td><td>Filter is bypassed in this example.</td></tr>
<tr><td colspan="4">Figure 18-58. Pause mode

</td></tr>
<tr><td>Exam3</td><td>Event mode
The counter will start to count when a rising</td><td>TRGS[2:0] = 3’b111
ETIFP is selected.</td><td>ETP = 0, the polarity of ETI does not change.</td><td>ETPSC = 1, ETI is divided by 2.
ETFC = 0, ETI does not</td></tr>
</table>

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	edge of trigger input comes.			filter.

Figure 18-59. Event mode



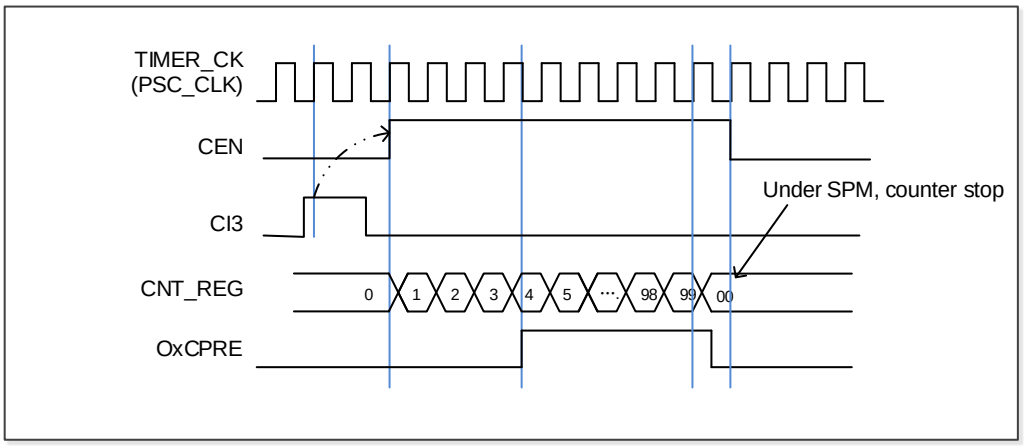
Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signal will change to, as the compare match event occurs without taking the comparison result into account.

Figure 18-60. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x99



Timers interconnection

Refer to ***Advanced timer (TIMERx, x=0, 7, 19, 20)Timers interconnection***.

Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

18.2.5. Registers definition (TIMERx, x=1)

TIMER1 base address: 0x4000 0000

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: f_{DTS}= f_{CK_TIMER} 01: f_{DTS}= f_{CK_TIMER} /2 10: f_{DTS}= f_{CK_TIMER} /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Only when counting down, CHxF bit can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Only when counting up, CHxF bit can be set. 11: Center-aligned and counting up/down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Both when counting up and counting down, CHxF bit can be set.

After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.

4	DIR	Direction 0: Count up 1: Count down If the timer work in center-aligned mode or decoder mode, this bit is read only.
3	SPM	Single pulse mode. 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate an update interrupt or a DMA request: – The UPG bit is set – The counter generates an overflow or underflow event – The restart mode generates an update event. 1: This event generates update interrupts or DMA requests: The counter generates an overflow or underflow event
1	UPDIS	Update disabled. This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event: – The UPG bit is set – The counter generates an overflow or underflow event – The restart mode generates an update event. 1: Update event disable. **Note:** When this bit is set to 1, setting UPG bit or restart mode does not generate an update event, but the counter and prescaler are initialized.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI0S	MMC[2:0]			DMAS	Reserved		
								rw	rw			rw			

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. Master mode control These bits control the selection of TRGO signal, which is sent in master mode to slave timers for synchronization function. 000: When a counter reset event occurs, a TRGO trigger signal is output. The counter reset source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a counter start event occurs, a TRGO trigger signal is output. The counter start source: CEN control bit is set The trigger input in pause mode is high 010: When an update event occurs, a TRGO trigger signal is output. The update source depends on UPDIS bit and UPS bit. 011: When a capture or compare pulse event occurs in channel0, a TRGO trigger signal is output. 100: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O0CPRE. 101: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O1CPRE. 110: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O2CPRE. 111: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O3CPRE.
3	DMAS	DMA request source selection 0: When capture or compare event occurs, the DMA request of channel x is sent 1: When update event occurs, the DMA request of channel x is sent.
2:0	Reserved	Must be kept at reset value.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			Reserved	SMC[2:0]		
rw	rw	rw		rw				rw	rw				rw		

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of SMC is used to enable External clock mode1. In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled. When the slave mode is configured as restart mode, pause mode or event mode, the timer can still work in the external clock 1 mode by setting this bit. But the TRGS bits must not be 3'b111 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in this register's SMC bit-field.
13:12	ETPSC[1:0]	The prescaler of external trigger The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: The prescaler is 2. 10: The prescaler is 4. 11: The prescaler is 8.
11:8	ETFC[3:0]	External trigger filter control The external trigger can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the external trigger signal according to f_{SAMP} and record the number of times of the same level of the signal.

After reaching the filtering capacity configured by this bit-field, it is considered to be an effective level.

The filtering capability configuration is as follows:

EXTFC[3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS_CK}/2$
4'b0101	8	
4'b0110	6	$f_{DTS_CK}/4$
4'b0111	8	
4'b1000	6	$f_{DTS_CK}/8$
4'b1001	8	
4'b1010	5	$f_{DTS_CK}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS_CK}/32$
4'b1110	6	
4'b1111	8	

7 MSM

Master-slave mode

This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected together.

0: Master-slave mode disabled

1: Master-slave mode enabled

6:4 TRGS[2:0]

Trigger selection

This bit-field specifies which signal is selected as the trigger input to synchronize the timers.

000: Internal trigger input 0 (ITI0)

001: Internal trigger input 1 (ITI1)

010: Internal trigger input 2 (ITI2)

011: Internal trigger input 3 (ITI3)

100: CI0 edge flag (CI0F_ED)

101: The filtered output of channel 0 input (CI0FE0)

110: The filtered output of channel 1 input (CI1FE1)

111: The filtered output of external trigger input (ETIFP)

These bits must not be changed when slave mode is enabled.

3 Reserved

Must be kept at reset value.

2:0 SMC[2:0]

Slave mode control

000: Disable slave mode. The slave mode is disabled; The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high.

001: Quadrature decoder mode 0. The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.
010: Quadrature decoder mode 1. The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level.
011: Quadrature decoder mode 2. The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on the level of the other (CI1FE1 or CI0FE0).
100: Restart mode. The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input.
101: Pause mode. The trigger input enables the counter clock when it is high and disables the counter clock when it is low.
110: Event mode. A rising edge of the trigger input enables the counter.
111: External clock mode0. The counter counts on the rising edges of the selected trigger.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	Reserved	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	Reserved	TRGIE	Reserved	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	Reserved	Must be kept at reset value.
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled

		1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	Reserved	Must be kept at reset value.
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	Reserved	Must be kept at reset value.
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CH3OF	CH2OF	CH1OF	CH0OF	Reserved		TRGIF	Reserved	CH3IF	CH2IF	CH1IF	CH0IF	UPIF

rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8:7	Reserved	Must be kept at reset value.
6	TRGIF	Trigger interrupt flag This flag is set on trigger event and cleared by software. When in pause mode, both edges on trigger input generates a trigger event, otherwise, only an active edge on trigger input can generates a trigger event. 0: No trigger event occurred. 1: Trigger interrupt occurred.
5	Reserved	Must be kept at reset value.
4	CH3IF	Channel 3 capture/compare interrupt enable Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt enable Refer to CH0IF description
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag

This bit is set by hardware when an update event occurs and cleared by software.
0: No update interrupt occurred
1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TRGG	Reserved	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

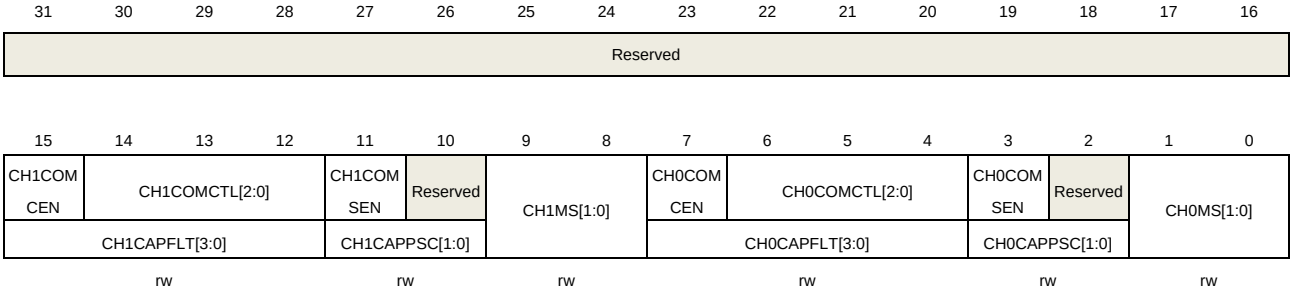
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_STAT register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	Reserved	Must be kept at reset value.
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event

0	UPG	This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Output compare mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	Reserved	Must be kept at reset value.
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (CH1EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 1 is programmed as output. 01: Channel 1 is programmed as input, IS1 is connected to CI1FE1. 10: Channel 1 is programmed as input, IS1 is connected to CI0FE1. 11: Channel 1 is programmed as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).
7	CH0COMCEN	Channel 0 output compare clear enable.

		When this bit is set, if the ETIFP signal is detected as high level, the O0CPRE signal will be cleared. 0: Channel 0 output compare clear disabled 1: Channel 0 output compare clear enabled
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field controls the behavior of O0CPRE which drives CH0_O. O0CPRE is active high, while CH0_O active level depends on CH0P bits. 000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 001: Set the channel output on match. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output on match. O0CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter is equals to the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced to low level. 101: Force high. O0CPRE is forced to high level. 110: PWM mode 0. When counting up, O0CPRE is active when the counter is smaller than TIMERx_CH0CV, otherwise it is inactive. When counting down, O0CPRE is inactive when the counter is larger than TIMERx_CH0CV, otherwise it is active. 111: PWM mode 1. When counting up, O0CPRE is inactive when the counter is smaller than TIMERx_CH0CV, otherwise it is active. When counting down, O0CPRE is active when the counter is larger than TIMERx_CH0CV, otherwise it is inactive. If configured in PWM mode, the O0CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes.
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1).
2	Reserved	Must be kept at reset value.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (CH0EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 0 is programmed as output. 01: Channel 0 is programmed as input, IS0 is connected to CI0FE0.

10: Channel 0 is programmed as input, IS0 is connected to CI1FE0.

11: Channel 0 is programmed as input, IS0 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description
9:8	CH1MS[1:0]	Channel 1 mode selection Same as output compare mode
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control The CI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:
3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared.

CH0CAPFLT [3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

		00: Prescaler disabled, input capture occurs on every channel input edge. 01: The input capture occurs on every 2 channel input edges 10: The input capture occurs on every 4 channel input edges 11: The input capture occurs on every 8 channel input edges
1:0	CH0MS[1:0]	Channel 0 mode selection Same as output compare mode

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	Reserved	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	Reserved	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH0COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH0COMCTL description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH0COMSEN description
10	Reserved	Must be kept at reset value.
9:8	CH3MS[1:0]	Channel 3 mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active (CH3EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 3 is programmed as output. 01: Channel 3 is programmed as input, IS3 is connected to CI3FE3. 10: Channel 3 is programmed as input, IS3 is connected to CI2FE3. 11: Channel 3 is programmed as input, IS3 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).

7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the O2CPRE signal will be cleared. 0: Channel 2 output compare clear disabled 1: Channel 2 output compare clear enabled
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field controls the behavior of O2CPRE which drives CH2_O. The active level of O2CPRE is high, while the active level of CH2_O depends on CH2P bit. 000: Timing mode. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 001: Set the channel output on match. O2CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH2CV. 010: Clear the channel output on match. O2CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH2CV. 011: Toggle on match. O2CPRE toggles when the counter is equals to the output compare register TIMERx_CH2CV. 100: Force low. O2CPRE is forced to low level. 101: Force high. O2CPRE is forced to high level. 110: PWM mode 0. When counting up, O2CPRE is active when the counter is smaller than TIMERx_CH2CV, otherwise it is inactive. When counting down, O2CPRE is inactive when the counter is larger than TIMERx_CH2CV, otherwise it is active. 111: PWM mode 1. When counting up, O2CPRE is inactive when the counter is smaller than TIMERx_CH2CV, otherwise it is active. When counting down, O2CPRE is active when the counter is larger than TIMERx_CH2CV, otherwise it is inactive. If configured in PWM mode, the O2CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes.
3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disabled 1: Channel 2 output compare shadow enabled The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1).
2	Reserved	Must be kept at reset value.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (CH2EN bit in TIMERx_CHCTL2 register is reset).

00: Channel 2 is programmed as output.

01: Channel 2 is programmed as input, IS2 is connected to CI2FE2.

10: Channel 2 is programmed as input, IS2 is connected to CI3FE2.

11: Channel 2 is programmed as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TRGS bits in TIMERx_SMCFG register).

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH0CAPFLT description
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH0CAPPSC description
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control The CI2 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI2 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

CH2CAPFLT [3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler

		is reset when CH2EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, input capture occurs on every channel input edge. 01: The input capture occurs on every 2 channel input edges 10: The input capture occurs on every 4 channel input edges 11: The input capture occurs on every 8 channel input edges
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	Reserved	CH3P	CH3EN	CH2NP	Reserved	CH2P	CH2EN	CH1NP	Reserved	CH1P	CH1EN	CH0NP	Reserved	CH0P	CH0EN
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH3NP	Channel 3 complementary capture/compare polarity Refer to CH0NP description.
14	Reserved	Must be kept at reset value.
13	CH3P	Channel 3 capture/compare function polarity Refer to CH0P description
12	CH3EN	Channel 3 capture/compare function enable Refer to CH0EN description
11	CH2NP	Channel 2 complementary capture/compare polarity Refer to CH0NP description.
10	Reserved	Must be kept at reset value.
9	CH2P	Channel 2 capture/compare function polarity Refer to CH0P description
8	CH2EN	Channel 2 capture/compare function enable Refer to CH0EN description
7	CH1NP	Channel 1 complementary capture/compare polarity Refer to CH0NP description.

6	Reserved	Must be kept at reset value.
5	CH1P	Channel 1 capture/compare function polarity Refer to CH0P description
4	CH1EN	Channel 1 capture/compare function enable Refer to CH0EN description
3	CH0NP	Channel 0 complementary output polarity When channel 0 complementary is configured in output mode, this bit must be kept at reset value. When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.
2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, this bit specifies the channel 0 input signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. 00: CIxFE0 rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. 01: CIxFE0 falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. 10: Reserved. 11: Noninverted/both CIxFE0's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter. **Note:** When the timer is configured in input capture mode, this register must be configured a non-zero value (such as 0xFFFF) which is larger than user expected value.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event.And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMATC[4:0]					Reserved			DMATA[4:0]				

rw rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC[4:0]	DMA transfer count This filed defines the number(n) of the register that DMA will access(R/W), n = (DMATC [4:0] +1). DMATC [4:0] is from 5'b00000 to 5'b10001.
7:5	Reserved	Must be kept at reset value.
4:0	DMATA[4:0]	DMA transfer access start address This field define the first address for the DMA access the TIMERx_DMATB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the TIMERx_DMATB, you will access the address of start address + 0x4. In a word: Start Address = TIMERx_CTL0 + DMASAR*4.

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address range (Start Addr + Transfer Timer* 4) will be accessed. The transfer Timer is calculated by hardware, and ranges from 0 to DMATC.

Channel input remap register (TIMERx_IRMP)

Address offset: 0x50

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CI0_RMP[1:0]	
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	CI0_RMP[1:0]	Channel 0 input remap 00: Channel 0 input is connected to GPIO (TIMER1_CH0) 01: Channel 0 input is connected to the LXTAL 10: Channel 0 input is connected to HXTAL/128 clock 11: Channel 0 input is connected to CK_OUT.

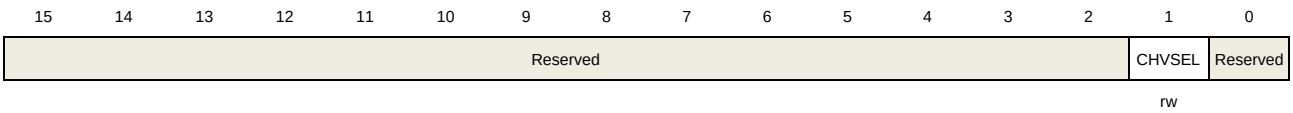
Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	Reserved	Must be kept at reset value.

18.3. Basic timer (TIMERx, x=5, 6)

18.3.1. Overview

The basic timer module (TIMER5/6) has a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate a DMA request and a TRGO to connect to DAC.

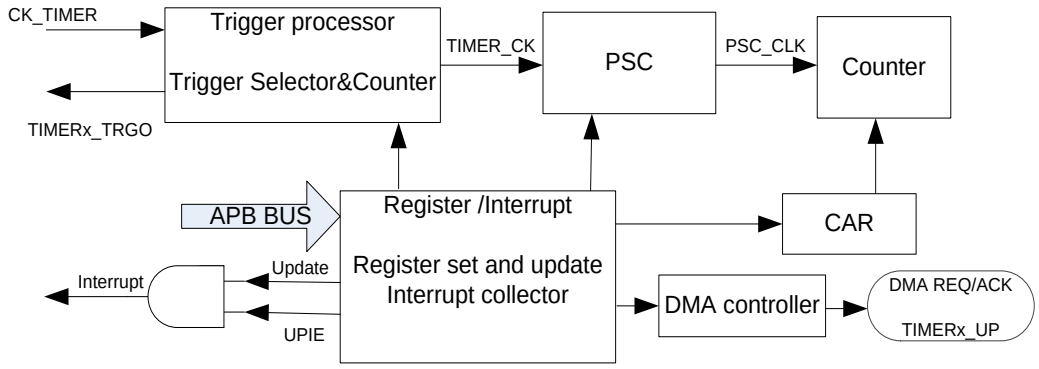
18.3.2. Characteristics

- Counter width: 16 bits.
- Source of count clock is internal clock only.
- Multiple counter modes: count up.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Auto reload function.
- Interrupt output or DMA request: update event.

18.3.3. Block diagram

Figure 18-61. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 18-61. Basic timer block diagram



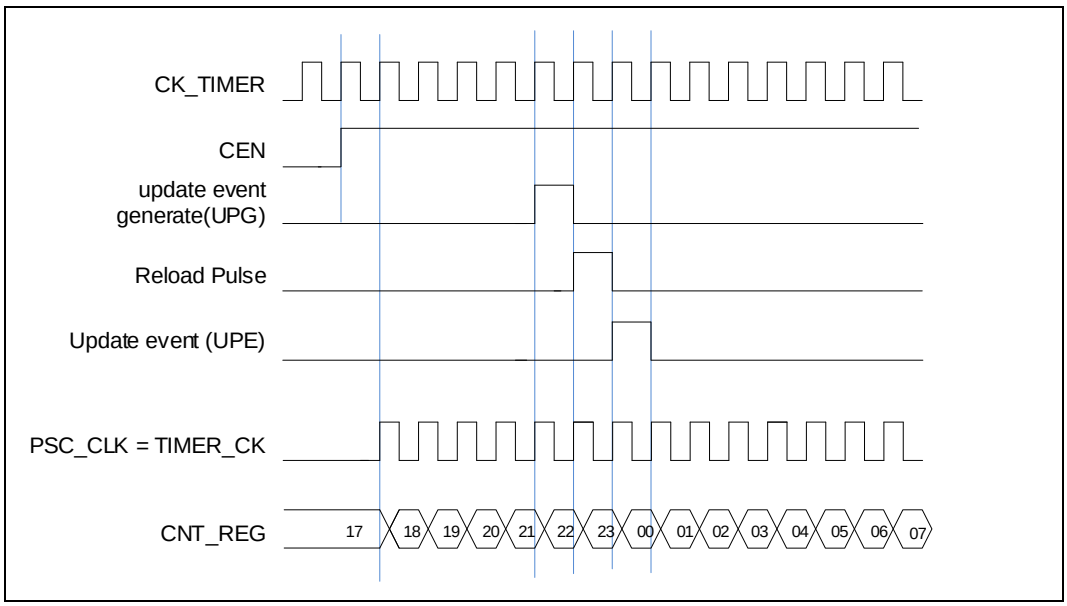
18.3.4. Function overview

Clock source configuration

The basic TIMER can only be clocked by the internal timer clock CK_TIMER, which is from the source named CK_TIMER in RCU

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

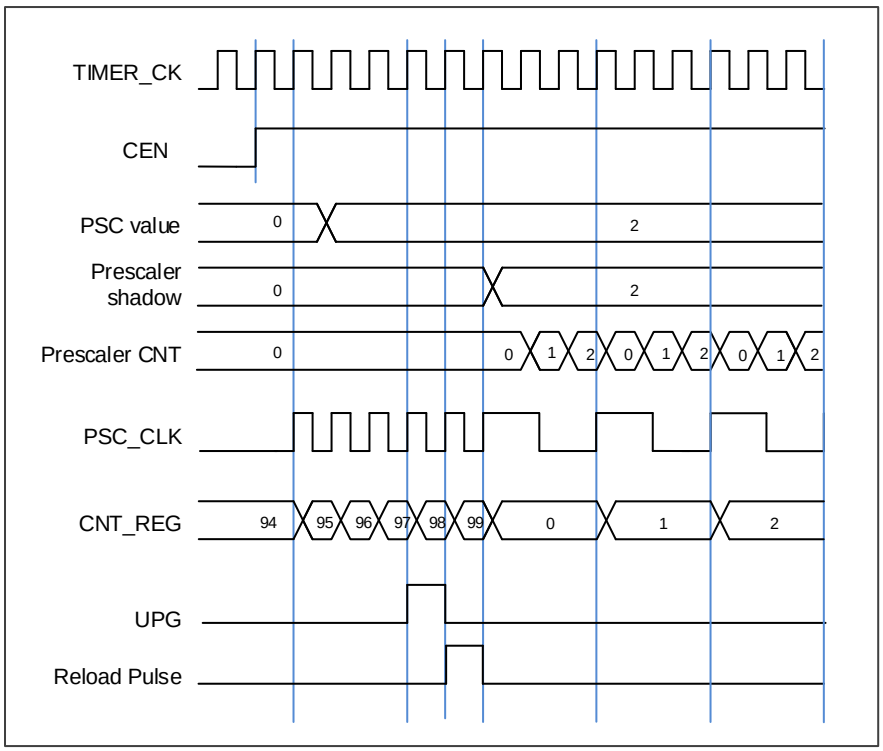
Figure 18-62. Timing chart of internal clock divided by 1



Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescaler factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 18-63. Timing chart of PSC value change from 0 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again. The update event is generated at each counter overflow. The counting direction bit DIR in the TIMERx_CTL1 register should be set to 0 for the up counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and generates an update event.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (auto reload register, prescaler register) are updated.

The following figures show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 18-64. Timing chart of up counting mode, PSC=0/2

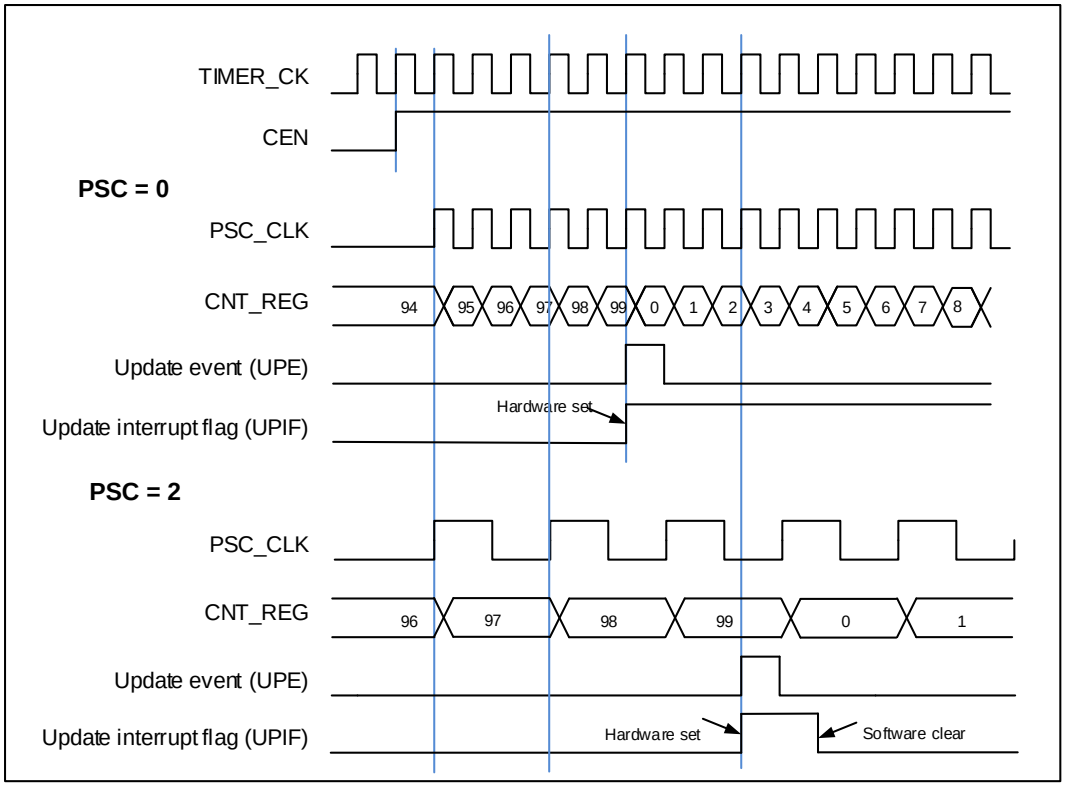
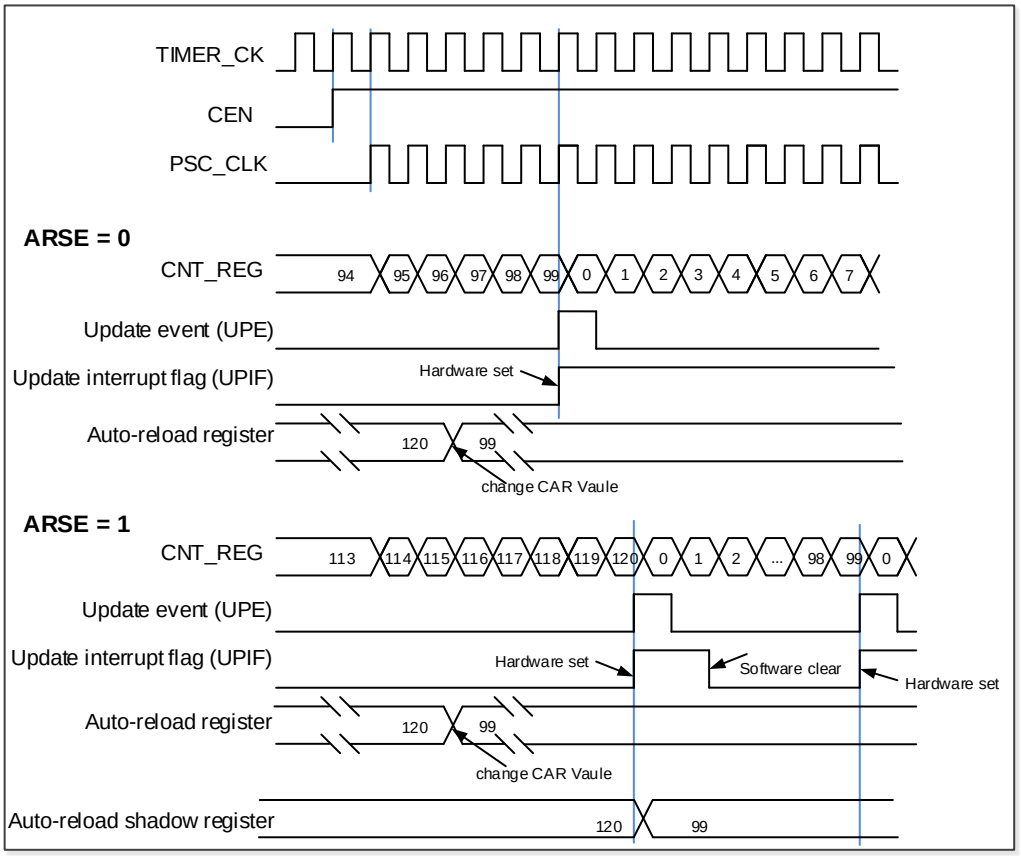


Figure 18-65. Timing chart of up counting mode, change TIMERx_CAR on the go



Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event.

Once the timer is set to operate in the single pulse mode, it is necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter, then the CEN bit keeps at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 using software, the counter will be stopped and its value held.

Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

18.3.5. Registers definition (TIMERx, x=5, 6)

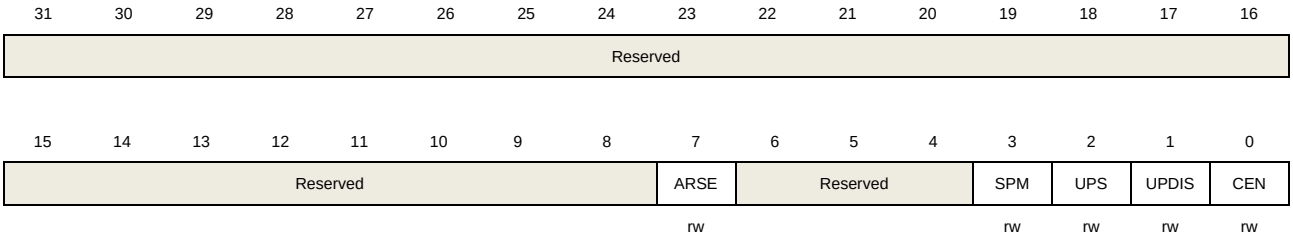
TIMER5 base address: 0x4000 1000

TIMER6 base address: 0x4000 1400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate an update interrupt or a DMA request: – The UPG bit is set – The counter generates an overflow or underflow event – The restart mode generates an update event. 1: This event generates update interrupts or DMA requests: The counter generates an overflow or underflow event
1	UPDIS	Update disable. This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event:

- The UPG bit is set
- The counter generates an overflow or underflow event
- The restart mode generates an update event.

1: Update event disable.

Note: When this bit is set to 1, setting UPG bit or the restart mode does not generate an update event, but the counter and prescaler are initialized.

0 CEN Counter enable

0: Counter disable

1: Counter enable

The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									MMC[2:0]			Reserved			
									rw						

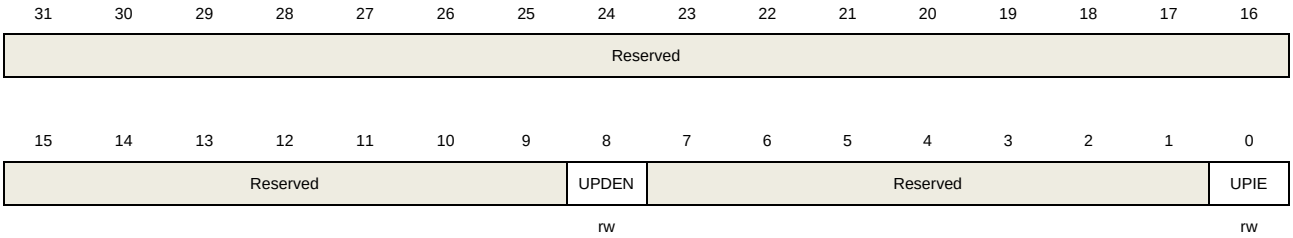
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 000: When a counter reset event occurs, a TRGO trigger signal is output. The counter reset source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a counter start event occurs, a TRGO trigger signal is output. The counter start source: CEN control bit is set The trigger input in pause mode is high 010: When an update event occurs, a TRGO trigger signal is output. The update source depends on UPDIS bit and UPS bit. 100~111: Reserved.
3:0	Reserved	Must be kept at reset value.

Interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



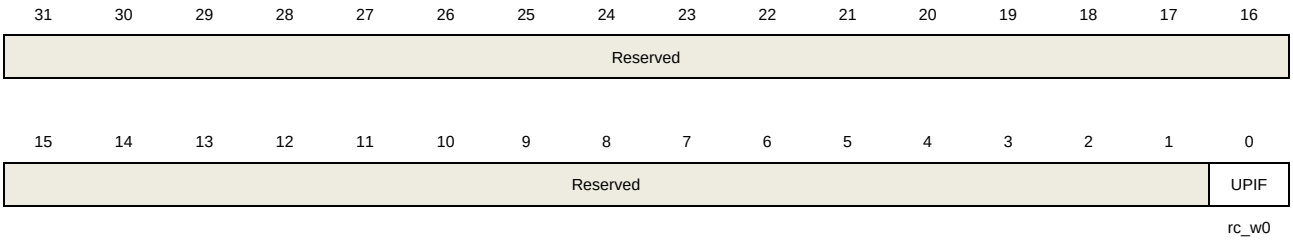
Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7:1	Reserved	Must be kept at reset value.
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

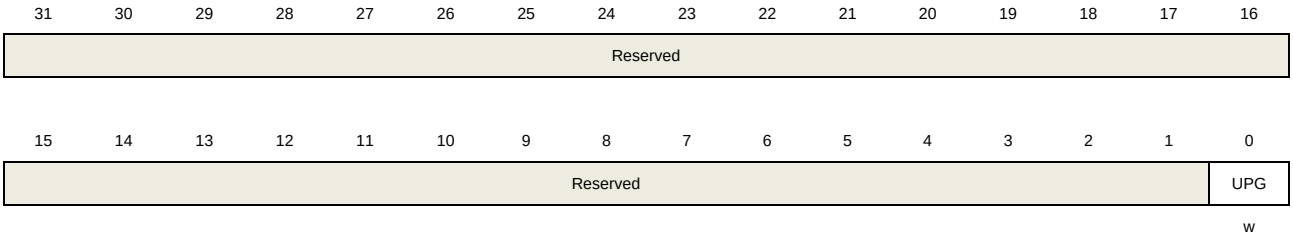


Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

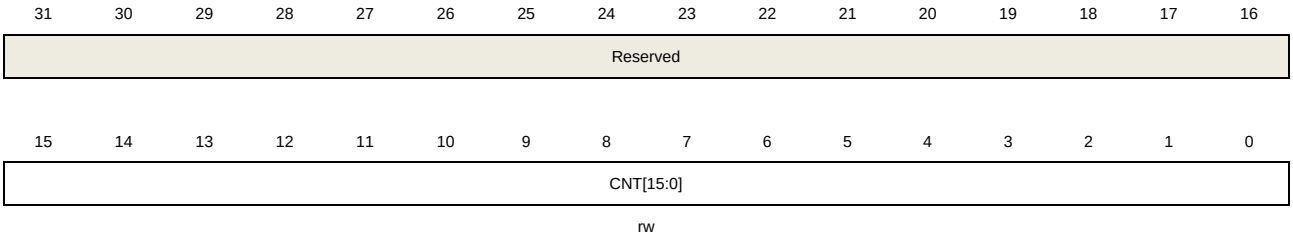


Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPG	This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter. **Note:** When the timer is configured in input capture mode, this register must be configured a non-zero value (such as 0xFFFF) which is larger than user expected value.

19. Universal synchronous/asynchronous receiver /transmitter (USART)

19.1. Overview

The Universal Synchronous/Asynchronous Receiver/Transmitter (USART) provides a flexible serial data exchange interface. Data frames can be transferred in full duplex or half duplex mode, synchronously or asynchronously through this interface. A programmable baud rate generator divides the UCLK (PCLK, CK_SYS, LXTAL or IRC16M) to produces a dedicated wide range baudrate clock for the USART transmitter and receiver.

Besides the standard asynchronous receiver and transmitter mode, the USART implements several other types of serial data exchange modes, such as IrDA (infrared data association) SIR mode, smartcard mode, LIN (local interconnection network) mode, half-duplex mode and synchronous mode. It also supports multiprocessor communication mode, and hardware flow control protocol (CTS/RTS). The data frame can be transferred from LSB or MSB bit. The polarity of the TX/RX pins can be configured independently and flexibly.

All USARTs support DMA function for high-speed data communication.

19.2. Characteristics

- NRZ standard format
- Asynchronous, full duplex communication
- Half duplex single wire communications
- Receive FIFO function
- Dual clock domain:
  - Asynchronous PCLK and USART clock
  - Baud rate programming independent from the PCLK reprogramming
- Programmable baud-rate generator allowing speed up to 12.5 MBits/s when the clock frequency is 100 MHz and oversampling is by 8
- Fully programmable serial interface characteristics:
  - A data word (8 or 9 bits) LSB or MSB first
  - Even, odd or no-parity bit generation/detection
  - 0.5, 1, 1.5 or 2 stop bit generation
- Swappable TX/RX pin
- Configurable data polarity
- Hardware Modem operations (CTS/RTS) and RS485 drive enable
- Configurable multibuffer communication using centralized DMA
- Separate enable bits for Transmitter and Receiver
- Parity control
  - Transmits parity bit

  - Checks parity of received data byte
- LIN Break generation and detection
- IrDA Support
- Synchronous mode and transmitter clock output for synchronous transmission
- ISO 7816-3 compliant smartcard interface
  - Character mode (T=0)
  - Block mode (T=1)
  - Direct and inverse convention
- Multiprocessor communication
  - Enter into mute mode if address match does not occur
  - Wake up from mute mode by idle line, address match detection or data match detection
- Support for ModBus communication
  - Timeout feature
  - CR/LF character recognition
- Wake up from Deep-sleep mode
  - By standard RBNE interrupt
  - By WUF interrupt
- Various status flags
  - Flags for transfer detection: Receive buffer not empty (RBNE), receive FIFO full (RFF), Transmit buffer empty (TBE), transfer complete (TC).
  - Flags for error detection: overrun error (ORERR), noise error (NERR), frame error (FERR) and parity error (PERR)
  - Flag for hardware flow control: CTS changes (CTSF)
  - Flag for LIN mode: LIN break detected (LBDF)
  - Flag for multiprocessor communication: IDLE frame detected (IDLEF)
  - Flag for ModBus communication: Address/character match (AMF) and receiver timeout (RTF)
  - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF)
  - Wakeup from Deep-sleep mode flag
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set

While USART0, USART1 and USART2 are fully implemented.

19.3. Function overview

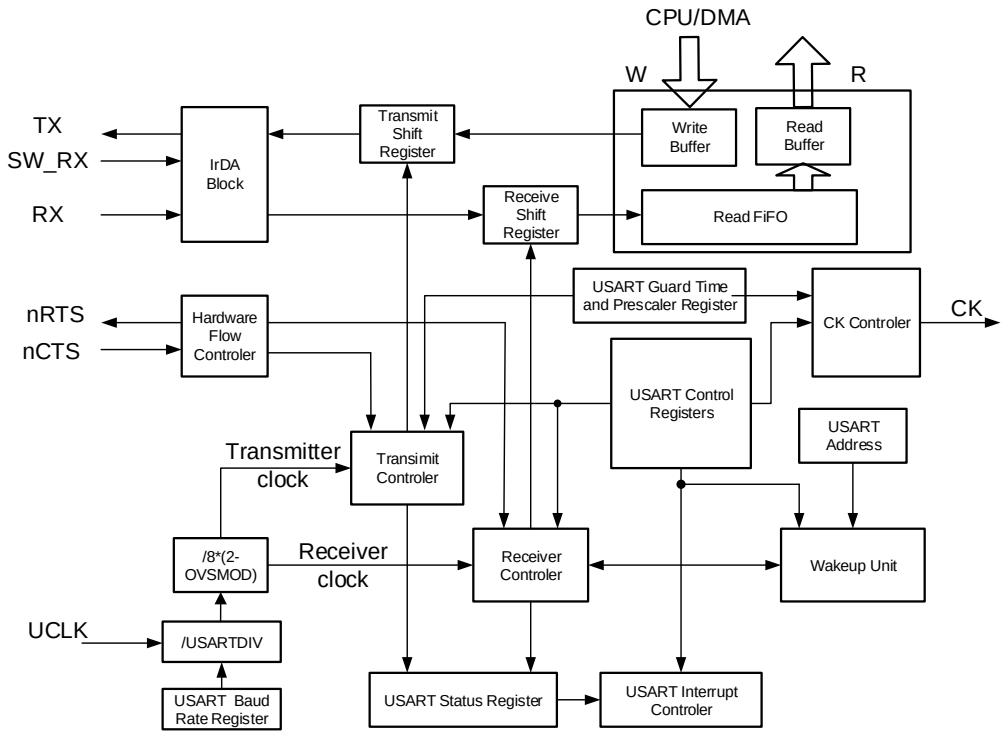
The interface is externally connected to another device by the main pins listed in ***Table 19-1. Description of USART important pins***.

Table 19-1. Description of USART important pins

Pin	Type	Description
RX	Input	Receive Data
TX	Output I/O (single-wire/smartcard mode)	Transmit Data. High level When enabled but nothing to be transmitted

Pin	Type	Description
CK	Output	Serial clock for synchronous communication
nCTS	Input	Clear to send in Hardware flow control mode
nRTS	Output	Request to send in Hardware flow control mode

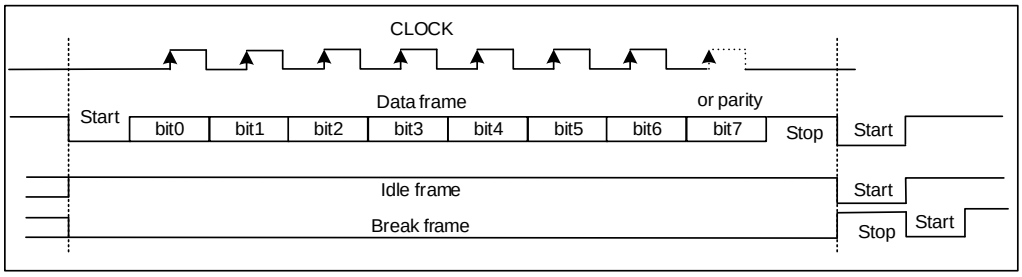
Figure 19-1. USART module block diagram



19.3.1. USART frame format

The USART frame starts with a start bit and ends up with a number of stop bits. The length of the data frame is configured by the WL bit in the USART_CTL0 register. The last data bit can be used as parity check bit by setting the PCEN bit of in USART_CTL0 register. When the WL bit is reset, the parity bit is the 7th bit. When the WL bit is set, the parity bit is the 8th bit. The method of calculating the parity bit is selected by the PM bit in USART_CTL0 register.

Figure 19-2. USART character frame (8 bits data and 1 stop bit)



In transmission and reception, the number of stop bits can be configured by the STB[1:0] bits in the USART_CTL1 register.

Table 19-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	Default value
01	0.5	Smartcard mode for receiving
10	2	Normal USART and single-wire modes
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK, the configuration of the baud rate generator and the oversampling mode.

19.3.2. Baud rate generation

The baud-rate divider is a 16-bit number which consists of a 12-bit integer and a 4-bit fractional part. The number formed by these two values is used by the baud rate generator to determine the bit period. Having a fractional baud-rate divider allows the USART to generate all the standard baud rates.

The baud-rate divider (USARTDIV) has the following relationship with the peripheral clock:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (19\text{-}1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (19\text{-}2)$$

For example, when oversampled by 16:

1. Get USARTDIV by caculating the value of USART_BUAD:
   If USART_BUAD=0x21D, then INTDIV=33 (0x21), FRADIV=13 (0xD).
   USARTDIV=33+13/16=33.81.
2. Get the value of USART_BUAD by calculating the value of USARTDIV:
   If USARTDIV=30.37, then INTDIV=30 (0x1E).
   16*0.37=5.92, the nearest integer is 6, so FRADIV=6 (0x6).
   USART_BUAD=0x1E6.

   Note: If the roundness of FRADIV is 16 (overflow), the carry must be added to the integer part.

19.3.3. USART transmitter

If the transmit enable bit (TEN) in USART_CTL0 register is set, when the transmit data buffer is not empty, the transmitter shifts out the transmit data frame through the TX pin. The polarity

of the TX pin can be configured by the TINV bit in the USART_CTL1 register. Clock pulses can output through the CK pin.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while the transmission is ongoing.

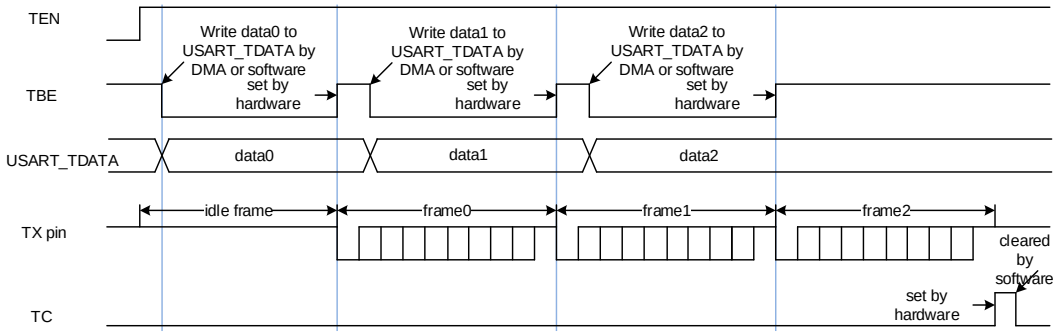
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit in the USART_STAT register is asserted. The TBE bit is cleared by writing USART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the USART_TDATA register while a transmission is ongoing, it will be firstly stored in the transmit buffer, and transferred to the transmit shift register after the current transmission is done. If a data is written to the USART_TDATA register while no transmission is ongoing, the TBE bit will be cleared and set soon, because the data will be transferred to the transmit shift register immediately.

If a frame is transmitted and the TBE bit is asserted, the TC bit of the USART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in ***Figure 19-3. USART transmit procedure***. The software operating process is as follows:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the TEN bit in USART_CTL0.
7. Wait for the TBE being asserted.
8. Write the data to the USART_TDATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 19-3. USART transmit procedure



It is necessary to wait for the TC bit to be asserted before disabling the USART or entering the power saving mode. This bit can be cleared by set the TCC bit in USART_INTC register.

The break frame is sent when the SBKCMD bit is set, and SBKCMD bit is reset after the transmission.

19.3.4. USART receiver

After power on, the USART receiver can be enabled by the following procedure:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1.
3. Enable DMA (DENR bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the REN bit in USART_CTL0.

After being enabled, the receiver receives a bit stream after a valid start pulse has been detected. Detection on noisy error, parity error, frame error and overrun error is performed during the reception of a frame.

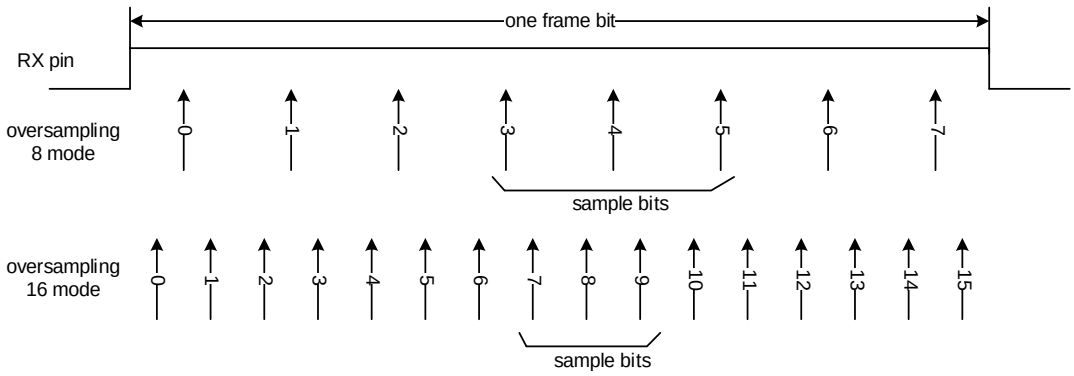
When a frame is received, the RBNE bit in USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

The software can get the received data by reading the USART_RDATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_RDATA register, whatever it is performed by software directly, or through DMA.

The REN bit should not be disabled when reception is ongoing, or the current frame will be lost.

By default, the receiver gets three samples to evaluate the value of a frame bit. If the oversampling 8 mode is enabled, the 3rd, 4th and 5th samples are used, while in the oversampling 16 mode, the 7th, 8th, and 9th samples are used. If two or more samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the value of the three samples of any bit are not the same, whatever it is a start bit, data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame. An interrupt will be generated, If the ERRIE bit in USART_CTL2 register is set. If the OSB bit in USART_CTL2 register is set, the receiver gets only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 19-4. Oversampling method of a receive frame bit (OSB=0)



If the parity check function is enabled by setting the PCEN bit in the USART_CTL0 register, the receiver calculates the expected parity value while receiving a frame. The received parity

bit will be compared with this expected value. If they are not the same, the parity error (PERR) bit in USART_STAT register will be set. An interrupt is generated, if the PERRIE bit in USART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in USART_STAT register will be set. An interrupt is generated, If the ERRIE bit in USART_CTL2 register is set. According to the configuration of the stop bit, there are the following situations:

- 0.5 stop bit: When 0.5 stop bit, stop bit is not sampled.
- 1 stop bit: When 1 stop bit, sampling in the middle of stop bit.
- 1.5 stop bits: When 1.5 stop bits, the 1.5 stop bits are divided into 2 parts: the 0.5 stop bit part is not sampled and sampling in the middle of 1 stop bit.
- 2 stop bits: When 2 stop bits, if a frame error is detected during the first stop bit, the frame error flag is set, the second stop bit is not checked frame error. If no frame error is detected during the first stop bit, then continue to check the second stop bit for frame error.

When a frame is received, if the RBNE bit is not cleared yet, the last frame will not be stored in the receive data buffer. The overrun error (ORERR) bit in USART_STAT register will be set. An interrupt is generated, if the ERRIE bit in USART_CTL2 register is set, or if the RBNEIE is set.

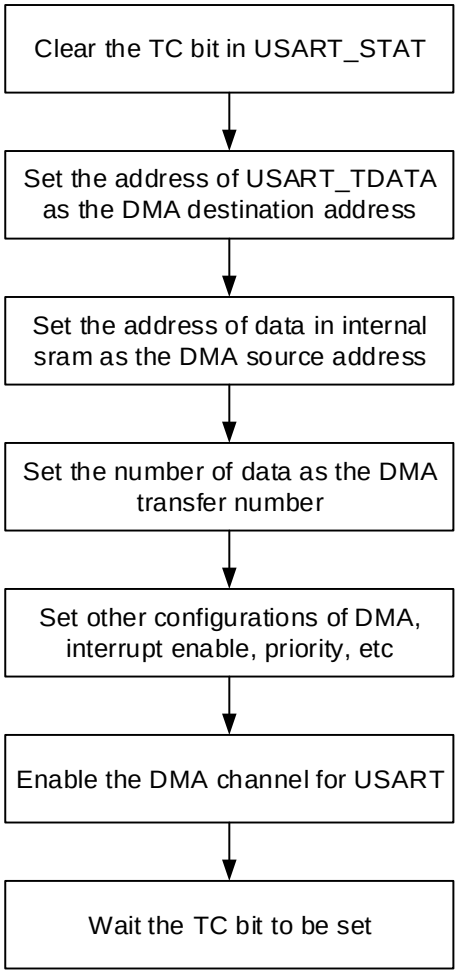
The RBNE, NERR, PERR, FERR and ORERR flags are always set at the same time in a reception. If the receive DMA is not enabled, software can check NERR, PERR, FERR and ORERR flags when serving the RBNE interrupt.

19.3.5. Use DMA for data buffer access

To reduce the burden of the processor, DMA can be used to access the transmitting and receiving data buffer. The DENT bit in USART_CTL2 is used to enable the DMA transmission, and the DENR bit in USART_CTL2 is used to enable the DMA reception.

When DMA is used for USART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the USART. The configuration step are shown in ***Figure 19-5. Configuration step when using DMA for USART transmission***.

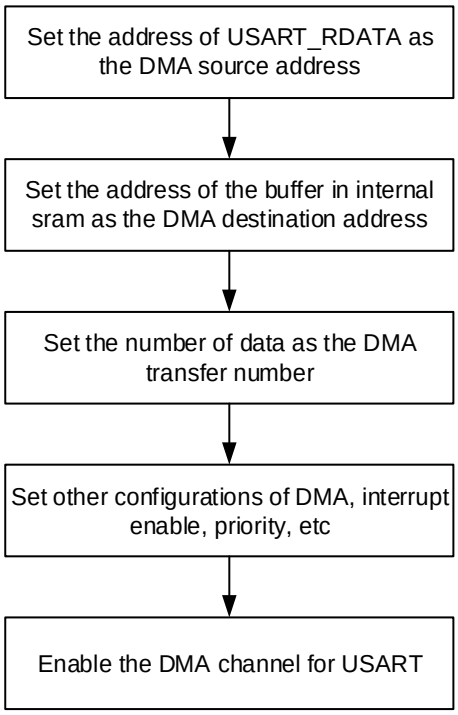
Figure 19-5. Configuration step when using DMA for USART transmission



After all of the data frames are transmitted, the TC bit in USART_STAT is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

When DMA is used for USART reception, DMA transfers data from the receive data buffer of the USART to the internal SRAM. The configuration steps are shown in ***Figure 19-6. Configuration step when using DMA for USART reception***. If the ERRIE bit in USART_CTL2 is set, interrupts can be generated by the Error status bits (FERR, ORERR and NERR) in USART_STAT.

Figure 19-6. Configuration step when using DMA for USART reception

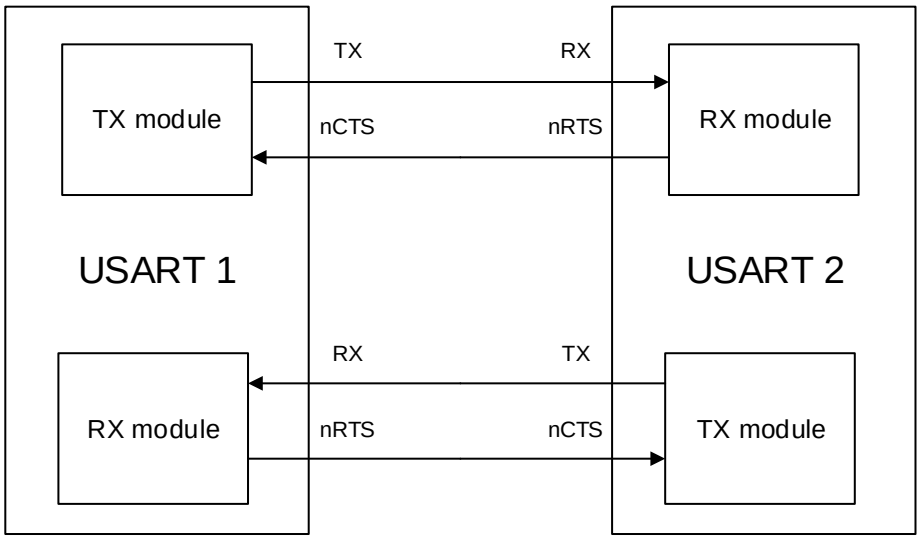


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

19.3.6. Hardware flow control

The hardware flow control function is realized by the nCTS and nRTS pins. The RTS flow control is enabled by writing ‘1’ to the RTSEN bit in USART_CTL2 and the CTS flow control is enabled by writing ‘1’ to the CTSEN bit in USART_CTL2.

Figure 19-7. Hardware flow control between two USARTs



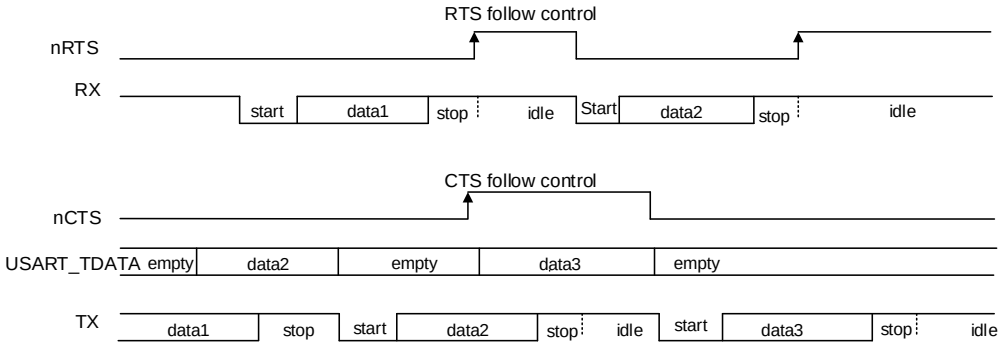
RTS flow control

The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When data frame is received, the nRTS signal goes high to prevent the transmitter from sending next frame. The nRTS signal keeps high when the receive buffer is full.

CTS flow control

The USART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in USART_STAT is '0' and the nCTS signal is low, the transmitter transmits the data frame. When the nCTS signal goes high during a transmission, the transmitter stops after the current transmission is accomplished.

Figure 19-8. Hardware flow control



RS485 Driver Enable

The driver enable feature, which is enabled by setting bit DEM in the USART_CTL2 control register, allows the user to activate the external transceiver control, through the DE (Driver Enable) signal. The assertion time, which is programmed using the DEA [4:0] bits field in the USART_CTL0 control register, is the time between the activation of the DE signal and the beginning of the START bit. The de-assertion time, which is programmed using the DED [4:0] bits field in the USART_CTL0 control register, is the time between the end of the last stop bit and the de-activation of the DE signal. The polarity of the DE signal can be configured using the DEP bit in the USART_CTL2 control register.

19.3.7. Multi-processor communication

In multiprocessor communication, several USARTs are connected as a network. It will be a big burden for a device to monitor all of the messages on the RX pin. To reduce the burden of a device, software can put an USART module into a mute mode by writing 1 to the MMCMD bit in USART_CMD register.

If a USART is in mute mode, all of the receive status bits cannot be set. The USART can also be wake up by hardware by one of the three methods: idle frame method, address match method and data match method.

The idle frame wake up method is selected by default. If the RWU bit is reset, an idle frame

is detected on the RX pin, the IDLEF bit in USART_STAT will be set. If the RWU bit is set, an idle frame is detected on the RX pin, the hardware clears the RWU bit and exits the mute mode. When it is woken up by an idle frame, the IDLEF bit in USART_STAT will not be set.

When the WM[1:0] of in USART_CTL0 register is 0b01, the USART be waken up by address match method. If the address flag is high, the frame is treated as an address frame. If the address flag is low, the frame is treated as a data frame. If the LSB 4 or 7 bits, which are configured by the ADDM bit of the USART_CTL1 register, of an address frame is the same as the ADDR_DATA bits in the USART_CTL1 register, the hardware will clear the RWU bit and exits the mute mode. The status bits are available in the USART_STAT register. If the LSB 4 or 7 bits of an address frame defers from the ADDR_DATA bits in the USART_CTL1 register, the hardware sets the RWU bit and enters mute mode automatically. In this situation, the RBNE bit is not set.

When the WM[1:0] of in USART_CTL0 register is 0b1x, the USART be waken up by data match method, in this situation, the ADDM bit must be set. If the data frame is the same as the ADDR_DATA[7:0] bits in the USART_CTL1 register, the hardware will clear the RWU bit and exits the mute mode. The RBNE bit will be set when the frame that wakes up the USART. In these situations, the RBNE bit is not set. If the frame defers from the ADDR_DATA bits in the USART_CTL1 register, the hardware sets the RWU bit and enters mute mode automatically. In this situation, the RBNE bit is not set.

If the PCEN bit in USART_CTL0 is set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address or data bit. When the USART will be waken up by address match, if the ADDM bit is set and the receive frame is a 7bit data, the LSB 6 bits will be compared with ADDR_DATA[5:0]. If the ADDM bit is set and the receive frame is a 9bit data, the LSB 8 bits will be compared with ADDR_DATA[7:0]. When the USART will be waken up by data match, if the the receive frame is a 7bit data, the LSB 6 bits will be compared with ADDR_DATA[5:0]. If the receive frame is a 9bit data, the LSB 8 bits will be compared with ADDR_DATA[7:0].

Note: If the MEN bit is set, the WM[1:0] bits is reset and the RWU bit is reset, an idle frame is detected on the RX pin, the IDLEF bit will be set. If the RWU bit is set, the IDLEF is not set.

19.3.8. LIN mode

The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

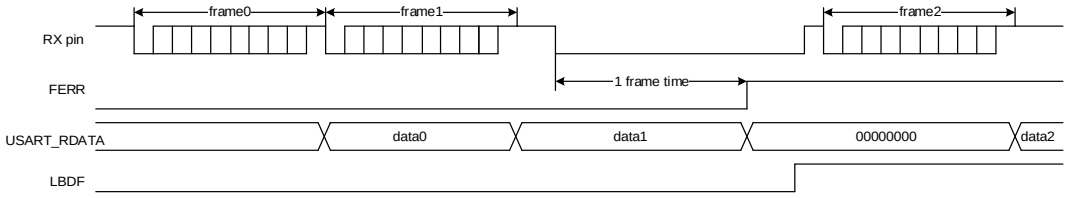
When transmitting a normal data frame, the transmission procedure is the same as the normal USART mode. The data bits length can only be 8. And the break frame is 13-bit ’0’, followed by 1 stop bit.

The break detection function is totally independent of the normal USART receiver. So a break frame can be detected during the idle state or during a frame. The expected length of a break frame can be selected by configuring LBLEN in USART_CTL1. When the RX pin is detected

at low state for a time that is equal to or longer than the expected break frame length (10 bits when LBLEN=0, or 11 bits when LBLEN=1), the LBDF bit in USART_STAT is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

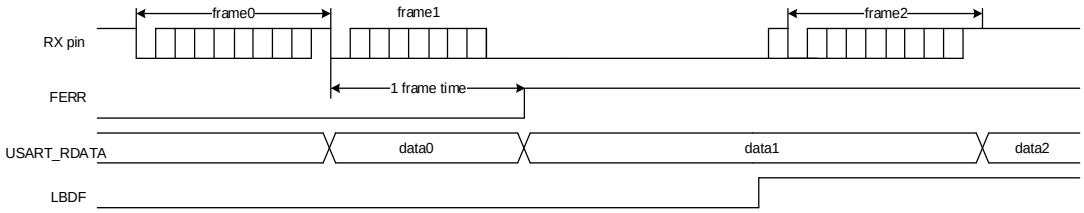
As shown in ***Figure 19-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 19-9. Break frame occurs during idle state



As shown in ***Figure 19-10. Break frame occurs during a frame***, if a break frame occurs during a frame on the RX pin, the FERR status will be asserted for the current frame.

Figure 19-10. Break frame occurs during a frame



19.3.9. Synchronous mode

The USART can be used for full-duplex synchronous serial communications only in master mode, by setting the CKEN bit in USART_CTL1. The LMEN bit in USART_CTL1 and SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in synchronous mode. The CK pin is the clock output of the synchronous USART transmitter, and can be only activated when the TEN bit is enabled. No clock pulse will be sent through the CK pin during the transmission of the start bit and stop bit. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the last (address flag) bit transmission. The clock output is also not activated during idle and break frame sending. The CPH bit in USART_CTL1 can be used to determine whether data is captured on the first or the second clock edge. The CPL bit in USART_CTL1 can be used to configure the clock polarity in the USART Synchronous idle state.

The CPL, CPH and CLEN bits in USART_CTL1 determine the waveform on the CK pin. Software can only change them when the USART is disabled (UEN=0).

The clock is synchronized with the data transmitted. The receiver in synchronous mode samples the data on the transmitter clock without any oversampling.

Figure 19-11. Example of USART in synchronous mode

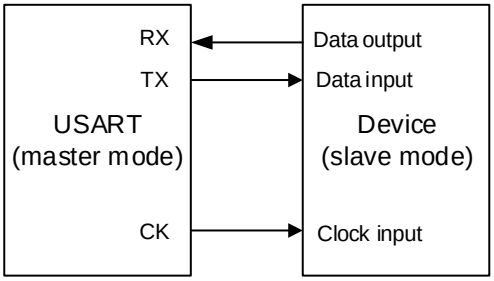
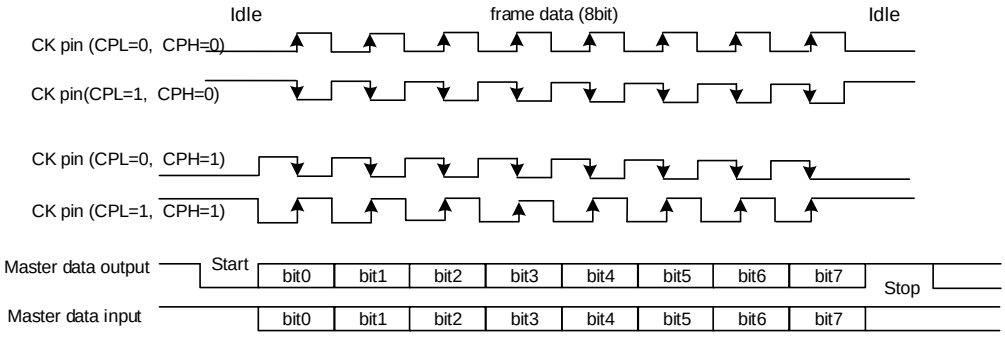


Figure 19-12. 8-bit format USART synchronous waveform (CLEN=1)

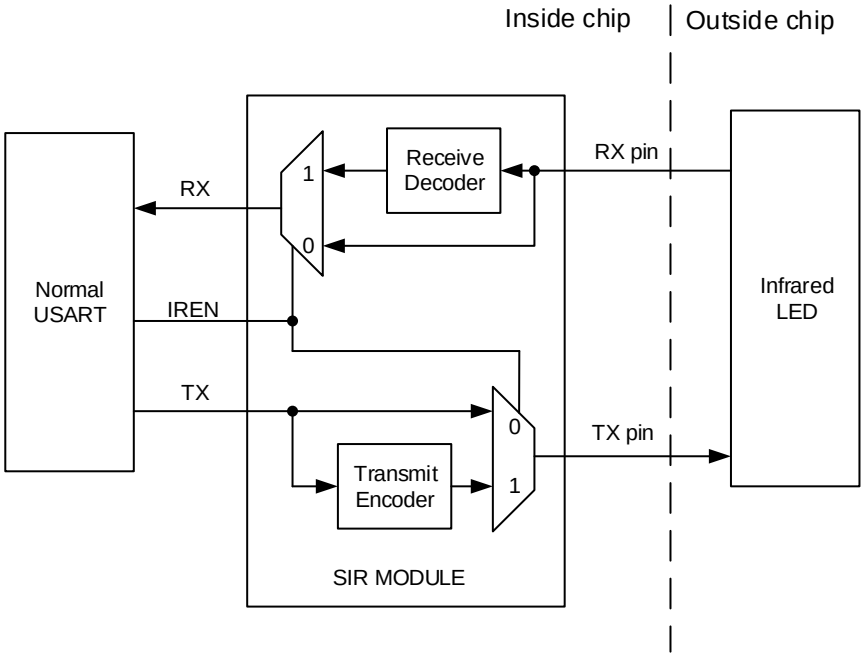


19.3.10. IrDA SIR ENDEC mode

The IrDA mode is enabled by setting the IREN bit in USART_CTL2. The LMEN, STB[1:0], CKEN bits in USART_CTL1 and HDEN, SCEN bits in USART_CTL2 should be cleared in IrDA mode.

In IrDA mode, the USART transmission data frame is modulated in the SIR transmit encoder and transmitted to the infrared LED through the TX pin. The SIR receive decoder receives the modulated signal from the infrared LED through the RX pin, and puts the demodulated data frame to the USART receiver. The baud rate should not be larger than 115200 for the encoder.

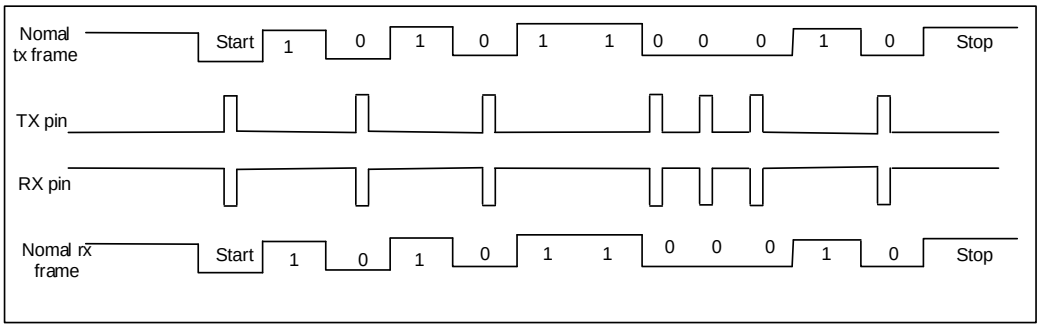
Figure 19-13. IrDA SIR ENDEC module



In IrDA mode, the polarity of the TX and RX pins is different. The TX pin is usually at low state, while the RX pin is usually at high state. The IrDA pins keep stable to represent the logic '1', while an infrared light pulse on the IrDA pins (a Return to Zero signal) represents the logic '0'. The pulse width should be 3/16 of a bit period. The IrDA could not detect any pulse if the pulse width is less than 1 PSC clock. While it can detect a pulse by chance if the pulse width is greater than 1 but smaller than 2 times of PSC clock.

Because the IrDA is a half-duplex protocol, the transmission and the reception should not be carried out at the same time in the IrDA SIR ENDEC block.

Figure 19-14. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The division ratio is configured by the PSC[7:0] bits in USART_GP register. The pulse width on the TX pin is 3 cycles of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

19.3.11. Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in USART_CTL2. The LMEN, CKEN bits in USART_CTL1 and SCEN, IREN bits in USART_CTL2 should be cleared in half-duplex communication mode.

Only one wire is used in half-duplex mode. The TX and RX pins are connected together internally, and the RX pin is no longer used. The TX pin should be configured as open drain mode and communication conflicts are handled by software. When TEN is set, the data in the data register will be sent.

19.3.12. Smartcard (ISO7816-3) mode

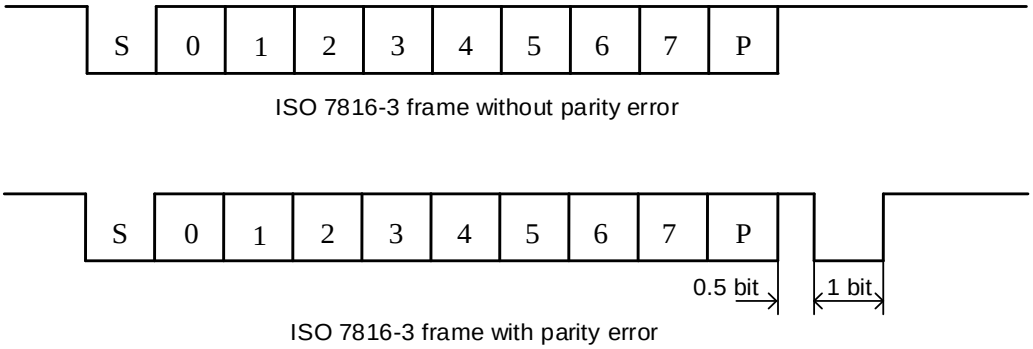
The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T=0) mode and the block (T=1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be reset in smartcard mode.

A clock is provided to the smartcard if the CKEN bit is set. The clock can be divided for other use.

The frame consists of 1 start bit, 9 data bits (1 parity bit included) and 1.5 stop bits.

The smartcard mode is a half-duplex communication protocol. When connected to a smartcard, the TX pin must be configured as open drain mode, and drives a bidirectional line that is also driven by the smartcard.

Figure 19-15. ISO7816-3 frame format



Character (T=0) mode

Compared to the timing in normal operation, the transmission time from transmit shift register to the TX pin is delayed by half baud clock, and the TC flag assertion time is delayed by a guard time that is configured by the GUAT[7:0] bits in USART_GP. In Smartcard mode, the internal guard time counter starts counting up after the stop bits of the last data frame, and the GUAT[7:0] bits should be configured as the character guard time (CGT) in ISO7816-3 protocol minus 12. The TC status is forced reset while the guard time counter is counting up. When the counter reaches the programmed value TC is asserted high.

During USART transmission, if a parity error event is detected, the smartcard may NACK the current frame by pulling down the TX pin during the last 1 bit time of the stop bits. The USART can automatically resend data according to the protocol for SCRTNUM times. An interframe gap of 2.5 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the frame error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE/receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the Smartcard mode.

Block (T=1) mode

In block (T=1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. A timeout interrupt will be generated, if no answer is received from the card before the expiration of this period. If the first character is received before the expiration of the period, it is signaled by the RBNE interrupt. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

In order to allow the automatic check of the maximum wait time between two consecutive characters, the USART_RT register must be programmed to the CWT (character wait time) - 11 value, which is expressed in baudtime units, after the reception of the first character (RBNE interrupt). The USART signals to the software through the RT flag and interrupt (when RTIE bit is set), if the smartcard doesn’t send a new character in less than the CWT period after the end of the previous character.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE=0), to count the number of received characters. The length of the block, which must be programmed in the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). This register field must be programmed to the minimum value (0x0), before the start of the block, when using DMA mode. With this value, an interrupt is generated after the 4th received character. The software must read the third byte as block length from the receive buffer.

In interrupt driven receive mode, the length of the block may be checked by software or by programming the BL value. However, before the start of the block, the maximum value of BL

(0xFF) may be programmed. The real value will be programmed after the reception of the third character.

The total block length (including prologue, epilogue and information fields) equals BL+4. The end of the block is signaled to the software through the EBF flag and interrupt (when EBIE bit is set). The RT interrupt may occur in case of an error in the block length.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

The direct convention is defined as: LSB first, logical bit value of 1 corresponds to H state of the line and parity is even. In this case, the following control bits must be programmed: MSBF=0, DINV=0 (default values).

The inverse convention is defined as: MSB first, logical bit value 1 corresponds to an L state on the signal line and parity is even. In this case, the following control bits must be programmed: MSBF=1, DINV=1.

19.3.13. ModBus communication

The USART offers basic support for the implementation of ModBus/RTU and ModBus/ASCII protocols by implementing an end of block detection.

In the ModBus/RTU mode, the end of one block is recognized by an idle line for more than 2 characters time. This function is implemented through the programmable timeout function.

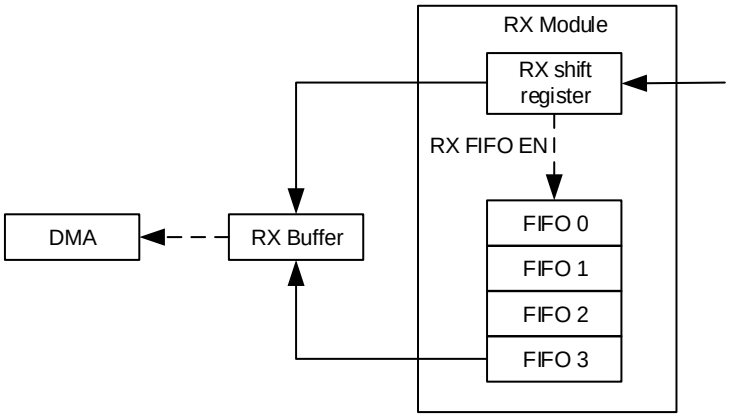
To detect the idle line, the RTEN bit in the USART_CTL1 register and the RTIE in the USART_CTL0 register must be set. The USART_RT register must be set to the value corresponding to a timeout of 2 characters time. After the last stop bit is received, when the receive line is idle for this duration, an interrupt will be generated, informing the software that the current block reception is completed.

In the ModBus/ASCII mode, the end of a block is recognized by a specific (CR/LF) character sequence. The USART manages this mechanism using the character match function by programming the LF ASCII code in the ADDR field and activating the address match interrupt (AMIE=1). When a LF has been received or can check the CR/LF in the DMA buffer, the software will be informed.

19.3.14. Receive FIFO

The receive FIFO can be enabled by setting the RFEN bit of the USART_RFCS register to avoid the overrun error when the CPU can’t serve the RBNE interrupt immediately. Up to 5 frames receive data can be stored in the receive FIFO and receive buffer. The RFFINT flag will be set when the receive FIFO is full. An interrupt is generated if the RFFIE bit is set.

Figure 19-16. USART Receive FIFO structure



If the software read receive data buffer in the routing of the RBNE interrupt, the RBNEIE bit should be reset at the beginning of the routing and set after all of the receive data is read out. The PERR/NERR/FERR/EBF flags should be cleared before reading a receive data out.

19.3.15. Wakeup from Deep-sleep mode

The USART is able to wake up the MCU from Deep-sleep mode by the standard RBNE interrupt, the WUM interrupt or data match method.

The UESM bit must be set and the USART clock must be set to IRC8M or LXTAL (refer to the reset and clock unit RCU section).

When using the standard RBNE interrupt, the RBNEIE bit must be set before entering Deep-sleep mode.

When using the WUIE interrupt, the source of WUIE interrupt may be selected through the WUM bit fields.

DMA must be disabled before entering Deep-sleep mode. Before entering Deep-sleep mode, software must check that the USART is not performing a transfer, by checking the BSY flag in the USART_STAT register. The REA bit must be checked to ensure the USART is actually enabled.

When the wakeup event is detected, the WUF flag is set by hardware and a wakeup interrupt is generated if the WUIE bit is set, independently of whether the MCU is in stop or active mode.

19.3.16. USART interrupts

The USART interrupt events and flags are listed in ***Table 19-3. USART interrupt requests***.

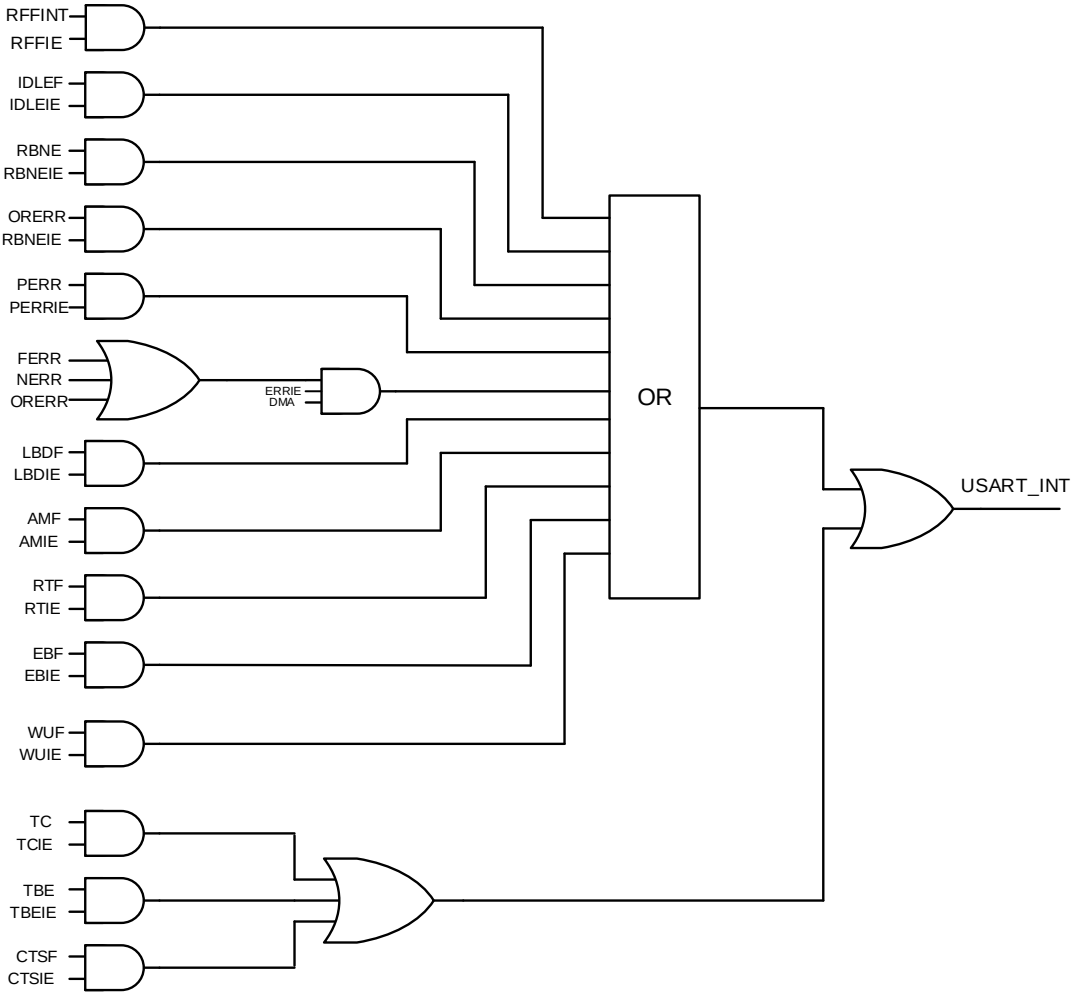
Table 19-3. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty	TBE	TBEIE
CTS flag	CTSF	CTSIE
Transmission complete	TC	TCIE

Interrupt event	Event flag	Enable Control bit
Received data ready to be read	RBNE	RBNEIE
Overrun error detected	ORERR	
Receive FIFO full	RFFINT	RFFIE
Idle line detected	IDLEF	IDLEIE
Parity error flag	PERR	PERRIE
Break detected flag in LIN mode	LBDF	LBDIE
Reception Errors (Noise flag, overrun error, framing error)	NERR or ORERR or FERR	ERRIE
Character match	AMF	AMIE
Receiver timeout error	RTF	RTIE
End of Block	EBF	EBIE
Wakeup from Deep-sleep mode	WUF	WUIE

All of the interrupt events are ORed together before being sent to the interrupt controller, so the USART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine

Figure 19-17. USART interrupt mapping diagram



19.4. Register definition

USART0 base address: 0x4001 3800
USART1 base address: 0x4000 4400
USART2 base address: 0x4000 4800

19.4.1. Control register 0 (USART_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			WM1	EBIE	RTIE	DEA[4:0]					DED[4:0]				
			rw	rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
28	WM1	Wakeup method in mute mode, this bit with bit[11] determines the wakeup method. 00: Idle line. 01: Address match. 1x: Data match. This bit field cannot be written when the USART is enabled (UEN=1).
27	EBIE	End of Block interrupt enable. 0: End of Block interrupt is disabled. 1: End of Block interrupt is enabled.
26	RTIE	Receiver timeout interrupt enable. 0: Receiver timeout interrupt is disabled. 1: Receiver timeout interrupt is enabled.
25:21	DEA[4:0]	Driver Enable assertion time. These bits are used to define the time between the activation of the DE (Driver Enable) signal and the beginning of the start bit. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN=1).
20:16	DED[4:0]	Driver Enable de-assertion time. These bits are used to define the time between the end of the last stop bit, in a transmitted message, and the de-activation of the DE (Driver Enable) signal. It is

		expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN=1).
15	OVSMOD	Oversample mode. 0: Oversampling by 16. 1: Oversampling by 8. This bit must be kept cleared in LIN, IrDA and smartcard modes. This bit field cannot be written when the USART is enabled (UEN=1).
14	AMIE	ADDR match interrupt enable. 0: ADDR match interrupt is disabled. 1: ADDR match interrupt is enabled.
13	MEN	Mute mode enable. 0: Mute mode disabled. 1: Mute mode enabled.
12	WL	Word length. 0: 8 Data bits. 1: 9 Data bits. This bit field cannot be written when the USART is enabled (UEN=1).
11	WM0	Wakeup method in mute mode, this bit with bit[28] determines the wakeup method. This bit field cannot be written when the USART is enabled (UEN=1).
10	PCEN	Parity control enable. 0: Parity control disabled. 1: Parity control enabled. This bit field cannot be written when the USART is enabled (UEN=1).
9	PM	Parity mode. 0: Even parity. 1: Odd parity. This bit field cannot be written when the USART is enabled (UEN=1).
8	PERRIE	Parity error interrupt enable. 0: Parity error interrupt is disabled. 1: An interrupt will occur whenever the PERR bit is set in USART_STAT.
7	TBEIE	Transmitter register empty interrupt enable. 0: Interrupt is inhibited. 1: An interrupt will occur whenever the TBE bit is set in USART_STAT.
6	TCIE	Transmission complete interrupt enable. If this bit is set, an interrupt occurs when the TC bit in USART_STAT is set. 0: Transmission complete interrupt is disabled. 1: Transmission complete interrupt is enabled.

5	RBNEIE	Read data buffer not empty interrupt and overrun error interrupt enable. 0: Read data register not empty interrupt and overrun error interrupt disabled. 1: An interrupt will occur whenever the ORERR bit is set or the RBNE bit is set in USART_STAT.
4	IDLEIE	IDLE line detected interrupt enable. 0: IDLE line detected interrupt disabled. 1: An interrupt will occur whenever the IDLEF bit is set in USART_STAT.
3	TEN	Transmitter enable. 0: Transmitter is disabled. 1: Transmitter is enabled.
2	REN	Receiver enable. 0: Receiver is disabled. 1: Receiver is enabled and begins searching for a start bit.
1	UESM	USART enable in Deep-sleep mode. 0: USART not able to wake up the MCU from Deep-sleep mode. 1: USART able to wake up the MCU from Deep-sleep mode. Providing that the clock source for the USART must be IRC8M or LXTAL.
0	UEN	USART enable. 0: USART prescaler and outputs disabled. 1: USART prescaler and outputs enabled.

19.4.2. Control register 1 (USART_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR_DATA[7:0]								RTEN	Reserved			MSBF	DINV	TINV	RINV
rw								rw				rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	Reserved	LBDIE	LBLEN	ADDM	Reserved			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

Bits	Fields	Descriptions
31:24	ADDR_DATA[7:0]	Address or data of the USART terminal. These bits give the address or data of the USART terminal. In multiprocessor communication during mute mode or Deep-sleep mode, this is used for wakeup with address match or data match detection. The received frame will be compared to these bits. When WM[1:0] = 01, if the ADDM bit is reset, only the ADDR[3:0] bits are used to compare.

		In normal reception, these bits are also used for character detection. The whole received character (8-bit) is compared to the ADDR[7:0] value and AMF flag is set on matching. This bit field cannot be written when both reception (REN=1) and USART (UEN=1) are enabled.
23	RTEN	Receiver timeout enable. 0: Receiver timeout function disabled. 1: Receiver timeout function enabled.
22:20	Reserved	Must be kept at reset value.
19	MSBF	Most significant bit first. 0: Data is transmitted/received with the LSB first. 1: Data is transmitted/received with the MSB first. This bit field cannot be written when the USART is enabled (UEN=1).
18	DINV	Data bit level inversion. 0: Data bit signal values are not inverted. 1: Data bit signal values are inverted. This bit field cannot be written when the USART is enabled (UEN=1).
17	TINV	TX pin level inversion. 0: TX pin signal values are not inverted. 1: TX pin signal values are inverted. This bit field cannot be written when the USART is enabled (UEN=1).
16	RINV	RX pin level inversion. 0: RX pin signal values are not inverted. 1: RX pin signal values are inverted. This bit field cannot be written when the USART is enabled (UEN=1).
15	STRP	Swap TX/RX pins. 0: The TX and RX pins functions are not swapped. 1: The TX and RX pins functions are swapped. This bit field cannot be written when the USART is enabled (UEN=1).
14	LMEN	LIN mode enable. 0: LIN mode disabled. 1: LIN mode enabled. This bit field cannot be written when the USART is enabled (UEN=1).
13:12	STB[1:0]	STOP bits length. 00: 1 Stop bit. 01: 0.5 Stop bit. 10: 2 Stop bits. 11: 1.5 Stop bit. This bit field cannot be written when the USART is enabled (UEN=1).

11	CKEN	CK pin enable. 0: CK pin disabled. 1: CK pin enabled. This bit field cannot be written when the USART is enabled (UEN=1).
10	CPL	Clock polarity. 0: Steady low value on CK pin outside transmission window in synchronous mode. 1: Steady high value on CK pin outside transmission window in synchronous mode. This bit field cannot be written when the USART is enabled (UEN=1).
9	CPH	Clock phase. 0: The first clock transition is the first data capture edge in synchronous mode. 1: The second clock transition is the first data capture edge in synchronous mode. This bit field cannot be written when the USART is enabled (UEN=1).
8	CLEN	CK length. 0: The clock pulse of the last data bit (MSB) is not output to the CK pin in synchronous mode. 1: The clock pulse of the last data bit (MSB) is output to the CK pin in synchronous mode. This bit field cannot be written when the USART is enabled (UEN=1)
7	Reserved	Must be kept at reset value.
6	LBDIE	LIN break detection interrupt enable. 0: LIN break detection interrupt is disabled. 1: An interrupt will occur whenever the LBDF bit is set in USART_STAT.
5	LBLEN	LIN break frame length. 0: 10 bit break detection. 1: 11 bit break detection. This bit field cannot be written when the USART is enabled (UEN=1).
4	ADDM	Address detection mode. This bit is used to select between 4-bit address detection and full-bit address detection. 0: 4-bit address detection. 1: full-bit address detection. In 7-bit, 8-bit and 9-bit data modes, the address detection is done on 6-bit, 7-bit and 8-bit address (ADDR[5:0], ADDR[6:0] and ADDR[7:0]) respectively. This bit field cannot be written when the USART is enabled (UEN=1).
3:0	Reserved	Must be kept at reset value.

19.4.3. Control register 2 (USART_CTL2)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									WUIE	WUM[1:0]		SCRTNUM[2:0]			Reserved
									rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	WUIE	Wakeup from Deep-sleep mode interrupt enable. 0: Wakeup from Deep-sleep mode interrupt is disabled. 1: Wakeup from Deep-sleep mode interrupt is enabled.
21:20	WUM[1:0]	Wakeup mode from Deep-sleep mode. These bits are used to specify the event which activates the WUF (Wakeup from Deep-sleep mode flag) in the USART_STAT register. 00: WUF active on address or data match, which is defined by ADDR_DATA[7:0] bits and ADDM bit. 01: Reserved. 10: WUF active on Start bit. 11: WUF active on RBNE. This bit field cannot be written when the USART is enabled (UEN=1).
19:17	SCRTNUM[2:0]	Smartcard auto-retry number. In smartcard mode, these bits specify the number of retries in transmission and reception. In transmission mode, a transmission error (FERR bit set) will occur after this number of automatic retransmission retries. In reception mode, reception error (RBNE and PERR bits set) will occur after this number or erroneous reception trials. When these bits are configured as 0x0, there will be no automatic retransmission in transmit mode. This bit field is only can be cleared to 0 when the USART is enabled (UEN=1), to stop retransmission.
16	Reserved	Must be kept at reset value.
15	DEP	Driver enable polarity mode. 0: DE signal is active high. 1: DE signal is active low. This bit field cannot be written when the USART is enabled (UEN=1).
14	DEM	Driver enable mode. This bit is used to activate the external transceiver control, through the DE signal,

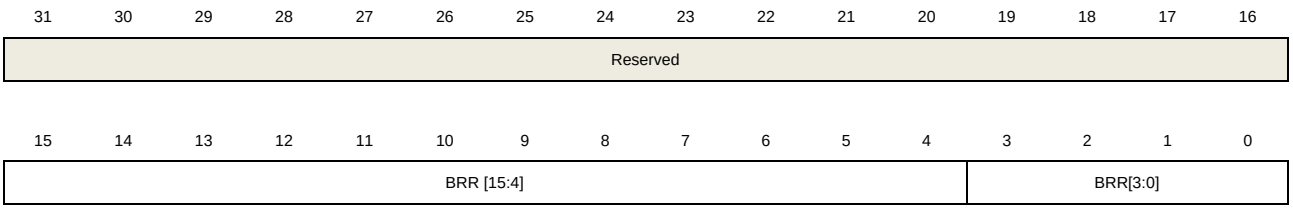
		which is output on the RTS pin. 0: DE function is disabled. 1: DE function is enabled. This bit field cannot be written when the USART is enabled (UEN=1).
13	DDRE	Mask DMA request on reception error. 0: DMA is not disabled in case of reception error. The DMA request is not asserted to make sure the erroneous data is not transferred, but the next correct received data will be transferred. The RBNE is kept 0 to prevent overrun when reception error, but the corresponding error flag is set. This mode can be used in Smartcard mode. 1: The DMA request is not asserted in case of reception error until the error flag is cleared. The RBNE flag and corresponding error flag will be set. The software must first disable the DMA request (DMAR = 0) or clear RBNE before clearing the error flag. This bit field cannot be written when the USART is enabled (UEN=1).
12	OVRD	Overrun disable. 0: Overrun functionality is enabled. The ORERR error flag will be set when received data is not read before receiving new data, and the new data will be lost. 1: Overrun functionality is disabled. The ORERR error flag will not be set when received data is not read before receiving new data, and the new received data overwrites the previous content of the USART_RDATA register. This bit field cannot be written when the USART is enabled (UEN=1).
11	OSB	One sample bit method. 0: Three sample bit method. 1: One sample bit method. This bit field cannot be written when the USART is enabled (UEN=1).
10	CTSIE	CTS interrupt enable. 0: CTS interrupt is disabled. 1: An interrupt will occur whenever the CTS bit is set in USART_STAT.
9	CTSEN	CTS enable. 0: CTS hardware flow control disabled. 1: CTS hardware flow control enabled. This bit field cannot be written when the USART is enabled (UEN=1).
8	RTSEN	RTS enable. 0: RTS hardware flow control disabled. 1: RTS hardware flow control enablNed, data can be requested only when there is space in the receive buffer. This bit field cannot be written when the USART is enabled (UEN=1).
7	DENT	DMA enable for transmission. 0: DMA mode is disabled for transmission.

		1: DMA mode is enabled for transmission.
6	DENR	DMA enable for reception. 0: DMA mode is disabled for reception. 1: DMA mode is enabled for reception.
5	SCEN	Smartcard mode enable. 0: Smartcard Mode disabled. 1: Smartcard Mode enabled. This bit field cannot be written when the USART is enabled (UEN=1).
4	NKEN	NACK enable in Smartcard mode. 0: Disable NACK transmission when parity error. 1: Enable NACK transmission when parity error. This bit field cannot be written when the USART is enabled (UEN=1).
3	HDEN	Half-duplex enable. 0: Half duplex mode is disabled. 1: Half duplex mode is enabled. This bit field cannot be written when the USART is enabled (UEN=1).
2	IRLP	IrDA low-power. 0: Normal mode. 1: Low-power mode. This bit field cannot be written when the USART is enabled (UEN=1).
1	IREN	IrDA mode enable. 0: IrDA disabled. 1: IrDA enabled. This bit field cannot be written when the USART is enabled (UEN=1).
0	ERRIE	Error interrupt enable. 0: Error interrupt disabled. 1: An interrupt will occur whenever the FERR bit or the ORERR bit or the NERR bit is set in USART_STAT in multibuffer communication.

19.4.4. Baud rate generator register (USART_BAUD)

Address offset: 0x0C
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).
This register cannot be written when the USART is enabled (UEN=1).



rw rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	BRR[15:4]	Integer of baud-rate divider. INTDIV = BRR[15:4].
3:0	BRR [3:0]	Fraction of baud-rate divider. If OVSMOD = 0, FRADIV = BRR[3:0]. If OVSMOD = 1, FRADIV = BRR[2:0], BRR[3] must be reset.

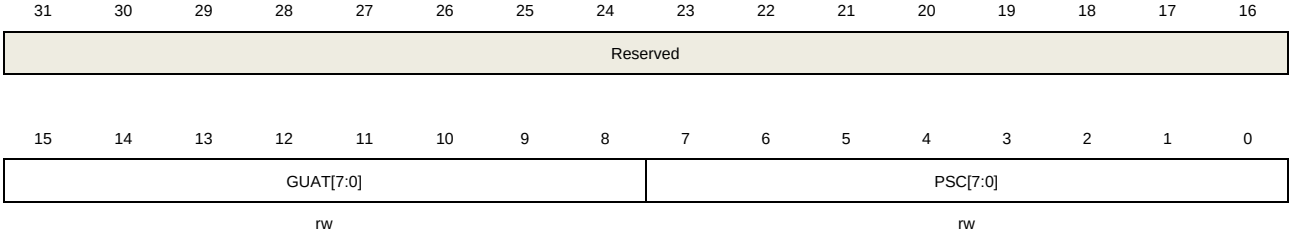
19.4.5. Prescaler and guard time configuration register (USART_GP)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This register cannot be written when the USART is enabled (UEN=1).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	GUAT[7:0]	Guard time value in smartcard mode. This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock. In IrDA Low-power mode, the division factor is the prescaler value. 00000000: Reserved - do not program this value. 00000001: divides the source clock by 1. 00000010: divides the source clock by 2. ... In IrDA normal mode. 00000001: can be set this value only In smartcard mode, the prescaler value for dividing the system clock is stored in PSC[4:0] bits. And the bits of PSC[7:5] must be kept at reset value. The division factor is twice as the prescaler value. 00000: Reserved - do not program this value. 00001: divides the source clock by 2. 00010: divides the source clock by 4.

00011: divides the source clock by 6.

...

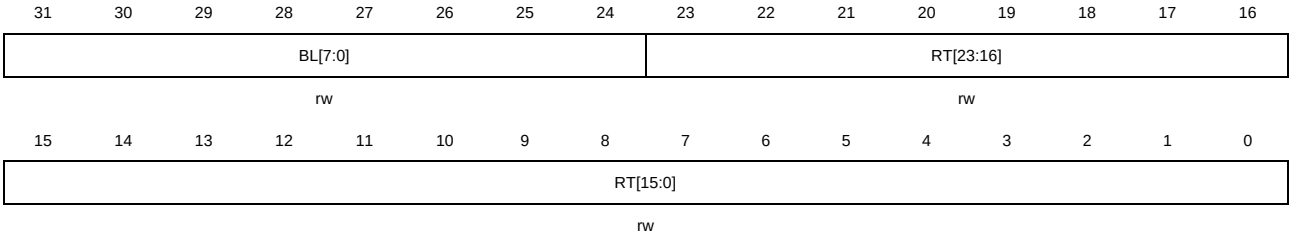
This bit field cannot be written when the USART is enabled (UEN=1).

19.4.6. Receiver timeout register (USART_RT)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).



Bits	Fields	Descriptions
31:24	BL[7:0]	Block Length These bits specify the block length in smartcard T=1 Reception. Its value equals the number of information characters + the length of the Epilogue Field (1-LEC/2-CRC) - 1. This value, which must be programmed only once per received block, can be programmed after the start of the block reception (using the data from the LEN character in the Prologue Field). The block length counter is reset when TBE=0 in smartcard mode. In other modes, when REN=0 (receiver disabled) and/or when the EBC bit is written to 1, the Block length counter is reset.
23:0	RT[23:0]	Receiver timeout threshold. These bits are used to specify receiver timeout value in terms of number of baud clocks. In standard mode, the RTF flag is set if no new start bit is detected for more than the RT value after the last received character. In smartcard mode, the CWT and BWT are implemented by this value. In this case, the timeout measurement is started from the start bit of the last received character. These bits can be written on the fly. The RTF flag will be set if the new value is lower than or equal to the counter. These bits must only be programmed once per received character.

19.4.7. Command register (USART_CMD)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TXFCMD	RXFCMD	MMCMD	SBKCMD	Reserved
											w	w	w	w	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TXFCMD	Transmit data flush request Writing 1 to this bit sets the TBE flag, to discard the transmit data.
3	RXFCMD	Receive data flush command. Writing 1 to this bit clears the RBNE flag to discard the received data without reading it.
2	MMCMD	Mute mode command Writing 1 to this bit makes the USART into mute mode and sets the RWU flag.
1	SBKCMD	Send break command. Writing 1 to this bit sets the SBKF flag and makes the USART send a BREAK frame, as soon as the transmit machine is idle.
0	Reserved	Must be kept at reset value.

19.4.8. Status register (USART_STAT)

Address offset: 0x1C

Reset value: 0x0000 00C0

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									REA	TEA	WUF	RWU	SBF	AMF	BSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	REA	Receive enable acknowledge flag. This bit, which is set/reset by hardware, reflects the receive enable state of the USART core logic. 0: The USART core receiving logic has not been enabled.

		1: The USART core receiving logic has been enabled.
21	TEA	Transmit enable acknowledge flag. This bit, which is set/reset by hardware, reflects the transmit enable state of the USART core logic. 0: The USART core transmitting logic has not been enabled. 1: The USART core transmitting logic has been enabled.
20	WUF	Wakeup from Deep-sleep mode flag. 0: No wakeup from Deep-sleep mode. 1: Wakeup from Deep-sleep mode. An interrupt is generated if WUFIE=1 in the USART_CTL2 register and the MCU is in Deep-sleep mode. This bit is set by hardware when a wakeup event, which is defined by the WUM bit field, is detected. Cleared by writing a 1 to the WUC in the USART_INTC register. This bit can also be cleared when UESM is cleared.
19	RWU	Receiver wakeup from mute mode. This bit is used to indicate if the USART is in mute mode. 0: Receiver in active mode. 1: Receiver in mute mode. It is cleared/set by hardware when a wakeup/mute sequence (address or IDLEIE) is recognized, which is selected by the WM bit in the USART_CTL0 register. This bit can only be set by writing 1 to the MMCMD bit in the USART_CMD register when wakeup on IDLEIE mode is selected.
18	SBF	Send break flag. 0: No break character is transmitted. 1: Break character will be transmitted. This bit indicates that a send break character was requested. Set by software, by writing 1 to the SBKCMD bit in the USART_CMD register. Cleared by hardware during the stop bit of break transmission.
17	AMF	ADDR match flag. 0: ADDR does not match the received character. 1: ADDR matches the received character, An interrupt is generated if AMIE=1 in the USART_CTL0 register. Set by hardware, when the character defined by ADDR [7:0] is received. Cleared by writing 1 to the AMC in the USART_INTC register.
16	BSY	Busy flag. 0: USART reception path is idle. 1: USART reception path is working.
15:13	Reserved	Must be kept at reset value.
12	EBF	End of block flag. 0: End of Block not reached.

		1: End of Block (number of characters) reached. An interrupt is generated if the EBIE=1 in the USART_CTL1 register. Set by hardware when the number of received bytes (from the start of the block, including the prologue) is equal or greater than BLEN + 4. Cleared by writing 1 to EBC bit in USART_INTC register.
11	RTF	Receiver timeout flag 0: Timeout value not reached 1: Timeout value reached without any data reception. An interrupt is generated if RTIE bit in the USART_CTL1 register is set. Set by hardware when the RT value, programmed in the USART_RT register has lapsed without any communication. Cleared by writing 1 to RTC bit in USART_INTC register. The timeout corresponds to the CWT or BWT timings in smartcard mode.
10	CTS	CTS level. This bit equals to the inverted level of the nCTS input pin. 0: nCTS input pin is in high level. 1: nCTS input pin is in low level.
9	CTSF	CTS change flag. 0: No change occurred on the nCTS status line. 1: A change occurred on the nCTS status line. An interrupt will occur if the CTSIE bit is set in USART_CTL2. Set by hardware when the nCTS input toggles. Cleared by writing 1 to CTSC bit in USART_INTC register.
8	LBDF	LIN break detected flag. 0: LIN Break is not detected. 1: LIN Break is detected. An interrupt will occur if the LBDIE bit is set in USART_CTL1. Set by hardware when the LIN break is detected. Cleared by writing 1 to LBDC bit in USART_INTC register.
7	TBE	Transmit data register empty. 0: Data is not transferred to the shift register. 1: Data is transferred to the shift register. An interrupt will occur if the TBEIE bit is set in USART_CTL0. Set by hardware when the content of the USART_TDATA register has been transferred into the transmit shift register or writing 1 to TXFCMD bit of the USART_CMD register. Cleared by a write to the USART_TDATA.
6	TC	Transmission completed. 0: Transmission is not completed. 1: Transmission is complete. An interrupt will occur if the TCIE bit is set in USART_CTL0.

		Set by hardware if the transmission of a frame containing data is completed and if the TBE bit is set. Cleared by writing 1 to TCC bit in USART_INTC register.
5	RBNE	Read data buffer not empty. 0: Data is not received. 1: Data is received and ready to be read. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. Set by hardware when the content of the receive shift register has been transferred to the USART_RDATA. Cleared by reading the USART_RDATA or writing 1 to RXFCMD bit of the USART_CMD register.
4	IDLEF	IDLE line detected flag. 0: No Idle Line is detected. 1: Idle Line is detected. An interrupt will occur if the IDLEIE bit is set in USART_CTL0. Set by hardware when an Idle Line is detected. It will not be set again until the RBNE bit has been set itself. Cleared by writing 1 to IDLEC bit in USART_INTC register.
3	ORERR	Overrun error. 0: No Overrun error is detected. 1: Overrun error is detected. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when the word in the receive shift register is ready to be transferred into the USART_RDATA register while the RBNE bit is set. Cleared by writing 1 to OREC bit in USART_INTC register.
2	NERR	Noise error flag. 0: No noise error is detected. 1: Noise error is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when noise error is detected on a received frame. Cleared by writing 1 to NEC bit in USART_INTC register.
1	FERR	Frame error flag. 0: No framing error is detected. 1: Frame error flag or break character is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when a de-synchronization, excessive noise or a break character is detected. This bit will be set when the maximum number of transmit attempts is reached without success (the card NACKs the data frame), when USART transmits in smartcard mode. Cleared by writing 1 to FEC bit in USART_INTC register.

0	PERR	Parity error flag. 0: No parity error is detected. 1: Parity error flag is detected. An interrupt will occur if the PERRIE bit is set in USART_CTL0. Set by hardware when a parity error occurs in receiver mode. Cleared by writing 1 to PEC bit in USART_INTC register.

19.4.9. Interrupt status clear register (USART_INTC)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											WUC	Reserved		AMC	Reserved
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EBC	RTC	Reserved	CTSC	LBDC	Reserved	TCC	Reserved	IDLEC	OREC	NEC	FEC	PEC
			w	w		w	w		w		w	w	w	w	w

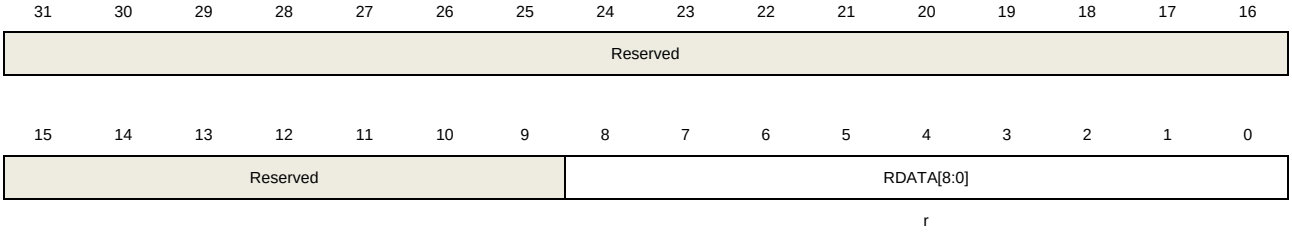
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	WUC	Wakeup from Deep-sleep mode clear. Writing 1 to this bit clears the WUF bit in the USART_STAT register.
19:18	Reserved	Must be kept at reset value.
17	AMC	ADDR match clear. Writing 1 to this bit clears the AMF bit in the USART_STAT register.
16:13	Reserved	Must be kept at reset value.
12	EBC	End of block clear. Writing 1 to this bit clears the EBF bit in the USART_STAT register.
11	RTC	Receiver timeout clear. Writing 1 to this bit clears the RTF flag in the USART_STAT register.
10	Reserved	Must be kept at reset value.
9	CTSC	CTS change clear. Writing 1 to this bit clears the CTSF bit in the USART_STAT register.
8	LBDC	LIN break detected clear. Writing 1 to this bit clears the LBDF flag in the USART_STAT register.
7	Reserved	Must be kept at reset value.
6	TCC	Transmission complete clear.

		Writing 1 to this bit clears the TC bit in the USART_STAT register.
5	Reserved	Must be kept at reset value.
4	IDLEC	Idle line detected clear. Writing 1 to this bit clears the IDLEF bit in the USART_STAT register.
3	OREC	Overrun error clear. Writing 1 to this bit clears the ORERR bit in the USART_STAT register.
2	NEC	Noise detected clear. Writing 1 to this bit clears the NERR bit in the USART_STAT register.
1	FEC	Frame error flag clear. Writing 1 to this bit clears the FERR bit in the USART_STAT register.
0	PEC	Parity error clear. Writing 1 to this bit clears the PERR bit in the USART_STAT register.

19.4.10. Receive data register (USART_RDATA)

Address offset: 0x24
Reset value: Undefined

This register has to be accessed by word (32-bit).

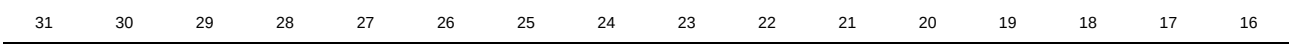


Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8:0	RDATA[8:0]	Receive Data value. The received data character is contained in these bits. The value read in the MSB (bit 7 or bit 8 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).

19.4.11. Transmit data register (USART_TDATA)

Address offset: 0x28
Reset value: Undefined

This register has to be accessed by word (32-bit).



Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TDATA[8:0]								
							rw								

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8:0	TDATA[8:0]	Transmit Data value. The transmit data character is contained in these bits. The value written in the MSB (bit 7 or bit 8 depending on the data length) will be replaced by the parity, when transmitting with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register). This register must be written only when TBE bit in USART_STAT register is set.

19.4.12. USART coherence control register (USART_CHC)

Address offset: 0xC0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EPERR	Reserved							HCM
							rc_w0								rw

Bits	Fields	Descriptions
31:9	Reserved	Forced by hardware to 0.
8	EPERR	Early parity error flag. This flag will be set as soon as the parity bit has been detected, which is before RBNE flag. This flag is cleared by writing 0. 0: No parity error is detected. 1: Parity error is detected.
7:1	Reserved	Forced by hardware to 0.
0	HCM	Hardware flow control coherence mode. 0: nRTS signal equals to the RBNE in status register. 1: nRTS signal is set when the last data bit (parity bit when pce is set) has been sampled.

19.4.13. USART receive FIFO control and status register (USART_RFCS)

Address offset: 0xD0

Reset value: 0x0000 0400

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFINT	RFCNT[2:0]			RFF	RFE	RFFIE	RFEN	Reserved							ELNACK
r_w0	r			r	r	rw	rw								rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	RFFINT	Receive FIFO full interrupt flag.
14:12	RFCNT[2:0]	Receive FIFO counter number.
11	RFF	Receive FIFO full flag. 0: Receive FIFO not full. 1: Receive FIFO full.
10	RFE	Receive FIFO empty flag. 0: Receive FIFO not empty. 1: Receive FIFO empty.
9	RFFIE	Receive FIFO full interrupt enable. 0: Receive FIFO full interrupt disable. 1: Receive FIFO full interrupt enable.
8	RFEN	Receive FIFO enable. This bit can be set when UESM = 1. 0: Receive FIFO disable. 1: Receive FIFO enable.
7:1	Reserved	Must be kept at reset value.
0	ELNACK	Early NACK when smartcard mode is selected. The NACK pulse occurs 1/16 bit time earlier when the parity error is detected. 0:Early NACKdisable when smartcard mode is selected. 1:Early NACKenable when smartcard mode is selected.

20. Inter-integrated circuit interface (I2C)

20.1. Overview

The I2C (inter-integrated circuit) module provides an I2C interface which is an industry standard two-line serial interface for MCU to communicate with external I2C interface. I2C bus uses two serial lines: a serial data line (SDA) and a serial clock line (SCL).

The I2C interface implements standard I2C protocol with standard mode, fast mode and fast mode plus as well as CRC calculation and checking, SMBus (system management bus), and PMBus (power management bus). It also supports multi-master I2C bus. The I2C interface provides DMA mode for users to reduce CPU overload.

20.2. Characteristics

- Parallel-bus to I2C-bus protocol converter and interface.
- Both master and slave functions with the same interface.
- Bi-directional data transfer between master and slave.
- Supports 7-bit and 10-bit addressing and general call addressing.
- Multiple 7-bit slave addresses (2 address with configurable mask).
- Programmable setup time and hold time.
- Multi-master capability.
- Supports standard mode (up to 100 kHz) and fast mode (up to 400 kHz) and fast mode plus (up to 1MHz).
- Configurable SCL stretching in slave mode.
- Supports DMA mode.
- SMBus 3.0 and PMBus 1.3 compatible.
- Optional PEC (packet error checking) generation and check.
- Programmable analog and digital noise filters.

20.3. Function overview

Figure 20-1. I2C module block diagram below provides details on the internal configuration of the I2C interface.

Figure 20-1. I2C module block diagram

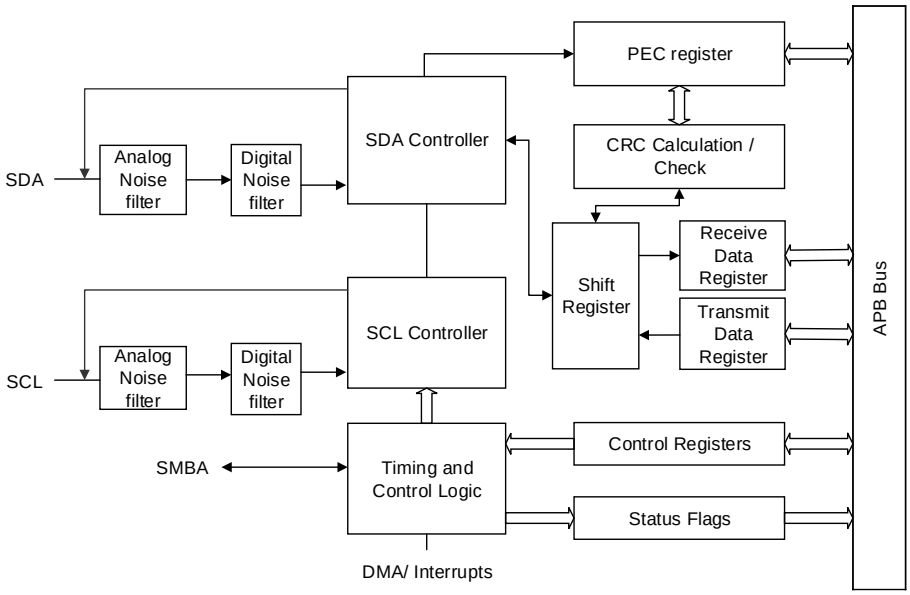


Table 20-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

Term	Description
Transmitter	the device which sends data to the bus
Receiver	the device which receives data from the bus
Master	the device which initiates a transfer, generates clock signals and terminates a transfer
Slave	the device addressed by a master
Multi-master	more than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	procedure to ensure that, if more than one master tries to control the bus simultaneously, only one is allowed to do so and the winning master's message is not corrupted

20.3.1. Clock requirements

The I2CCLK period t_{I2CCLK} must match the conditions as follows:

- $t_{I2CCLK} < (t_{LOW} - t_{filters})/4$
- $t_{I2CCLK} < t_{HIGH}$

with:

t_{LOW}: SCL low time

t_{HIGH}: SCL high time

$t_{filters}$: When the filters are enabled, represent the delays by the analog filter and digital filter.

Analog filter delay is maximum 260ns. Digital filter delay is DNF[3:0]×t_{I2CCLK}.

The period of PCLK clock t_{PCLK} match the conditions as follows:

- $t_{PCLK}<4/3*t_{SCL}$

with:

t_{SCL}: the period of SCL

Note: When the I2C kernel is provided by PCLK, this clock must match the conditions for t_{I2CCLK}.

20.3.2. I2C communication flow

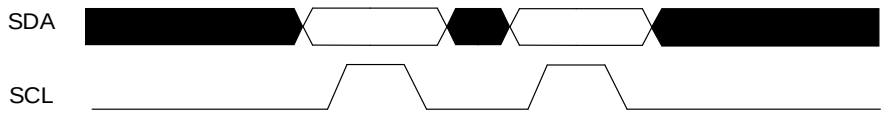
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

Data validation

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (see ***Figure 20-2. Data validation***). One clock pulse is generated for each data bit transferred.

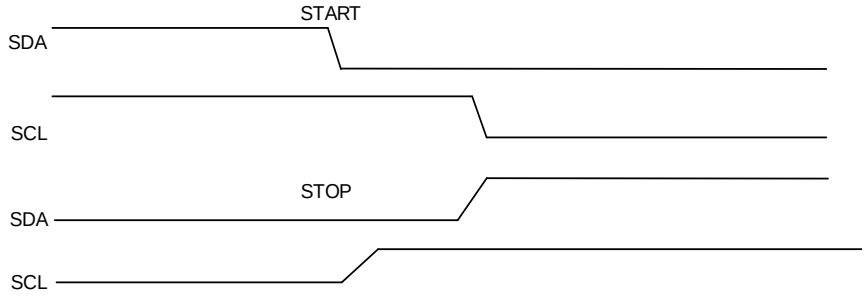
Figure 20-2. Data validation



START and STOP signal

All transactions begin with a START and are terminated by a STOP (see ***Figure 20-3. START and STOP signal***). A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START signal. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP signal.

Figure 20-3. START and STOP signal



Each I2C device is recognized by a unique address (whether it is a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. It operates in slave mode by default. When it generates a START signal, the interface automatically switches from slave to master. If an arbitration loss or a STOP generation occurs, then the interface switches from master to slave, allowing multimaster capability.

An I2C slave will continue to detect addresses after a START signalon I2C bus and compare the detected address with its slave address which is programmable by software. Once the two addresses match, the I2C slave will send an ACK to the I2C bus and responses to the following command on I2C bus: transmitting or receiving the desired data. Additionally, if General Call is enabled by software, the I2C slave always responses to a General Call Address (0x00). The I2C block support both 7-bit and 10-bit address modes.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the START signalcontain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

A 9th clock pulse follows the 8 clock cycles of byte transmission, during which the receiver must send an acknowledge bit to the transmitter. Acknowledge can be enabled or disabled by software.

An I2C master always initiates or end a transfer using START or STOP signaland it's also responsible for SCL clock generation.

In master mode, if AUTOEND=1, the STOP signalis generated automatically by hardware. If AUTOEND=0, the STOP signalgenerated by software, or the master can generate a RESTART signalto start a new transfer.

Figure 20-4. I2C communication flow with 10-bit address (Master Transmit)

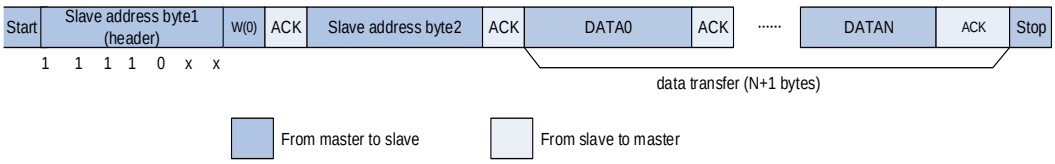


Figure 20-5. I2C communication flow with 7-bit address (Master Transmit)

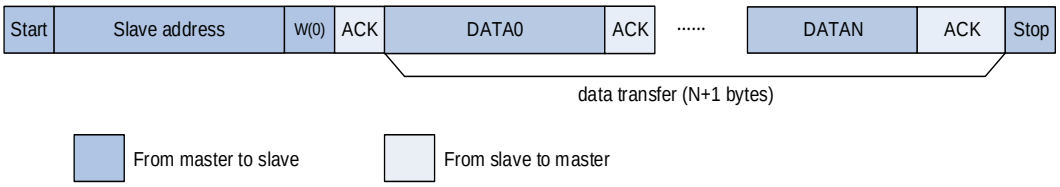
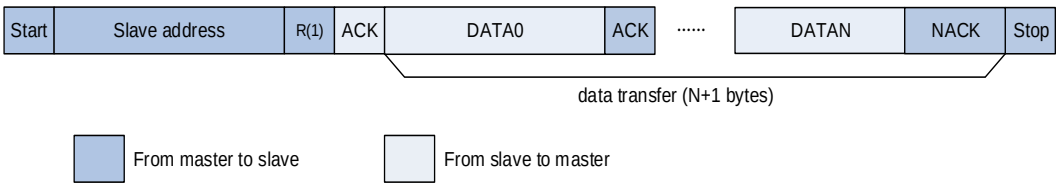


Figure 20-6. I2C communication flow with 7-bit address (Master Receive)



In 10-bit addressing mode, the HEAD10R bit can configured to decide whether the complete address sequence must be executed, or only the header to be sent. When HEAD10R=0, the complete 10 bit address read sequence must be excuted with START + header of 10-bit address in write direction + slave address byte 2 + RESTART + header of 10-bit address in read direction, as is shown in ***Figure 20-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)***.

In 10-bit addressing mode, if the master reception follows a master transmission between the same master and slave, the address read sequence can be RESTART + header of 10-bit address in read direction, as is shown in ***Figure 20-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)***.

Figure 20-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

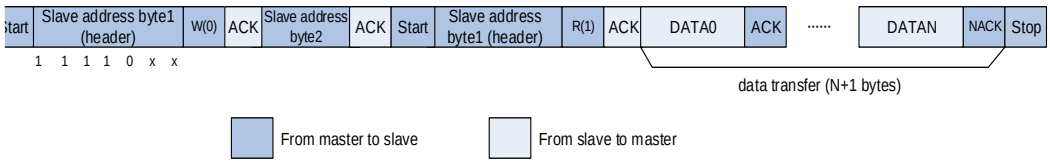
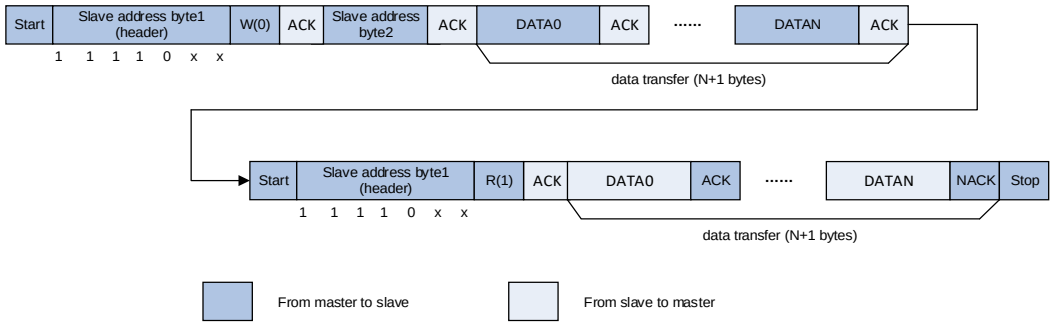


Figure 20-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



20.3.3. Noise filter

Analog noise filter and digital noise filter are integrated in I2C peripherals, the noise filters can

be configured before the I2C peripheral is enabled according to the actual requirements.

The analog noise filter is disabled by setting the ANOFF bit in I2C_CTL0 register and enabled when ANOFF is 0. It can suppress spikes with a pulse width up to 50ns in fast mode and fast mode plus.

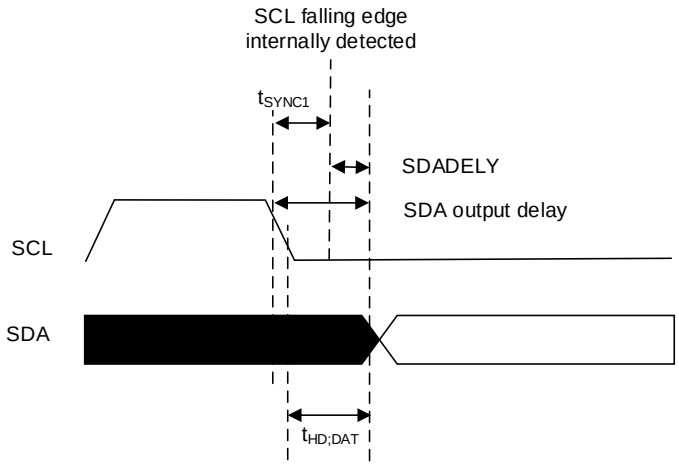
The digital noise filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will be changed if the level is stable for more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF[3:0].

20.3.4. I2C timings configuration

The PSC[3:0], SCLDELY[3:0] and SDADELY[3:0] bits in the I2C_TIMING register must be configured in order to guarantee a correct data hold and setup time used in I2C communication.

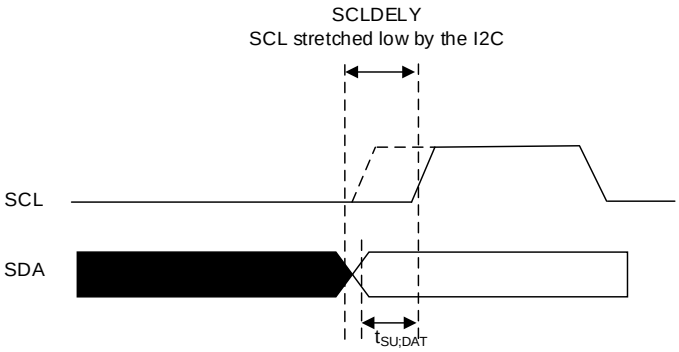
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 20-9. Data hold time***.

Figure 20-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in ***Figure 20-10. Data setup time***.

Figure 20-10. Data setup time



When the SCL falling edge is internally detected, a delay is inserted before sending SDA output. This delay is $t_{SDADELY}$ = SDADELY * t_{PSC} + t_{I2CCLK} where t_{PSC} = (PSC+1) * t_{I2CCLK}. $t_{SDADELY}$ effects $t_{HD;DAT}$. The total delay of SDA output is t_{SYNC1} + {[SDADELY * (PSC+1) + 1] * t_{I2CCLK}}. t_{SYNC1} depends on SCL falling slope, the delay of analog filter, the delay of digital filter and delay of SCL synchronization to I2CCLK clock. The delay of SCL synchronization to I2CCLK clock is 2 to 3 t_{I2CCLK}.

SDADELY must match condition as follows:

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

Note: t_{AF} is the delay of analog filter. The $t_{HD;DAT}$ should be less than the maximum of $t_{VD;DAT}$.

When SS = 0, after $t_{SDADELY}$ delay, the slave had to stretch the clock before the data writing to I2C_TDATA register, SCL is low during the data setup time. The setup time is $t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}. $t_{SCLDELY}$ effects $t_{SU;DAT}$.

SCLDELY must match condition as follows:

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

In master mode, the SCL clock high and low levels must be configured by programming the PSC[3:0], SCLH[7:0] and SCLL[7:0] bits in the I2C_TIMING register.

When the SCL falling edge is internally detected, a delay is inserted before releasing the SCL output. This delay is t_{SCLL}=(SCLL+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLL} impacts the SCL low time t_{LOW}.

When the SCL rising edge is internally detected, a delay is inserted before forcing the SCL output to low level. This delay is t_{SCLH}=(SCLH+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLH} impacts the SCL high time t_{HIGH}.

Note: When the I2C is enabled, the timing configuration and SS mode must not be changed.

Table 20-2. Data setup time and data hold time

Symbol	Parameter	Standard mode		Fast mode		Fast mode plus		SMBus		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{HD;DAT}$	Data hold time	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	Data valid time	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	Data setup time	250	-	100	-	50	-	250	-	ns
t_r	Rising time of SCL and SDA	-	1000	-	300	-	120	-	1000	
t_f	falling time of SCL and SDA	-	300	-	300	-	120	-	300	

20.3.5. I2C reset

A software reset can be performed by clearing the I2CEN bit in the I2C_CTL0 register. When a software reset is generated, the SCL and SDA are released. The communication control bits and status bits come back to the reset value. Software reset have no effect on configuration registers. The impacted register bits are START, STOP, NACKEN in I2C_CTL1 register, I2CBSY, TBE, TI, RBNE, ADDSEND, NACK, TCR, TC, STPDET, BERR, LOSTARB and OUERR in I2C_STAT register. Additionally, when the SMBus is supported, PECTRANS in I2C_CTL1 register, PECERR, TIMEOUT and SMBALT in I2C_STAT are also impacted.

In order to perform the software reset, I2CEN must be kept low during at least 3 APB clock cycles. This is ensured by writing software sequence as follows:

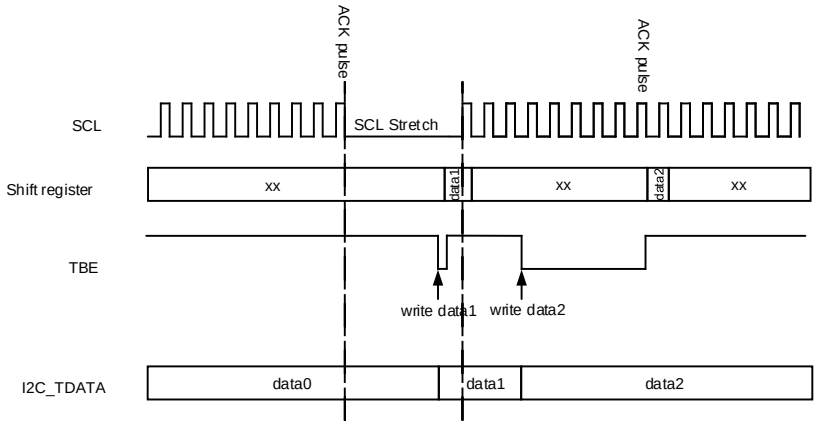
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

20.3.6. Data transfer

Data Transmission

When transmitting data, if TBE is 0, it indicates that the I2C_TDATA register is not empty, the data in I2C_TDATA register is moved to the shift register after the 9th SCL pulse. Then the data will be transmitted through the SDA line from the shift register. If TBE is 1, it indicates that the I2C_TDATA register is empty, the SCL line is stretched low until I2C_TDATA is not empty. The stretch begins after the 9th SCL pulse.

Figure 20-11. Data transmission

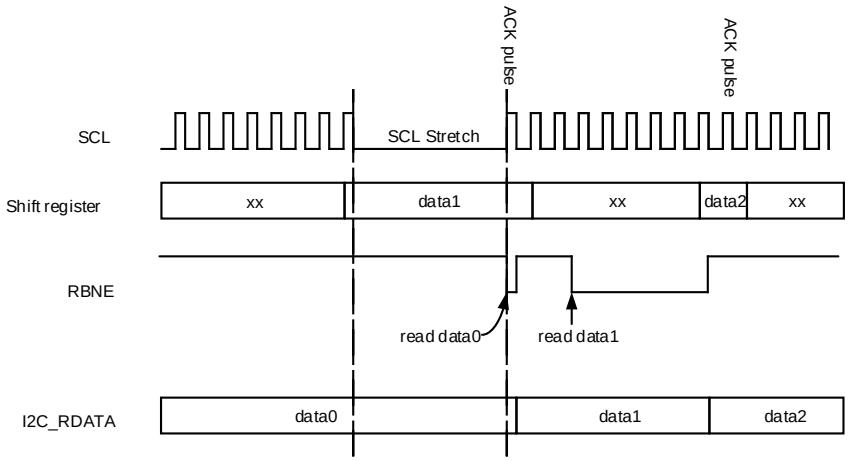


Data Reception

When receiving data, the data will be received in the shift register first. If RBNE is 0, the data in the shift register will move into I2C_RDATA register. If RBNE is 1, the SCL line wii be stretched until the previous received data in I2C_RDATA is read. The stretch is inserted before

the acknowledge pulse.

Figure 20-12. Data reception



Reload and automatic end mode

In order to manage byte transfer and to shut down the communication in modes as is shown in ***Table 20-3. Communication modes to be shut down***, the I2C embedded a byte counter in the hardware.

Table 20-3. Communication modes to be shut down

Working mode	Action
Master mode	NACK, STOP and RESTART generation
Slave receiver mode	ACK control
SMBus mode	PEC generation/checking

The number of bytes to be transferred is configured by BYTENUM[7:0] in I2C_CTL1 register. If BYTENUM is greater than 255, or in slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In reload mode, when BYTENUM counts to 0, the TCR bit will be set, and an interrupt will be generated if TCIE is set. Once the TCR flag is set, SCL is stretched. The TCR bit is cleared by writing a non-zero number in BYTENUM.

Note: The reload mode must be disabled after the last reloading of BYTENUM[7:0].

The reload mode must be disabled when the automatic end mode is enabled. In automatic end mode, the master will send a STOP signal automatically when the BYTENUM[7:0] counts to 0.

When reload mode and automatic end mode are disabled, the I2C communication process needs to be terminated by software. If the number of bytes in BYTENUM[7:0] has been transferred, the STOP bit should be set by software to generate a STOP signal, and then TC flag must be cleared.

20.3.7. I2C slave mode

Initialization

When works in slave mode, at least one slave address should be enabled. Slave address 1 can be programmed in I2C_SADDR0 register and slave address 2 can be programmed in I2C_SADDR1 register. ADDRESSEN in I2C_SADDR0 register and ADDRESS2EN in I2C_SADDR1 register should be set when the corresponding address is used. 7-bit address or 10-bit address can be programmed in ADDRESS[9:0] in I2C_SADDR0 register by configuring the ADDFORMAT bit in 7-bit address or 10-bit address.

The ADDM[6:0] in I2C_CTL2 register defines which bits of ADDRESS[7:1] are compared with an incoming address byte, and which bits are ignored.

The ADDMSK2[2:0] is used to mask ADDRESS2[7:1] in I2C_SADDR1 register. For details, refer to the description of ADDMSK2[2:0] in I2C_SADDR1 register.

When the I2C received address matches one of its enabled addresses, the ADDSEND will be set, and an interrupt is generated if the ADDMIE bit is set. The READDR[6:0] bits in I2C_STAT register will store the received address. And TR bit in I2C_STAT register updates after the ADDSEND is set. The bit will let the slave to know whether to act as a transmitter or receiver.

SCL line stretching

The clock stretching is used in slave mode by default (SS=0), the SCL line can be stretched low if necessary. The SCL will be stretched in following cases.

- The SCL is stretched when the ADDSEND bit is set, and released when the ADDSEND bit is cleared.
- In slave transmitting mode, after the ADDSEND bit is cleared, the SCL will be stretched before the first data byte writing to the I2C_TDATA register. Or the SCL will be stretched before the new data is written to the I2C_TDATA register after the previous data transmission is completed.
- In slave receiving mode, a new reception is completed but the data in I2C_RDATA register has not been read.
- When SBCTL=1 and RELOAD=1, after the transfer of the last byte, TCR is set. Before the TCR is cleared, the SCL will be stretched.
- The I2C stretches SCL low during $[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}$ after detecting the SCL falling edge.

The clock stretching can be disabled by setting the SS bit in I2C_CTL0 register (SS=1). The SCL will not be stretched in following cases.

- When the ADDSEND is set, the SCL will be not stretched.
- In slave transmitting mode, before the first SCL pulse, the data must be written in the I2C_TDATA register . Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, an interrupt will be generated. When the STPDET bit is set and the first

data transmission starts, OUERR bit in the I2C_STAT register will also be set.

- In slave receiving mode, before the 9th SCL pulse (ACK pulse) occurred by the next data byte, the data must be read out from the I2C_RDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, and an interrupt will be generated.

Slave byte control mode

In slave receiving mode, the slave byte control mode can be enabled by setting the SBCTL bit in the I2C_CTL0 register to allow byte ACK control. When SS=1, the slave byte control mode is not allowed.

When using slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In slave byte control mode, BYTENUM[7:0] in I2C_CTL1 register must be configured as 1 in the ADDSEND interrupt service routine and reloaded to 1 after each byte received. The TCR bit in I2C_STAT register will be set when a byte is received, the SCL will be stretched low by slave between the 8th and 9th clock pulses. Then the data can be read from the I2C_RDATA register, and the slave determines to send an ACK or a NACK by configuring the NACKEN bit in the I2C_CTL1 register. When the BYTENUM[7:0] is written a non-zero value, the slave will release the stretch.

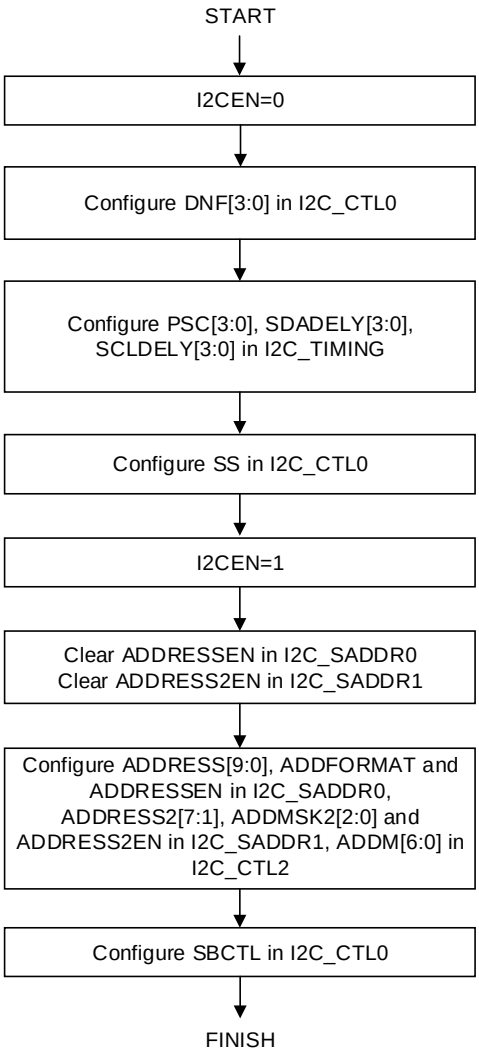
When the BYTENUM[7:0] is greater than 0x1, there is no stretch between the reception of two data bytes.

Note: The SBCTL bit can be configured in following cases:

1. I2CEN=0.
2. The slave has not been addressed.
3. ADDSEND=1.

Only when the ADDSEND=1, or TCR=1, the RELOAD bit can be modified.

Figure 20-13. I2C initialization in slave mode



Programming model in slave transmitting mode

When the I2C_TDATA register is empty, the TI bit in I2C_STAT register will be set. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The NACK bit in I2C_STAT register will be set when a NACK is received. And an interrupt is generated if the NACKIE bit is set in the I2C_CTL0 register. The TI bit in I2C_STAT register will not be set when a NACK is received.

The STPDET bit in I2C_STAT register will be set when a STOP is received. If the STPDETIE in I2C_CTL0 register is set, an interrupt will be generated.

When SBCTL is 0, if ADDSEND=1, and the TBE bit in I2C_STAT register is 0, the data in I2C_TDATA register can be chosed to be transmitted or flushed. The data is flushed by setting the TBE bit.

When SBCTL=1, the slave works in slave byte control mode, the BYTENUM[7:0] must be configured in the ADDSEND interrupt service routine. And the number of TI events is equal to the value of BYTENUM[7:0].

When SS=1, the SCL will not be stretched when ADDSEND bit in I2C_STAT register is set. In this case, the data in I2C_TDATA register can not be flushed in ADDSEND interrupt service routine. So the first byte to be sent must be programmed in the I2C_TDATA register previously.

- This data can be the one which is written in the last TI event of the last transfer.
- Setting the TBE bit can flush the data if it is not the one to be sent, then a new byte can be written in I2C_TDATA register. The STPDET must be 0 when the data transmission begins. Or else the OUERR bit in I2C_STAT register will be set and an underrun error occurs.
- When interrupt or DMA is used in slave transmitter, if a TI event is needed, in order to generate a TI event both the TI bit and the TBE bit must be set.

Figure 20-14. Programming model for slave transmitting when SS=0

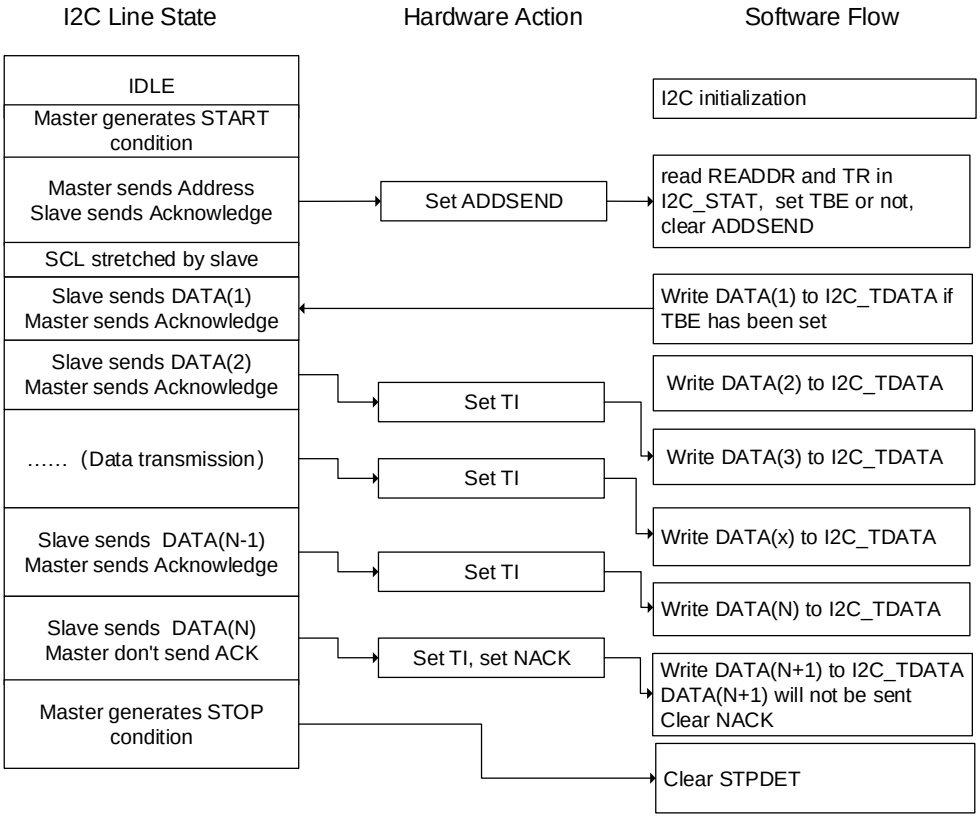
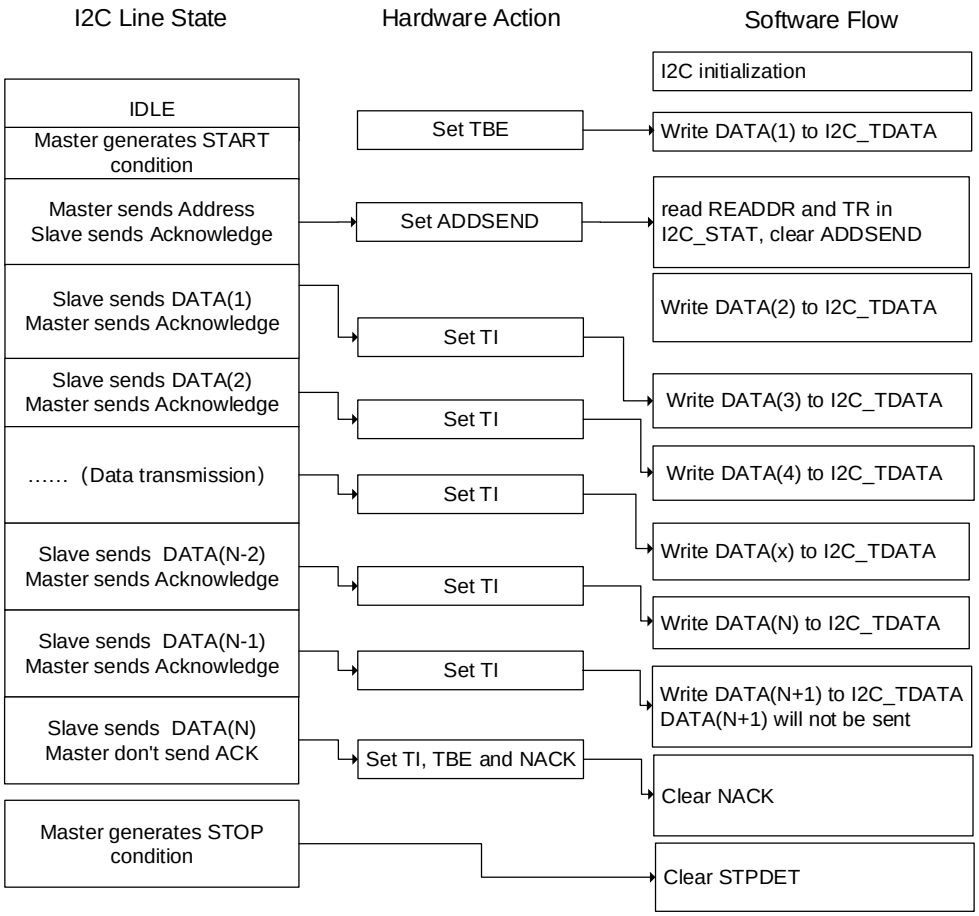


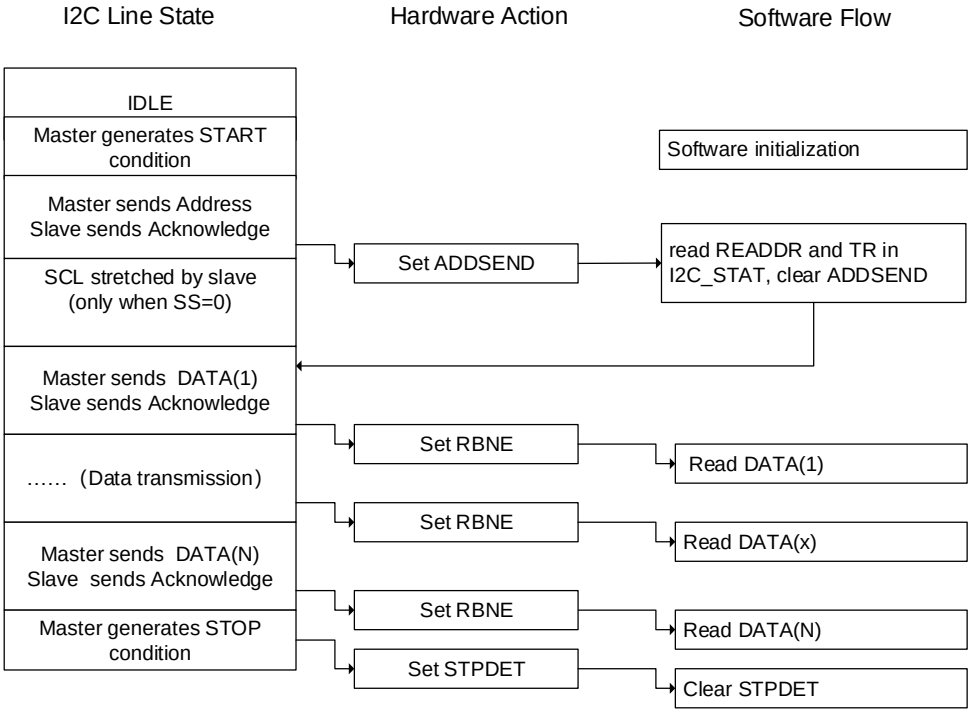
Figure 20-15. Programming model for slave transmitting when SS=1



Programming model in slave receiving mode

When the I2C_RDATA is not empty, the RBNE bit in I2C_STAT register is set, and if the RBNEIE bit in I2C_CTL0 register is set, an interrupt will be generated. When a STOP is received, STPDET will be set in I2C_STAT register. If the STPDETIE bit in I2C_CTL0 register is set, and an interrupt will be generated.

Figure 20-16. Programming model for slave receiving



20.3.8. I2C master mode

Initialization

The SCLH[7:0] and SCLL[7:0] in I2C_TIMING register should be configured when I2CEN is 0. In order to support multi-master communication and slave clock stretching, a clock synchronization mechanism is implemented.

The SCLL[7:0] and SCLH[7:0] are used for the low level counting and high level couting respectively. After a t_{SYNC1} delay, when the SCL low level is detected, the SCLL[7:0] starts counting, if the SCLL[7:0] in I2C_TIMING register is reached by SCLL[7:0] counter, the I2C will release the SCL clock. After a t_{SYNC2} delay, when the SCL high level is detected, the SCLH[7:0] starts counting, if the SCLH[7:0] in I2C_TIMING register is reached by SCLH[7:0] counter, the I2C will stretch the SCL clock.

So the master clock period is:

$$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}.$$

The t_{SYNC1} depends on the SCL falling slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The delay by digital noise filter is $DNF[3:0]*t_{I2CCLK}$.

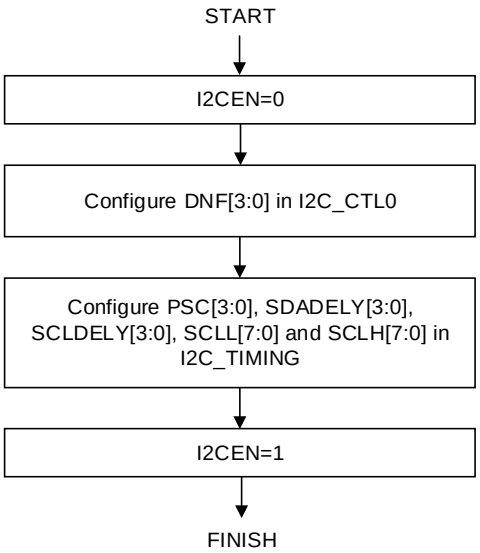
When works in master mode, the ADD10EN bit, SADDRESS[9:0] bits, TRDIR bit should be configured in I2C_CTL1 register. When the addressing mode is 10-bit in master receiving

mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to be sent. The number of bytes to be transferred should be configured in BYTENUM[7:0] in I2C_CTL1 register. If the number of bytes to be transferred is equal to or greater than 255, BYTENUM[7:0] should be configured as 0xFF. Then the master sends the START signal. All the bits above should be configured before the START is set. The slave address will be sent after the START signalwhen the I2CBSY bit I2C_STAT register is detected as 0. When the arbitration is lost, the master changes to slave mode and the START bit will be cleared by hardware. When the slave address has been sent, the START bit will be cleared by hardware.

In 10-bit addressing mode, if the master receives a NACK after the transmission of 10-bit header, the master will resend it until ACK is received. The ADDSENDC bit must be set to stop sending the slave address.

If the START bit is set, meanwhile the ADDSEND is set by addressing as a slave, the master changes to slave mode. The ADDSENDC bit must be set to clear the START bit.

Figure 20-17. I2C initialization in master mode



Programming model in master transmitting mode

In master transmitting mode, the TI bit is set after the ACK is received of each byte transmission. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The bytes to be transferred is programmed in BYTENUM[7:0] in I2C_CTL1 register. If the bytes to be transferred is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches unitil BYTENUM[7:0] is modified with a non-zero value.

When a NACK is received, the TI bit will not set.

- If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND

bit in I2C_CTL1 can be set to generate a STOP signalautomatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signalby setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signalto start a new transfer. The TC bit is cleared when the START / STOP bit is set.

- If a NACK is received, a STOP signalis automatically generated, the NACK is set in I2C_STAT register, if the NACKIE bit is set, an interrupt will be generated.

Note: When the RELOAD bit is 1, the AUTOEND has no effect.

Figure 20-18. Programming model for master transmitting (N<=255)

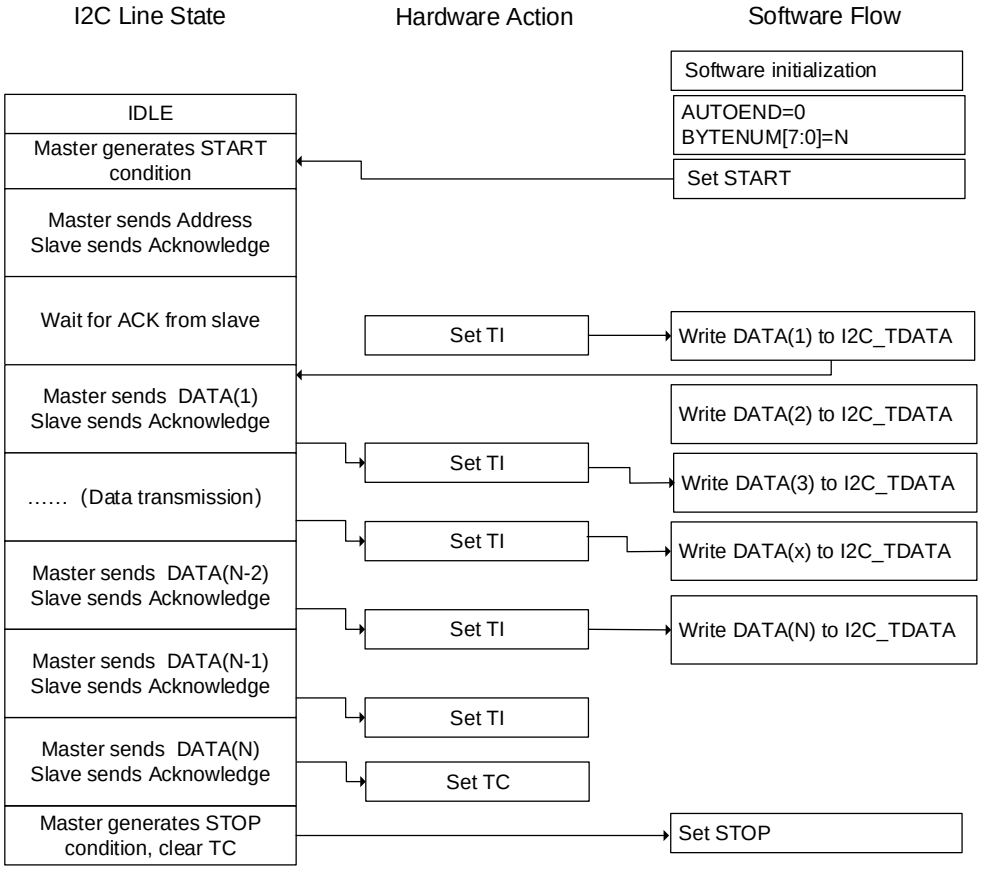
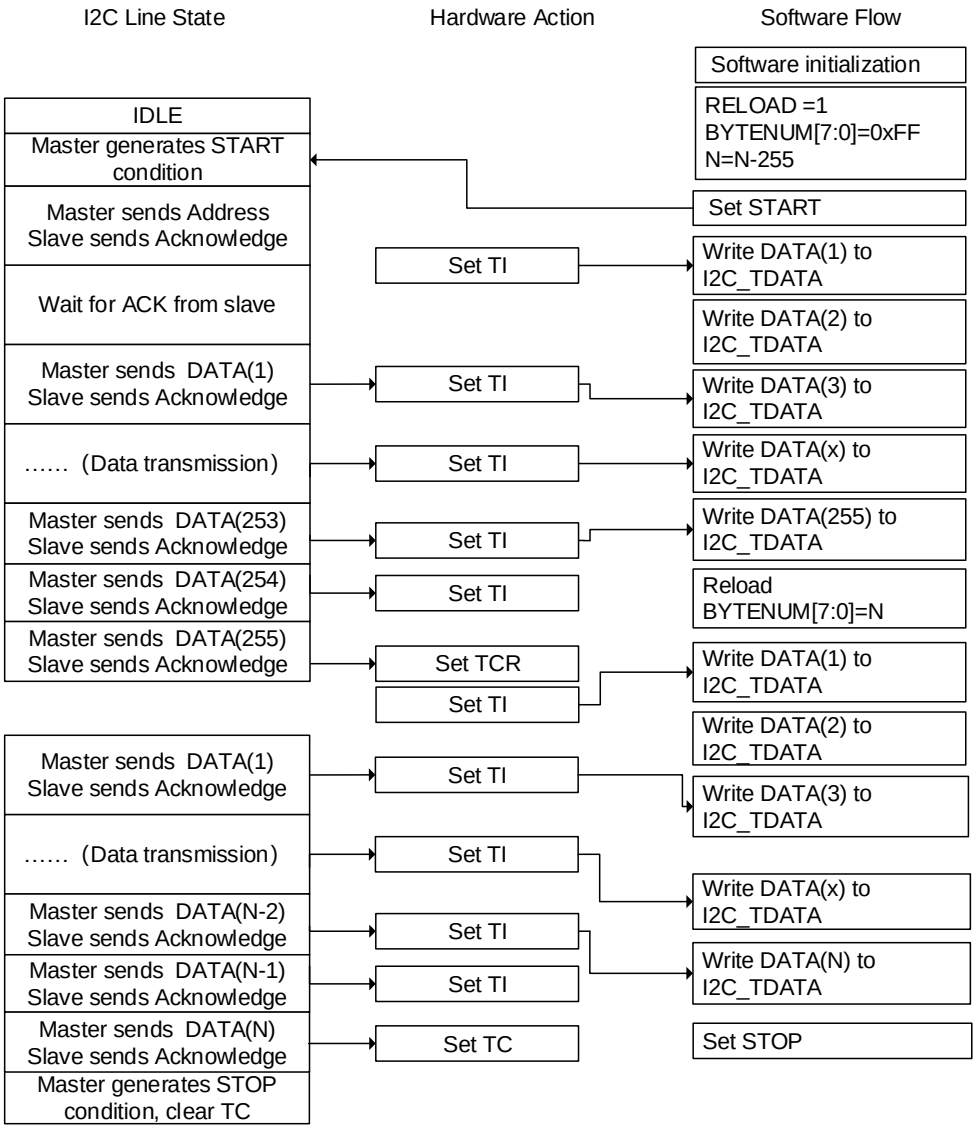


Figure 20-19. Programming model for master transmitting (N>255)



Programming model in master receiving mode

In master receiving mode, the RBNE bit in I2C_STAT register will be set when a byte is received. If the RBNEIE bit is set in I2C_CTL0 register, an interrupt will be generated. If the number of bytes to be received is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches unitil BYTENUM[7:0] is modified with a non-zero value.

If data of BYTENUM[7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signalautomatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signalby setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signalto start a new transfer. The TC bit is cleared when the START/STOP bit is set.

Figure 20-20. Programming model for master receiving (N<=255)

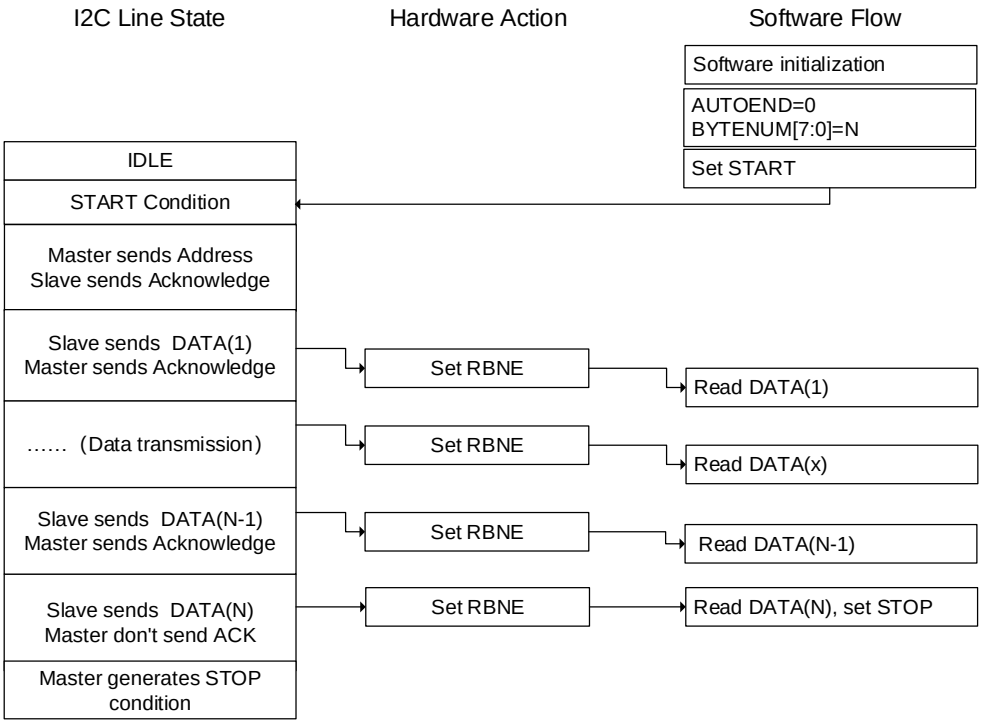
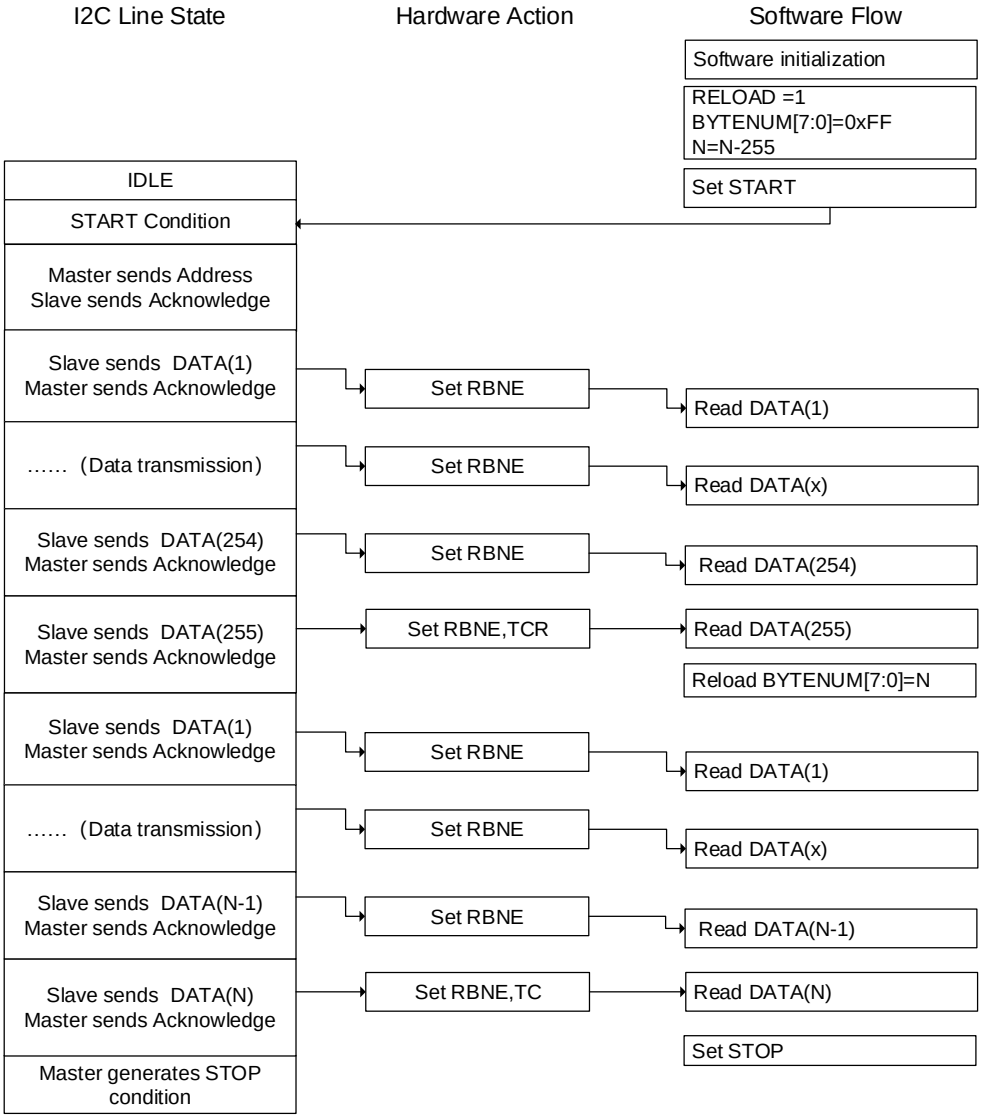


Figure 20-21. Programming model for master receiving (N>255)



20.3.9. SMBus support

The System Management Bus (abbreviated to SMBus or SMB) is a single-ended simple two-wire bus for the purpose of lightweight communication. Most commonly it is found in computer motherboards for communication with power source for ON/OFF instructions. It is derived from I2C for communication with low-bandwidth devices on a motherboard, especially power related chips such as a laptop's rechargeable battery subsystem (see Smart Battery Data).

SMBus protocol

Each message transaction on SMBus follows the format of one of the defined SMBus protocols. The SMBus protocols are a subset of the data transfer formats defined in the I2C specifications. I2C devices that can be accessed through one of the SMBus protocols are compatible with the SMBus specifications. I2C devices that do not adhere to these protocols cannot be accessed by standard methods as defined in the SMBus and Advanced Configuration and Power Management Interface (abbreviated to ACPI) specifications.

Address resolution protocol

The SMBus uses I2C hardware and I2C hardware addressing, but adds second-level software for building special systems. Additionally, its specifications include an Address Resolution Protocol that can make dynamic address allocations. Dynamic reconfiguration of the hardware and software allow bus devices to be 'hot-plugged' and used immediately, without restarting the system. The devices are recognized automatically and assigned unique addresses. This advantage results in a plug-and-play user interface. In this protocol there is a very useful distinction between a system host and all the other devices in the system, that is the host provides address assignment function.

SMBus slave byte control

The slave byte control of SMBus receiver is the same as I2C. It allows the ACK control of each byte.The Slave Byte Control mode must be enabled by setting SBCTL bit in I2C_CTL0 register.

Host Notify protocol

When the SMBHAEN bit in the I2C_CTL0 register is set, the SMBus supports the host notify protocol. In this protocol, the device acts as a master and the host as a slave, and the host will acknowledge the SMBus host address.

Time-out feature

SMBus has a time-out feature which resets devices if a communication takes too long. This explains the minimum clock frequency of 10 kHz to prevent locking up the bus. I2C can be a 'DC' bus, meaning that a slave device stretches the master clock when performing some routine while the master is accessing it. This will notify to the master that the slave is busy but does not want to lose the communication. The slave device will allow continuation after its task is completed. There is no limit in the I2C bus protocol as to how long this delay can be, whereas for a SMBus system, it would be limited to 25~35ms. SMBus protocol just assumes that if something takes too long, then it means that there is a problem on the bus and that all devices must reset in order to clear this mode. Slave devices are not allowed to hold the clock low too long.

The timeout detection can be enabled by setting TOEN and EXTOEN bits in the I2C_TIMEOUT register. The timer must be configured to guarantee that the timeout detected before the maximum time given in the SMBus specification.

The value programmed in BUSTOA[11:0] is used to check the $t_{TIMEOUT}$ parameter. To detect the SCL low level timeout, the TOIDLE bit must be 0. And the timer can be enabled by setting the TOEN bit in the I2C_TIMEOUT register, after the TOEN bit is set, the BUSTOA[11:0] and the TOIDLE bit cannot be changed. If the low level time of SCL is greater than (BUSTOA+1)*2048*t_{I2CCLK}, the TIMEOUT flag will be set in I2C_STAT register.

The BUSTOB[11:0] is used to check the $t_{LOW:SEXT}$ of the slave and the $t_{LOW:MEXT}$ of the

master. The timer can be enabled by setting the EXTOEN bit in the I2C_TIMEOUT register, after the EXTOEN bit is set, the BUSTOB[11:0] cannot be changed. If the SCL stretching time of the SMBus peripheral is greater than (BUSTOB+1)*2048*t_{I2CCLK} and within the timeout interval described in the bus idle detection section, the TIMEOUT bit in the I2C_STAT register will be set.

Packet error checking

There is a CRC-8 calculator in I2C block to perform Packet Error Checking for I2C data. A PEC (packet error code) byte is appended at the end of each transfer. The byte is calculated as CRC-8 checksum, calculated over the entire message including the address and read/write bit. The polynomial used is x^8+x^2+x+1 (the CRC-8-ATM HEC algorithm, initialized to zero).

When I2C is disabled, the PEC can be enabled by setting the PECEN bit in I2C_CTL0 register.Since the PEC transmission is managed by BYTENUM[7:0] in I2C_CTL1 register, SBCTL bit must be set when act as a slave. When PECTRANS is set and the RELOAD bit is cleared, PEC is transmitted after the BYTENUM[7:0]-1 data byte. The PECTRANS has no effect if RELOAD is set.

Table 20-4. SMBus with PEC configuration

Mode	SBCTL bit	RELOAD bit	AUTOEND bit	PECTRANS bit
Master Tx/Rx BYTENUM + PEC+ STOP	x	0	1	1
Master Tx/Rx BYTENUM + PEC + RESTART	x	0	0	1
Slave Tx/Rx with PEC	1	0	x	1

SMBus alert

The SMBus has an extra optional shared interrupt signal called SMBALERT# which can be used by slaves to tell the host to ask its slaves about events of interest. The host processes the interrupt and accesses all SMBALERT# devices through the Alert Response Address at the same time. If the SMBALERT# is pulled low by the devices, the devices will acknowledge the Alert Response Address. When SMBHAEN is 0, it is configured as a slave device, the SMBA pin will be pulled low by setting the SMBALTEN bit in the I2C_CTL0 register. Meanwhile the Alert Response Address is enabled. When SMBHAEN is 1, it is configured as a host, and the SMBALTEN is 1, as soon as a falling edge is detected on the SMBA pin, the SMBALT flag will be set in the I2C_STAT register. If the ERRIE bit is set in the I2C_CTL0 register, an interrupt will be generated. When SMBALTEN is 0, the level of ALERT line is considered high even if the SMBA pin is low. The SMBA pin can be used as a standard GPIO if SMBALTEN is 0.

Bus idle detection

If the master detects that the high level duration of the clock and data signals is greater than $t_{HIGH,MAX}$, the bus can be considered idle.

This timing parameter includes the case of a master that has been dynamically added to the bus and may not have detected a state transition on a SMBCLK or SMBDAT lines. In this case, in order to ensure that there is no ongoing transmission, the master must wait long enough.

The BUSTOA[11:0] bits must be programmed with the timer reload value to enable the t_{IDLE} check in order to obtain the t_{IDLE} parameter. To detect SCL and SDA high level timeouts, the TOIDLE bit must be set. Then setting the TOEN bit in the I2C_TIMEOUT register to enable the timer, after the TOEN bit is set, the BUSTOA[11:0] bit and the TOIDLE bit cannot be changed. If the high level time of both SCL and SDA is greater than (BUSTOA+1)*4*t_{I2CCLK}, the TIMEOUT flag will be set in the I2C_STAT register.

SMBus slave mode

The SMBus receiver must be able to NACK each command or data it receives. For ACK control in slave mode, slave byte control mode can be enabled by setting SBCTL bit in I2C_CTL0 register.

SMBus-specific addresses should be enabled when needed. The SMBus Device Default address (0b1100 001) is enabled by setting the SMBDAEN bit in the I2C_CTL0 register. The SMBus Host address (0b0001 000) is enabled by setting the SMBHAEN bit in the I2C_CTL0 register. The Alert Response Address (0b0001 100) is enabled by setting the SMBALTEN bit in the I2C_CTL0 register.

20.3.10. SMBus mode

SMBus Master Transmitter and Slave Receiver

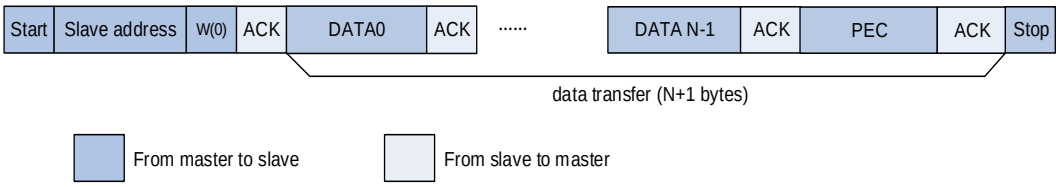
The PEC in SMBus master mode can be transmitted by setting the PECTRANS bit before setting the START bit, and the number of bytes in the BYTENUM[7:0] field must be configured. In this case, the total number of transmissions when TI interrupt occur is BYTENUM-1. So if BYTENUM=0x1 and PECTRANS bit is set at the same time, the contents of the I2C_PEC register are automatically transferred. If the automatic end mode is selected (AUTOEND=1), the SMBus master automatically sends the STOP signalafter the PEC byte. If the automatic end mode is not selected (AUTOEND=0), the SMBus master can send a RESTART signalafter the PEC. The I2C_PEC register content will be sent after BYTENUM-1 bytes, and the TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART must be set in the TC interrupt routine.

When used as slave receiver, in order to allow PEC checking at the end of the number of bytes transmitted, SBCTL must be set. To configure ack control for each byte, the RELOAD must be set. In order to check the PEC byte, it is necessary to clear the RELOAD bit and set PECTRANS bit. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register. If the PEC values does not match, the NACK is automatically generated. If the PEC values matches, the ACK is automatically generated, regardless of the NACKEN bit value. When PEC byte is received, it is also copied into the

I2C_RDATA register, and RBNE flag will be set. If the ERRIE bit in I2C_CTL0 register is 1, when PEC value does not match, the PECERR flag will be set and the interrupt will be generated. If ACK control is not required, then PECTRANS can be set to 1 and BYTENUM can be programmed according to the number of bytes to be received.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 20-22. SMBus Master Transmitter and Slave Receiver communication flow



SMBus Master Receiver and Slave Transmitter

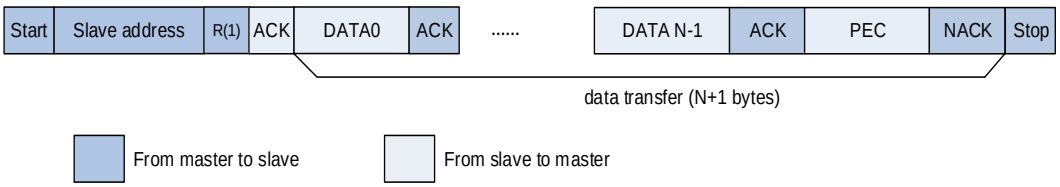
If the SMBus master is required to receive PEC at the end of bytes transfer, automatic end mode can be enabled. Before sending a START signal on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. A NACK is respond to the PEC byte before STOP signal.

If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, the software mode (AUTOEND = 0) must be selected. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the contents of the I2C_PEC register automatically. The TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART can be set in the TC interrupt routine.

When used as slave transmitter, in order to allow PEC transmission at the end of BYTENUM[7:0] bytes, SBCTL must be set. If PECTRANS bit is set, the number of bytes in BYTENUM[7:0] contains PEC byte. In this case, if the number of bytes requested by the master is greater than BYTENUM-1, the total number of TI interrupts will be BYTENUM-1, and the contents of the I2C_PEC register will be transmitted automatically.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 20-23. SMBus Master Receiver and Slave Transmitter communication flow



20.3.11. Use DMA for data transfer

As is shown in I2C slave mode and I2C master mode, each time TI or RBNE is asserted, software should write or read a byte, this may cause CPU's high overload. The DMA controller

can be used to process TI and RBNE flag: each time TI or RBNE is asserted, DMA controller does a read or write operation automatically.

The DMA transmission request is enabled by setting the DENT bit in I2C_CTL0 register. The DMA reception request is enabled by setting the DENR bit in I2C_CTL0 register. In master mode, the slave address, transmission direction, number of bytes and START bit are programmed by software. The DMA must be initialized before setting the START bit. The number of bytes to be transferred is configured in the BYTENUM[7:0] in I2C_CTL1 register. In slave mode, the DMA must be initialized before the address match event or in the ADDSEND interrupt routine, before clearing the ADDSEND flag.

20.3.12. I2C error and interrupts

The I2C error flags are listed in ***Table 20-5. I2C error flags***.

Table 20-5. I2C error flags

I2C Error Name	Description
BERR	Bus error
LOSTARB	Arbitration lost
OUERR	Overrun/Underrun flag
PECERR	CRC value doesn't match
TIMEOUT	Bus timeout in SMBus mode
SMBALT	SMBus Alert

The I2C interrupt events and flags are listed in ***Table 20-6. I2C interrupt events***.

Table 20-6. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP signaldetected in slave mode	STPDET	STPDETIE
Transfer complete reload	TCR	TCIE
Transfer complete	TC	
Address match	ADDSEND	ADDMIE
Not acknowledge received	NACK	NACKIE
Bus error	BERR	ERRIE
Arbitration Lost	LOSTARB	
Overrun/Underrun error	OUERR	
PEC error	PECERR	
Timeout error	TIMEOUT	
SMBus Alert	SMBALT	

20.3.13. I2C debug mode

When the microcontroller enters the debug mode (Cortex®-M33 core halted), the SMBus timeout either continues to work normally or stops, depending on the I2Cx_HOLD

configuration bits in the DBG module.

20.4. Register definition

I2C0 base address: 0x4000 5400

I2C1 base address: 0x4000 5800

20.4.1. Control register 0 (I2C_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PECEN	SMBALTEN	SMBDAEN	SMBHAEN	GCEN	Reserved	SS	SBCTL
								rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	Reserved	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETIE	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	PECEN	PEC Calculation Switch 0: PEC Calculation off 1: PEC Calculation on
22	SMBALTEN	SMBus Alert enable 0: SMBA pin is not pulled down (device mode) or SMBus Alert pin SMBA is disabled (host mode) 1: SMBA pin is pulled down (device mode) or SMBus Alert pin SMBA is enabled (host mode)
21	SMBDAEN	SMBus device default address enable 0: Device default address is disabled, the default address 0b1100001x will be not acknowledged. 1: Device default address is enabled, the default address 0b1100001x will be acknowledged.
20	SMBHAEN	SMBus host address enable 0: Host address is disabled, address 0b0001000x will be not acknowledged. 1: Host address is enabled, address 0b0001000x will be acknowledged.
19	GCEN	Whether or not to response to a General Call (0x00) 0: Slave won't response to a General Call 1: Slave will response to a General Call

18	Reserved	Must be kept at reset value.
17	SS	Whether to stretch SCL low when data is not ready in slave mode. This bit is set and cleared by software. 0: SCL Stretching is enabled 1: SCL Stretching is disabled **Note:** When in master mode, this bit must be 0. This bit can be modified when I2CEN = 0.
16	SBCTL	Slave byte control This bit is used to enable hardware byte control in slave mode. 0: Slave byte control is disabled 1: Slave byte control is enabled
15	DENR	DMA enable for reception 0: DMA is disabled for reception 1: DMA is enabled for reception
14	DENT	DMA enable for transmission 0: DMA is disabled for transmission 1: DMA is enabled for transmission
13	Reserved	Must be kept at reset value.
12	ANOFF	Analog noise filter disable 0: Analog noise filter is enabled 1: Analog noise filter is disabled **Note:** This bit can only be programmed when the I2C is disabled (I2CEN = 0).
11:8	DNF[3:0]	Digital noise filter These bits are used to configure the digital noise filter on SDA and SCL input. The digital filter will filter spikes with a length of up to DNF[3:0]*t_{I2CCLK} 0000: Digital filter is disabled 0001: Digital filter is enabled and filter spikes with a length of up to 1 t_{I2CCLK} ... 1111: Digital filter is enabled and filter spikes with a length of up to15 t_{I2CCLK} These bits can only be modified when the I2C is disabled (I2CEN = 0).
7	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled. When BERR, LOSTARB, OUERR, PECERR, TIMEOUT or SMBALT bit is set, an interrupt will be generated.
6	TCIE	Transfer complete interrupt enable 0: Transfer complete interrupt is disabled 1: Transfer complete interrupt is enabled
5	STPDETIE	Stop detection interrupt enable 0: Stop detection (STPDET) interrupt is disabled

		1: Stop detection (STPDET) interrupt is enabled
4	NACKIE	Not acknowledge received interrupt enable 0: NACK received interrupt is disabled 1: NACK received interrupt is enabled
3	ADDMIE	Address match interrupt enable in slave mode 0: Address match interrupt is disabled 1: Address match interrupt is enabled
2	RBNEIE	Receive interrupt enable 0: Receive (RBNE) interrupt is disabled 1: Receive (RBNE) interrupt is enabled
1	TIE	Transmit interrupt enable 0: Transmit (TI) interrupt is disabled 1: Transmit (TI) interrupt is enabled
0	I2CEN	I2C peripheral enable 0: I2C is disabled 1: I2C is enabled

20.4.2. Control register 1 (I2C_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PECTRANS	PEC Transfer Set by software. Cleared by hardware in the following cases: When PEC byte is transferred or ADDSEND bit is set or STOP signal is detected or I2CEN=0. 0: Don’t transfer PEC value 1: Transfer PEC

		Note: This bit has no effect when RELOAD=1, or SBCTL=0 in slave mode.
25	AUTOEND	Automatic end mode in master mode 0: TC bit is set when the transfer of BYTENUM[7:0] bytes is completed. 1: a STOP signal is sent automatically when the transfer of BYTENUM[7:0] bytes is completed. **Note:** This bit works only when RELOAD=0. This bit is set and cleared by software.
24	RELOAD	Reload mode 0: After the data of BYTENUM[7:0] bytes transfer, the transfer is completed. 1: After data of BYTENUM[7:0] bytes transfer, the transfer is not completed and the new BYTENUM[7:0] will be reloaded. Every time when the BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set. This bit is set and cleared by software.
23:16	BYTENUM[7:0]	Number of bytes to be transferred These bits are programmed with the number of bytes to be transferred. When SBCTL=0, these bits have no effect. **Note:** These bits should not be modified when the START bit is set.
15	NACKEN	Generate NACK in slave mode 0: an ACK is sent after receiving a new byte. 1: a NACK is sent after receiving a new byte. **Note:** The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP signal is detected or ADDSEND is set, or when I2CEN=0. When PEC is enabled, whether to send an ACK or a NACK is not depend on the NACKEN bit. When SS=1, and the OUERR bit is set, the value of NACKEN is ignored and a NACK will be sent.
14	STOP	Generate a STOP signal on I2C bus This bit is set by software and cleared by hardware when I2CEN=0 or STOP signal is detected. 0: STOP will not be sent 1: STOP will be sent
13	START	Generate a START signal on I2C bus This bit is set by software and cleared by hardware after the address is sent. When the arbitration is lost, or a timeout error occurred, or I2CEN=0, this bit can also be cleared by hardware. It can be cleared by software by setting the ADDSENDC bit in I2C_STATC register. 0: START will not be sent 1: START will be sent
12	HEAD10R	10-bit address header executes read direction only in master receive mode 0: The 10 bit master receive address sequence is START + header of 10-bit address (write) + slave address byte 2 + RESTART + header of 10-bit address (read). 1: The 10 bit master receive address sequence is RESTART + header of 10-bit address (read).

		Note: When the START bit is set, this bit can not be changed.
11	ADD10EN	10-bit addressing mode enable in master mode 0: 7-bit addressing in master mode 1: 10-bit addressing in master mode **Note:** When the START bit is set, this bit can not be modified.
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive **Note:** When the START bit is set, this bit can not be modified.
9:0	SADDRESS[9:0]	Slave address to be sent SADDRESS[9:8]: Slave address bit 9:8 If ADD10EN = 0, these bits have no effect. If ADD10EN = 1, these bits should be written with bits 9:8 of the slave address to be sent. SADDRESS[7:1]: Slave address bit 7:1 If ADD10EN = 0, these bits should be written with the 7-bit slave address to be sent. If ADD10EN = 1, these bits should be written with bits 7:1 of the slave address to be sent. SADDRESS0: Slave address bit 0 If ADD10EN = 0, this bit has no effect. If ADD10EN = 1, this bit should be written with bit 0 of the slave address to be sent **Note:** When the START bit is set, the bit filed can not be modified.

20.4.3. Slave address register 0 (I2C_SADDR0)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESSEN	Reserved				ADDFORMAT	ADDRESS[1:0]		ADDRESS[6:0]							ADDRESS0
rw					rw	rw		rw							rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESSEN	I2C address enable 0: I2C address disable. 1: I2C address enable.

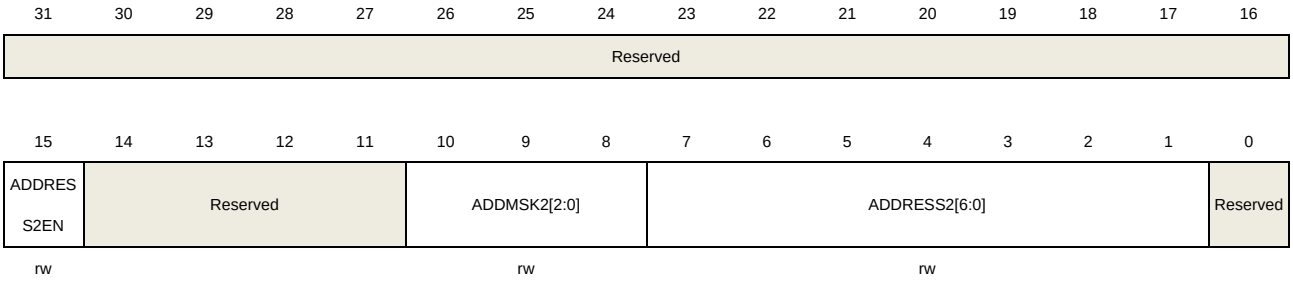
14:11	Reserved	Must be kept at reset value.
10	ADDFORMAT	Address mode for the I2C slave 0: 7-bit address 1: 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.
9:8	ADDRESS[1:0]	Highest two bits of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.
7:1	ADDRESS[6:0]	7-bit address or bits 7:1 of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.
0	ADDRESS0	Bit 0 of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.

20.4.4. Slave address register 1 (I2C_SADDR1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESS2EN	Second I2C address enable 0: Second I2C address disable. 1: Second I2C address enable.
14:11	Reserved	Must be kept at reset value.
10:8	ADDMSK2[2:0]	ADDRESS2[7:1] mask Defines which bits of ADDRESS2[7:1] are compared with an incoming address byte, and which bits are masked (don't care). 000: No mask, all the bits must be compared. n(001~110): ADDRESS2[n:1] is masked. Only ADDRESS2[7:n+1] are compared. 111: ADDRESS2[7:1] are masked. All 7-bit received addresses are acknowledged except the reserved address (0b0000xxx and 0b1111xxx). **Note:** When ADDRESS2EN is set, these bits should not be written. If ADDMSK2 is not equal to 0, the reserved I2C addresses (0b0000xxx and 0b1111xxx) are not

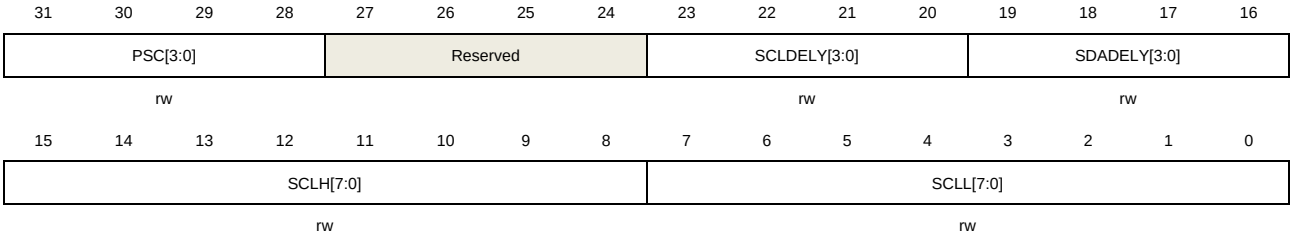
		acknowledged even if all the bits are matched.
7:1	ADDRESS2[6:0]	Second I2C address for the slave **Note:** When ADDRESS2EN is set, these bits should not be written.
0	Reserved	Must be kept at reset value.

20.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).



Bits	Fields	Descriptions
31:28	PSC[3:0]	Timing prescaler In order to generate the clock period t_{PSC} used for data setup and data hold counters, these bits are used to configure the prescaler for I2CCLK. The t_{PSC} is also used for SCL high and low level counters. $t_{PSC}=(PSC+1)*t_{I2CCLK}$
27:24	Reserved	Must be kept at reset value.
23:20	SCLDELY[3:0]	Data setup time A delay $t_{SCLDELY}$ between SDA edge and SCL rising edge can be generated by configuring these bits. And during $t_{SCLDELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$
19:16	SDADELY[3:0]	Data hold time A delay $t_{SDADELY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SDADELY}=SDADELY*t_{PSC}$
15:8	SCLH[7:0]	SCL high period SCL high period can be generated by configuring these bits. $t_{SCLH}=(SCLH+1)*t_{PSC}$ **Note:** These bits can only be used in master mode.
7:0	SCLL[7:0]	SCL low period SCL low period can be generated by configuring these bits.

$t_{SCLL}=(SCLL+1)*t_{PSC}$

Note: These bits can only be used in master mode.

20.4.6. Timeout register (I2C_TIMEOUT)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTOEN	Reserved			BUSTOB[11:0]											
rw				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEN	Reserved		TOIDLE	BUSTOA[11:0]											
rw			rw	rw											

Bits	Fields	Descriptions
31	EXTOEN	Extended clock timeout detection enable When a cumulative SCL stretch time is greater than $t_{LOW:EXT}$, a timeout error will be occurred. $t_{LOW:EXT}=(BUSTOB+1)*2048*t_{I2CCLK}$. 0: Extended clock timeout detection is disabled. 1: Extended clock timeout detection is enabled.
30:28	Reserved	Must be kept at reset value.
27:16	BUSTOB[11:0]	Bus timeout B Configure the cumulative clock extension timeout. In master mode, the master cumulative clock low extend time $t_{LOW:MEXT}$ is detected. In slave mode, the slave cumulative clock low extend time $t_{LOW:SEXT}$ is detected. $t_{LOW:EXT}=(BUSTOB+1)*2048*t_{I2CCLK}$. **Note:** These bits can be modified only when EXTOEN =0.
15	TOEN	Clock timeout detection enable If the SCL stretch time greater than $t_{TIMEOUT}$ when TOIDLE =0 or high for more than t_{IDLE} when TOIDLE =1, a timeout error is detected. 0: SCL timeout detection is disabled 1: SCL timeout detection is enabled
14:13	Reserved	Must be kept at reset value.
12	TOIDLE	Idle clock timeout detection 0: BUSTOA is used to detect SCL low timeout 1: BUSTOA is used to detect both SCL and SDA high timeout when the bus is idle **Note:** This bit can be written only when TOEN =0.
11:0	BUSTOA[11:0]	Bus timeout A

When TOIDLE=0, $t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK}

When TOIDLE=1, t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK}

Note: These bits can be written only when TOEN =0.

20.4.7. Status register (I2C_STAT)

Address offset: 0x18

Reset value: 0x0000 0001

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								READDR[6:0]							TR
								r							r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	Reserved	SMBALT	TIMEOUT	PECERR	OUERR	LOSTARB	BERR	TCR	TC	STPDET	NACK	ADDSEND	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:17	READDR[6:0]	Received match address in slave mode When the ADDSEND bit is set, these bits store the matched address. In the case of a 10-bit address, READDR[6:0] stores the header of the 10-bit address followed by the 2 MSBs of the address.
16	TR	Whether the I2C is a transmitter or a receiver in slave mode This bit is updated when the ADDSEND bit is set. 0: Receiver 1: Transmitter
15	I2CBSY	Busy flag This bit is set by hardware when a START signal is detected and cleared by hardware after a STOP signal. When I2CEN=0, this bit is also cleared by hardware. 0: No I2C communication. 1: I2C communication active.
14	Reserved	Must be kept at reset value.
13	SMBALT	SMBus Alert When SMBHAEN=1, SMBALTEN=1, and a SMBALERT event (falling edge) is detected on SMBA pin, this bit will be set by hardware. It is cleared by software by setting the SMBALTC bit. This bit is cleared by hardware when I2CEN=0. 0: SMBALERT event is not detected on SMBA pin 1: SMBALERT event is detected on SMBA pin
12	TIMEOUT	TIMEOUT flag.

		When a timeout or extended clock timeout occurred, this bit will be set. It is cleared by software by setting the TIMEOUTC bit and cleared by hardware when I2CEN=0. 0: no timeout or extended clock timeout occur 1: a timeout or extended clock timeout occur
11	PECERR	PEC error This flag is set by hardware when the received PEC does not match with the content of I2C_PEC register. Then a NACK is automatically sent. It is cleared by software by setting the PECERRC bit and cleared by hardware when I2CEN=0. 0: Received PEC and content of I2C_PEC match 1: Received PEC and content of I2C_PEC don't match, I2C will send NACK regardless of NACKEN bit.
10	OUERR	Overrun/Underrun error in slave mode In slave mode with SS=1, when an overrun/underrun error occurs, this bit will be set by hardware. It is cleared by software by setting the OUERRC bit and cleared by hardware when I2CEN=0. 0: No overrun or underrun occurs 1: Overrun or underrun occurs
9	LOSTARB	Arbitration Lost It is cleared by software by setting the LOSTARBC bit and cleared by hardware when I2CEN=0. 0: No arbitration lost. 1: Arbitration lost occurs and the I2C block changes back to slave mode.
8	BERR	Bus error When an unexpected START or STOP signal on I2C bus is detected, a bus error occurs and this bit will be set. It is cleared by software by setting BERRC bit and cleared by hardware when I2CEN=0. 0: No bus error 1: A bus error detected
7	TCR	Transfer complete reload This bit is set by hardware when RELOAD=1 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when BYTENUM[7:0] is written to a non-zero value. 0: When RELOAD=1, transfer of BYTENUM[7:0] bytes is not completed 1: When RELOAD=1, transfer of BYTENUM[7:0] bytes is completed
6	TC	Transfer complete in master mode This bit is set by hardware when RELOAD=0, AUTOEND=0 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when START bit or STOP bit is set. 0: Transfer of BYTENUM[7:0] bytes is not completed 1: Transfer of BYTENUM[7:0] bytes is completed
5	STPDET	STOP signal detected in slave mode

		This flag is set by hardware when a STOP signal is detected on the bus. It is cleared by software by setting STPDETC bit and cleared by hardware when I2CEN=0. 0: STOP signal is not detected. 1: STOP signal is detected.
4	NACK	Not Acknowledge flag This flag is set by hardware when a NACK is received. It is cleared by software by setting NACKC bit and cleared by hardware when I2CEN=0. 0: ACK is received. 1: NACK is received.
3	ADDSEND	Address received matches in slave mode. This bit is set by hardware when the received slave address matched with one of the enabled slave addresses. It is cleared by software by setting ADDSENDC bit and cleared by hardware when I2CEN=0. 0: Received address not matched 1: Received address matched
2	RBNE	I2C_RDATA is not empty during receiving This bit is set by hardware when the received data is shift into the I2C_RDATA register. It is cleared when I2C_RDATA is read. 0: I2C_RDATA is empty 1: I2C_RDATA is not empty, software can read
1	TI	Transmit interrupt This bit is set by hardware when the I2C_TDATA register is empty and the I2C is ready to transmit data. It is cleared when the next data to be sent is written in the I2C_TDATA register.When SS=1, this bit can be set by software, in order to generate a TI event (interrupt if TIE=1 or DMA request if DENT =1). 0: I2C_TDATA is not empty or the I2C is not ready to transmit data 1: I2C_TDATA is empty and the I2C is ready to transmit data
0	TBE	I2C_TDATA is empty during transmitting This bit is set by hardware when the I2C_TDATA register is empty. It is cleared when the next data to be sent is written in the I2C_TDATA register. This bit can be set by software in order to empty the I2C_TDATA register. 0: I2C_TDATA is not empty 1: I2C_TDATA is empty

20.4.8. Status clear register (I2C_STATC)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	Reserved		STPDETC	NACKC	ADDSENDC	Reserved		
		w	w	w	w	w	w			w	w	w			

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit
10	OUERRC	Overrun/Underrun flag clear. Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit
9	LOSTARBC	Arbitration Lost flag clear. Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit
7:6	Reservced	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by writing 1 to this bit
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by writing 1 to this bit
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by writing 1 to this bit
2:0	Reserved	Must be kept at reset value.

20.4.9. PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

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

Reserved	PECV[7:0]
	r

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

20.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RDATA[7:0]							
								r							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

20.4.11. Transmit data register (I2C_TDATA)

Address offset: 0x28

Reset value: 0x0000 0000

This register can be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TDATA [7:0]							
								rw							

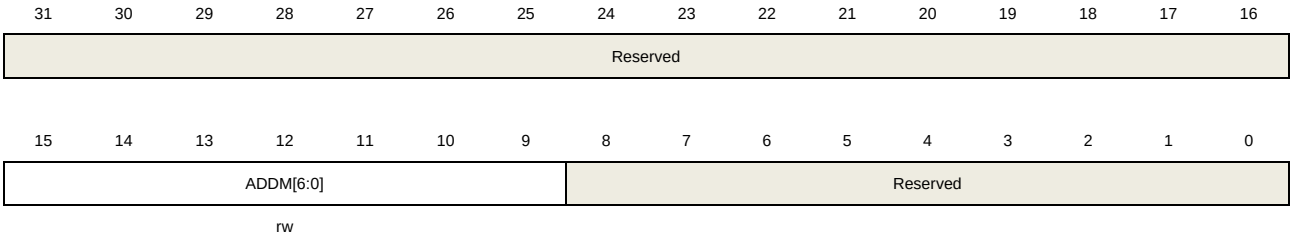
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

20.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

Reset value: 0x0000 FE00

This register can be accessed by word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:9	ADDM[6:0]	Defines which bits of ADDRESS[7:1] are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in ADDM[6:0] enables comparisons with the corresponding bit in ADDRESS[7:1]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
8:0	Reserved	Must be kept at reset value.

21. Serial peripheral interface/Inter-IC sound (SPI/I2S)

21.1. Overview

The SPI/I2S module can communicate with external devices using the SPI protocol or the I2S audio protocol.

The serial peripheral interface (SPI) provides a SPI protocol of data transmission and reception function in master or slave mode. Both full-duplex and simplex communication modes are supported, with hardware CRC calculation and checking. Quad-SPI master mode is only supported in SPI0.

The inter-IC sound (I2S) supports four audio standards: I2S Philips standard, MSB justified standard, LSB justified standard, and PCM standard. I2S works at either master or slave mode for transmission and reception.

21.2. Characteristics

21.2.1. SPI characteristics

- Master or slave operation with full-duplex or simplex mode.
- Separate transmit and receive buffer, 16 bits wide.
- Data frame size can be 8 or 16 bits.
- Bit order can be LSB or MSB.
- Software and hardware NSS management.
- Hardware CRC calculation, transmission and checking.
- Transmission and reception using DMA.
- SPI TI mode supported.
- SPI NSS pulse mode supported.
- Quad-SPI configuration available in master mode (only in SPI0).

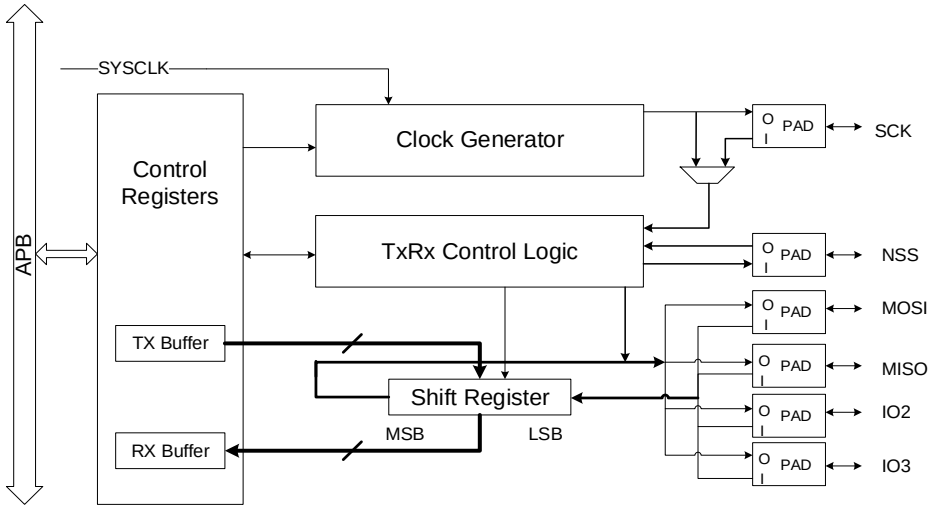
21.2.2. I2S characteristics

- Master or slave operation for transmission/reception.
- Four I2S standards supported: Philips, MSB justified, LSB justified and PCM standard.
- Data length can be 16 bits, 24 bits or 32 bits.
- Channel length can be 16 bits or 32 bits.
- Transmission and reception using a 16 bits wide buffer.
- Audio sample frequency can be 8 kHz to 192 kHz using I2S clock divider.
- Programmable idle state clock polarity.
- Master clock (MCK) can be output.
- Transmission and reception using DMA.

21.3. SPI function overview

21.3.1. SPI block diagram

Figure 21-1. Block diagram of SPI



21.3.2. SPI signal description

Normal configuration (not Quad-SPI mode)

Table 21-1. SPI signal description

Pin name	Direction	Description
SCK	I/O	Master: SPI clock output Slave: SPI clock input
MISO	I/O	Master: Data reception line Slave: Data transmission line Master with bidirectional mode: Not used Slave with bidirectional mode: Data transmission and reception line.
MOSI	I/O	Master: Data transmission line Slave: Data reception line Master with bidirectional mode: Data transmission and reception line. Slave with bidirectional mode: Not used
NSS	I/O	Software NSS mode: not used Master in hardware NSS mode: when NSSDRV=1, it is NSS output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master

Pin name	Direction	Description
		application. Slave in hardware NSS mode: NSS input, as a chip select signal for slave.

Quad-SPI configuration

SPI is in single wire mode by default and enters into Quad-SPI mode after QMOD bit in SPI_QCTL register is set (only available in SPI0). Quad-SPI mode can only work in master mode.

The IO2 and IO3 pins can be driven high in normal Non-Quad-SPI mode by configuring IO23_DRV bit in SPI_QCTL register.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 21-2. Quad-SPI signal description

Pin name	Direction	Description
SCK	O	SPI clock output
MOSI	I/O	Transmission/Reception data 0
MISO	I/O	Transmission/Reception data 1
IO2	I/O	Transmission/Reception data 2
IO3	I/O	Transmission/Reception data 3
NSS	O	NSS output

21.3.3. SPI clock timing and data format

CKPL and CKPH bits in SPI_CTL0 register decide the timing of SPI clock and data signal. The CKPL bit decides the SCK level when idle and CKPH bit decides either first or second clock edge is a valid sampling edge. These bits take no effect in TI mode.

Figure 21-2. SPI timing diagram in normal mode

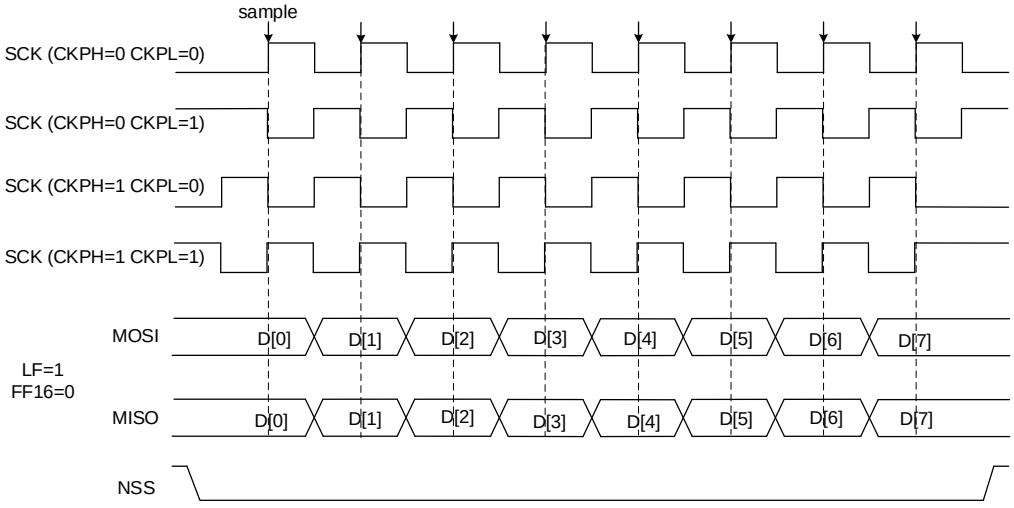
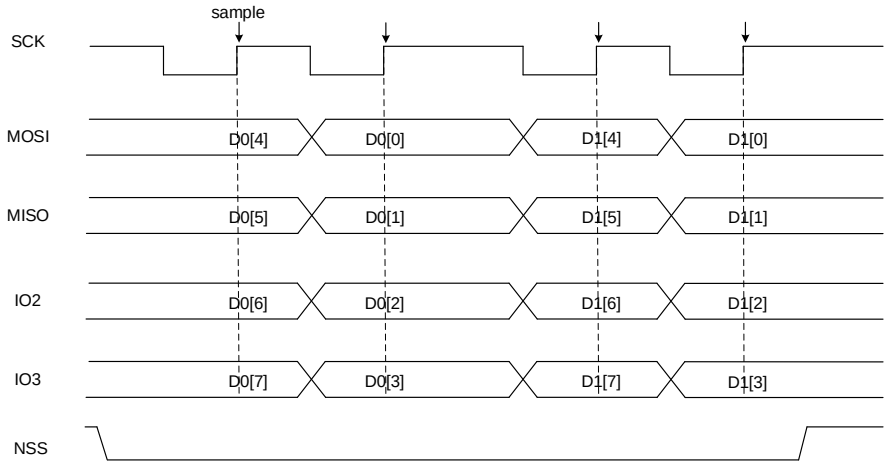


Figure 21-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



In normal mode, the length of data is configured by the FF16 bit in the SPI_CTL0 register. Data length is 16 bits if FF16=1, otherwise is 8 bits. The data frame length is fixed to 8 bits in Quad-SPI mode.

Data order is configured by the LF bit in SPI_CTL0 register, and SPI will first send the LSB first if LF=1, or the MSB first if LF=0. The data order is fixed to MSB first in TI mode.

21.3.4. NSS function

Slave mode

When slave mode is configured (MSTMOD=0), SPI gets NSS level from NSS pin in hardware NSS mode (SWNSSEN = 0) or from SWNSS bit in software NSS mode (SWNSSEN = 1), and SPI transmits/receives data only when NSS level is low. In software NSS mode, NSS pin is not used.

Table 21-3. NSS function in slave mode

Mode	Register configuration	Description
Slave hardware NSS mode	MSTMOD = 0 SWNSSEN = 0	SPI slave gets NSS level from NSS pin.
Slave software NSS mode	MSTMOD = 0 SWNSSEN = 1	SPI slave NSS level is determined by the SWNSS bit. SWNSS = 0: NSS level is low SWNSS = 1: NSS level is high

Master mode

In master mode (MSTMOD=1), if the application uses multi-master connection, NSS can be configured to hardware input mode (SWNSSEN=0, NSSDRV=0) or software mode (SWNSSEN=1). Then, once the NSS pin (in hardware NSS mode) or the SWNSS bit (in

software NSS mode) goes low, the SPI automatically enters slave mode and triggers a master fault flag CONFERR.

If the application wants to use NSS line to control the SPI slave, NSS should be configured to hardware output mode (SWNSSEN=0, NSSDRV=1). NSS goes low after SPI is enabled.

The application may also use a general purpose IO as NSS pin to realize more flexible NSS.

Table 21-4. NSS function in master mode

Mode	Register configuration	Description
Master hardware NSS output mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	Applicable to single-master mode. The master uses the NSS pin to control the SPI slave device. At this time, the NSS is configured as the hardware output mode. NSS goes low after enabling SPI.
Master hardware NSS input mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	Applicable to multi-master mode. At this time, NSS is configured as hardware input mode. Once the NSS pin is pulled low, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be set to 1.
Master software NSS mode	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV: Don’t care	Applicable to multi-master mode. Once SWNSS = 0, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be 1.
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV: Don’t care	The slave can use hardware or software NSS mode.

21.3.5. SPI operating modes

Table 21-5. SPI operating modes

Mode	Description	Register configuration	Data pin usage
MFD	Master full-duplex	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: Don’t care	MOSI: Transmission MISO: Reception
MTU	Master transmission with unidirectional connection	MSTMOD = 1 RO = 0 BDEN = 0	MOSI: Transmission MISO: Not used

Mode	Description	Register configuration	Data pin usage
		BDOEN: Don’t care	
MRU	Master reception with unidirectional connection	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: Don’t care	MOSI: Not used MISO: Reception
MTB	Master transmission with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Transmission MISO: Not used
MRB	Master reception with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Reception MISO: Not used
SFD	Slave full-duplex	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don’t care	MOSI: Reception MISO: Transmission
STU	Slave transmission with unidirectional connection	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don’t care	MOSI: Not used MISO: Transmission
SRU	Slave reception with unidirectional connection	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: Don’t care	MOSI: Reception MISO: Not used
STB	Slave transmission with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Not used MISO: Transmission
SRB	Slave reception with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Not used MISO: Reception

Figure 21-4. A typical full-duplex connection

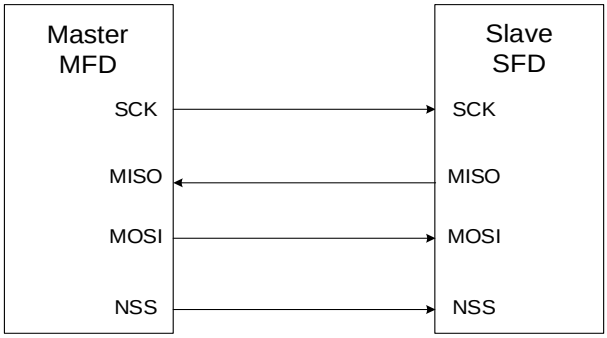


Figure 21-5. A typical simplex connection (Master: Receive, Slave: Transmit)

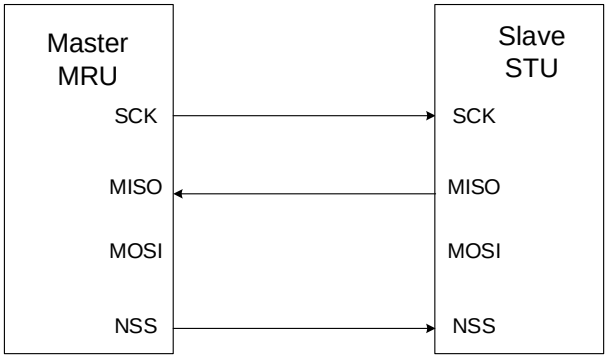


Figure 21-6. A typical simplex connection (Master: Transmit only, Slave: Receive)

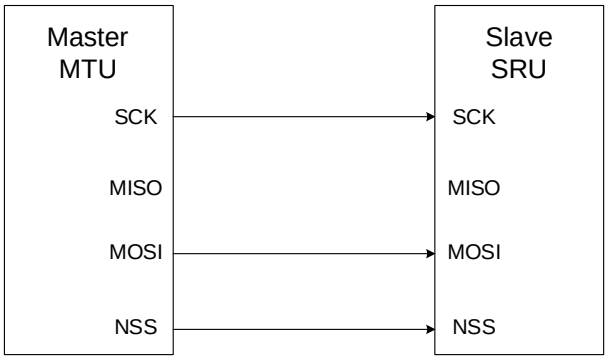
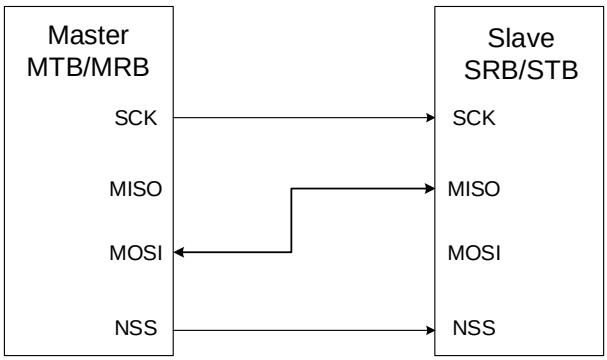


Figure 21-7. A typical bidirectional connection



SPI initialization sequence

Before transmiting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode or slave TI mode is used, program the PSC [2:0] bits in SPI_CTL0 register to generate SCK with desired baud rate or configure the Td time in TI mode, otherwise, ignore this step.
2. Program data format (FF16 bit in the SPI_CTL0 register).
3. Program the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
4. Program the frame format (LF bit in the SPI_CTL0 register).
5. Program the NSS mode (SWNSSEN bit in the SPI_CTL0 register and NSSDRV bit in the SPI_CTL1 register) according to the application's demand as described above in ***NSS function*** section.
6. If TI mode is used, set TMOD bit in SPI_CTL1 register, otherwise, ignore this step.
7. Configure MSTMOD, RO, BDEN and BDOEN depending on the operating modes described in***Table 21-5. SPI operating modes*** section.
8. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.
9. Enable the SPI (set the SPIEN bit).

Note: During communication, CKPH, CKPL, MSTMOD, PSC[2:0] and LF bits should not be changed.

SPI basic transmission and reception sequence

Transmission sequence

After the initialization sequence, the SPI is enabled and stays at idle state. In master mode, the transmission starts when the application writes a data into the transmit buffer. In slave mode, the transmission starts when SCK clock signal at SCK pin begins to toggle and NSS level is low, so application should ensure that data is already written into transmit buffer before the transmission starts in slave mode.

When SPI begins to send a data frame, it first loads this data frame from the data buffer to the shift register and then begins to transmit the loaded data frame, TBE (transmit buffer empty) flag is set after the first bit of this frame is transmitted. After TBE flag is set, which means the transmit buffer is empty, the application should write SPI_DATA register again if it has more data to transmit.

In master mode, software should write the next data into SPI_DATA register before the transmission of current data frame is completed if it desires to generate continuous transmission.

Reception sequence

After the last valid sample clock, the incoming data will be moved from shift register to the receive buffer and RBNE (receive buffer not empty) will be set. The application should read

SPI_DATA register to get the received data and this will clear the RBNE flag automatically. In MRU and MRB modes, hardware continuously sends clock signal to receive the next data frame, while in full-duplex master mode (MFD), hardware only receives the next data frame when the transmit buffer is not empty.

SPI operation sequence in different modes (Not Quad-SPI, TI mode or NSSP mode)

In full-duplex mode, either MFD or SFD, the RBNE and TBE flags should be monitored by the application. And then application should follow the sequences described above.

The transmission mode (MTU, MTB, STU or STB) is similar to the transmission sequence of full-duplex mode regardless of the RBNE and OVRE bits.

The master reception mode (MRU or MRB) is different from the reception sequence of full-duplex mode. In MRU or MRB mode, the SPI continuously generates SCK until the SPI is disabled after SPI is enabled. So the application should ignore the TBE flag and read out reception buffer in time after the RBNE flag is set, otherwise a data overrun fault will occur.

The slave reception mode (SRU or SRB) is similar to the reception sequence of full-duplex mode regardless of the TBE flag.

SPI TI mode

SPI TI mode takes NSS as a special frame header flag signal and its operation sequence is similar to normal mode described above. The modes described above (MFD, MTU, MRU, MTB, MRB, SFD, STU, SRU, STB and SRB) are still supported in TI mode. While, in TI mode the CKPL and CKPH bits in SPI_CTL0 registers take no effect and the SCK sample edge is falling edge.

Figure 21-8. Timing diagram of TI master mode with discontinuous transfer

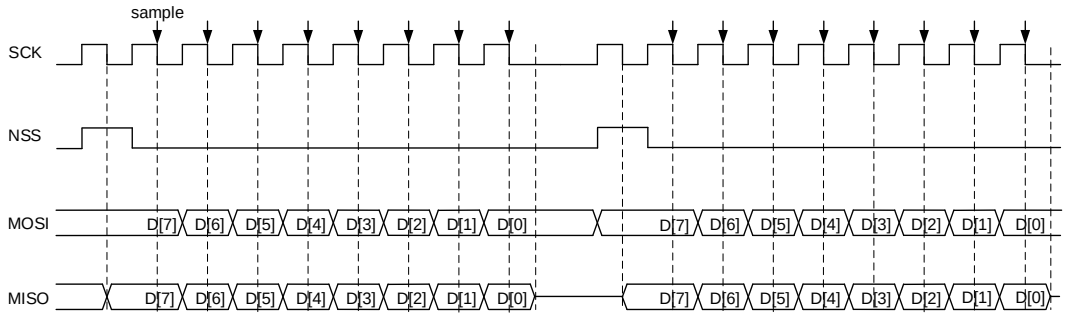
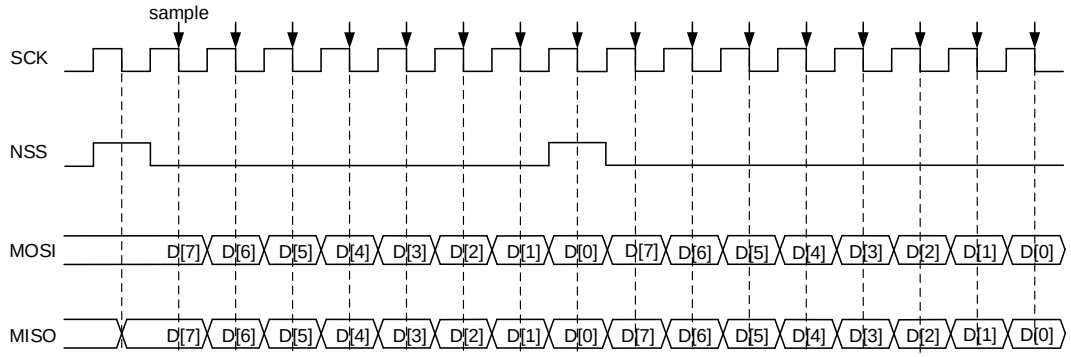
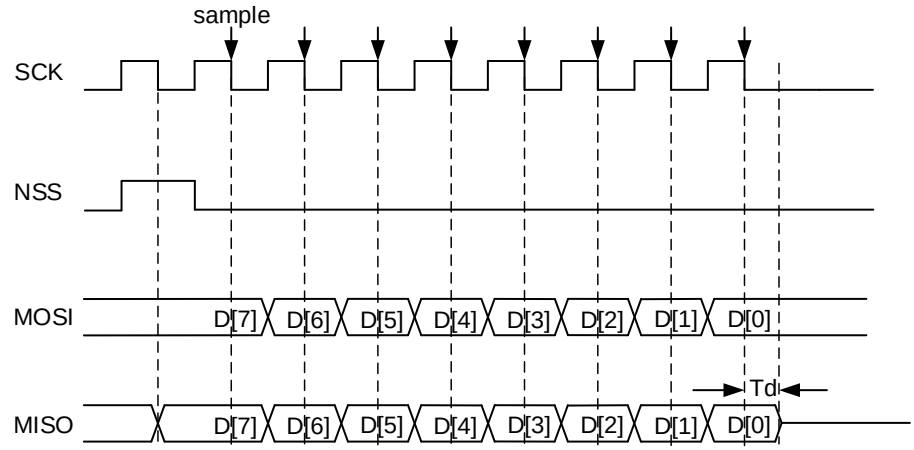


Figure 21-9. Timing diagram of TI master mode with continuous transfer



In master TI mode, SPI can perform continuous or non-continuous transfer. If the master writes SPI_DATA register fast enough, the transfer is continuous, otherwise non-continuous. In non-continuous transfer, there is an extra header clock cycle before each byte. While in continuous transfer, the extra header clock cycle only exists before the first byte and the following bytes' header clock is overlaid at the last bit of pervious bytes.

Figure 21-10. Timing diagram of TI slave mode



In slave TI mode, after the last rising edge of SCK in transfer, the slave begins to transmit the LSB bit of the last data byte, and after a half-bit time, the master begins to sample the line. To make sure that the master samples the right value, the slave should continue to drive this bit after the falling sample edge of SCK for a period of time before releasing the pin. This time is called T_d. T_d is decided by PSC [2:0] bits in SPI_CTL0 register.

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (21\text{-}1)$$

For example, if PSC[2:0] = 010, T_d is $9*T_{pclk}$.

In slave mode, the slave also monitors the NSS signal and sets an error flag FERR if it detects an incorrect NSS behavior, for example, toggles at the middle bit of a byte.

NSS pulse mode operation sequence

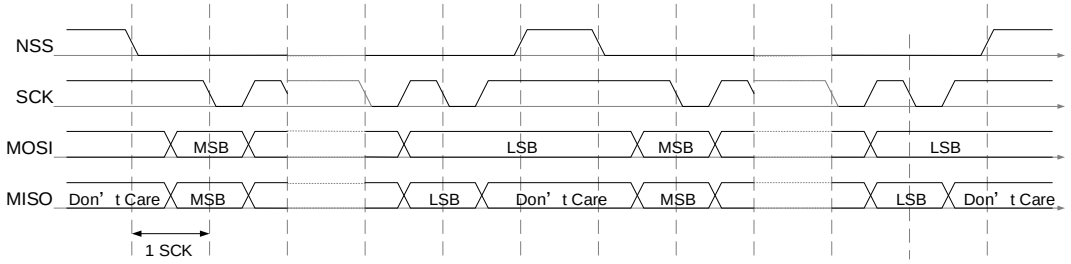
This function is controlled by NSSP bit in SPI_CTL1 register, for this function to fully take place, several additional conditions must be met, users must also set the device into master

mode, and frame format should follow the normal SPI protocol, and set the data capture edge to first clock transition.

In summary: NSSP = 1; MSTMOD = 1; CKPH = 0;

When active, a pluse duration of least 1 SCK clock priod is inserted between successive data frames depending on internal data transmit buffer status, multiple SCK clock cycle interval is possible if the transfer buffer stays empty. This function is designed for single master-slave configuration for the slave to latch data. The following diagram depicts its timing diagram.

Figure 21-11. Timing diagram of NSS pulse with continuous transmit



Quad-SPI mode operation sequence

The Quad-SPI mode is designed to control Quad-SPI flash.

In order to enter Quad-SPI mode, the software should first verify that the TBE bit is set and TRANS bit is cleared, then set QMOD bit in SPI_QCTL register. In Quad-SPI mode, BDEN, BDOEN, CRCEN, CRCNT, FF16, RO and LF bits in SPI_CTL0 register should be kept cleared and MSTMOD should be set to ensure that SPI is in master mode. SPIEN, PSC, CKPL and CKPH bits should be configured as desired.

There are two operation modes in Quad-SPI mode: quad write and quad read, decided by QRD bit in SPI_QCTL register.

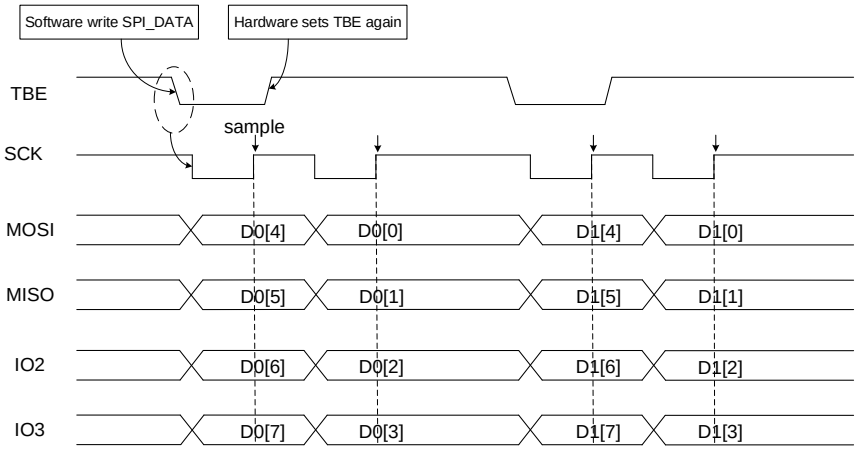
Quad write operation

SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Once SPI starts transmission, it always checks TBE status at the end of a frame and stops when condition is not met.

The operation flow for transmitting in quad mode is shown below:

1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 based on application requirements.
2. Set QMOD bit in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0.
3. Write a byte of data to SPI_DATA register and the TBE will be cleared.
4. Wait until TBE is set by hardware again before writing the next byte.

Figure 21-12. Timing diagram of quad write operation in Quad-SPI mode



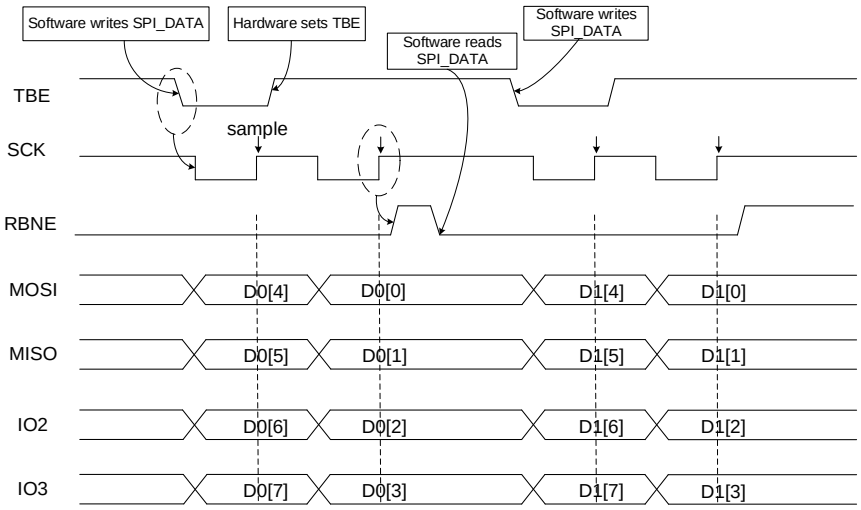
Quad read operation

SPI works in quad read mode when QMOD and QRD bits are both set in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as input pins. SPI begins to generate clock on SCK line as soon as a data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Writing data into SPI_DATA is only to generate SCK clocks, so the written data can be any value. Once SPI starts transmission, it always checks SPIEN and TBE status at the end of a frame and stops when condition is not met. So, dummy data should always be written into SPI_DATA to generate SCK.

The operation flow for receiving in quad mode is shown below:

1. Configure clock prescaler, clock polarity, phase, etc. It based on application requirements in SPI_CTL0 and SPI_CTL1 register.
2. Set QMOD and QRD bits in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0 register.
3. Write an arbitrary byte (for example, 0xFF) to SPI_DATA register.
4. Wait until the RBNE flag is set and read SPI_DATA to get the received byte.
5. Write an arbitrary byte (for example, 0xFF) to SPI_DATA to receive the next byte.

Figure 21-13. Timing diagram of quad read operation in Quad-SPI mode



SPI disabling sequence

Different sequences are used to disable the SPI in different operation modes.

MFD SFD

Wait for the last RBNE flag and then receive the last data. Confirm that TBE=1 and TRANS=0. At last, disable the SPI by clearing SPIEN bit.

MTU MTB STU STB

Write the last data into SPI_DATA and wait until the TBE flag is set and then wait until the TRANS flag is cleared. Disable the SPI by clearing SPIEN bit.

MRU MRB

After getting the second last RBNE flag, read out this data and delay for a SCK clock time and then, disable the SPI by clearing SPIEN bit. Wait until the last RBNE flag is set and read out the last data.

SRU SRB

Application can disable the SPI when it doesn't want to receive data, and then wait until the TRANS=0 to ensure the ongoing transfer completes.

TI mode

The disabling sequence of TI mode is the same as the sequences described above.

NSS pulse mode

The disabling sequence of NSSP mode is the same as the sequences described above.

Quad-SPI mode

Before leaving quad wire mode or disabling SPI, software should first check that TBE bit is set and TRANS bit is cleared, then the QMOD bit in SPI_QCTL register and SPIEN bit in SPI_CTL0 register are cleared.

21.3.6. DMA function

The DMA frees the application from data writing and reading process during transfer to improve the system efficiency.

DMA function in SPI is enabled by setting DMATEN and DMAREN bits in SPI_CTL1 register. To use DMA function, application should first correctly configure DMA modules, then configure SPI module according to the initialization sequence, at last enable SPI.

After being enabled, if DMATEN is set, SPI will generate a DMA request each time when TBE=1, then DMA will acknowledge to this request and write data into the SPI_DATA register automatically. If DMAREN is set, SPI will generate a DMA request each time when RBNE=1, then DMA will acknowledge to this request and read data from the SPI_DATA register automatically.

21.3.7. CRC function

There are two CRC calculators in SPI: one for transmission and the other for reception. The CRC calculation uses the polynomial defined in SPI_CRCPOLY register.

Application can enable the CRC function by setting CRCEN bit in SPI_CTL0 register. The CRC calculators continuously calculate CRC for each bit transmitted and received on lines, and the calculated CRC values can be read from SPI_TCRC and SPI_RCRC registers.

To transmit the calculated CRC value, application should set the CRCNT bit in SPI_CTL0 register after the last data is written to the transmit buffer. In full-duplex mode (MFD or SFD), when the SPI transmits a CRC and prepares to check the received CRC value, the SPI treats the incoming data as a CRC value. In reception mode (MRB, MRU, SRU and SRB), the application should set the CRCNT bit after the second last data frame is received. When CRC checking fails, the CRCERR flag will be set.

If DMA function is enabled, application doesn’t need to configure CRCNT bit and hardware will automatically process the CRC transmitting and checking.

Note: When SPI is in slave mode and CRC function is enable, the CRC calculator is sensitive to input SCK clock whether SPI is enable or not. The software must enable CRC only when the clock is stable to avoid wrong CRC calculation. And when SPI works as a slave, the NSS internal signal needs to be kept low between the data phase and CRC phase.

21.3.8. SPI interrupts

Status flags

- **Transmit buffer empty flag (TBE)**

This bit is set when the transmit buffer is empty, the software can write the next data to the transmit buffer by writing the SPI_DATA register.

- **Receive buffer not empty flag (RBNE)**

This bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer. And software can read the data by reading the SPI_DATA register.

- **SPI transmitting ongoing flag (TRANS)**

TRANS is a status flag to indicate whether the transfer is ongoing or not. It is set and cleared by hardware and not controlled by software. This flag doesn't generate any interrupt.

Error flags

- **Configuration fault error (CONFERR)**

CONFERR is an error flag in master mode. In NSS hardware mode and if the NSSDRV is not enabled, the CONFERR is set when the NSS pin is pulled low. In NSS software mode, the CONFERR is set when the SWNSS bit is 0. When the CONFERR is set, the SPIEN bit and the MSTMOD bit are cleared by hardware, the SPI is disabled and the device is forced into slave mode.

The SPIEN bit and MSTMOD bit are write protected until the CONFERR is cleared. The CONFERR bit of the slave cannot be set. In a multi-master configuration, the device can be in slave mode with CONFERR bit set, which means there might have been a multi-master conflict for system control.

- **Rx overrun error (RXORERR)**

The RXORERR bit is set if a data is received when the RBNE is set. That means, the last data has not been read out and the newly incoming data is received. The receive buffer contents won't be covered with the newly incoming data, so the newly incoming data is lost.

- **Format error (FERR)**

In slave TI mode, the slave also monitors the NSS signal and set an error flag if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

- **CRC error (CRCERR)**

When the CRCEN bit is set, the CRC calculation result of the received data in the SPI_RCRC register is compared with the received CRC value after the last data, the CRCERR is set when they are different.

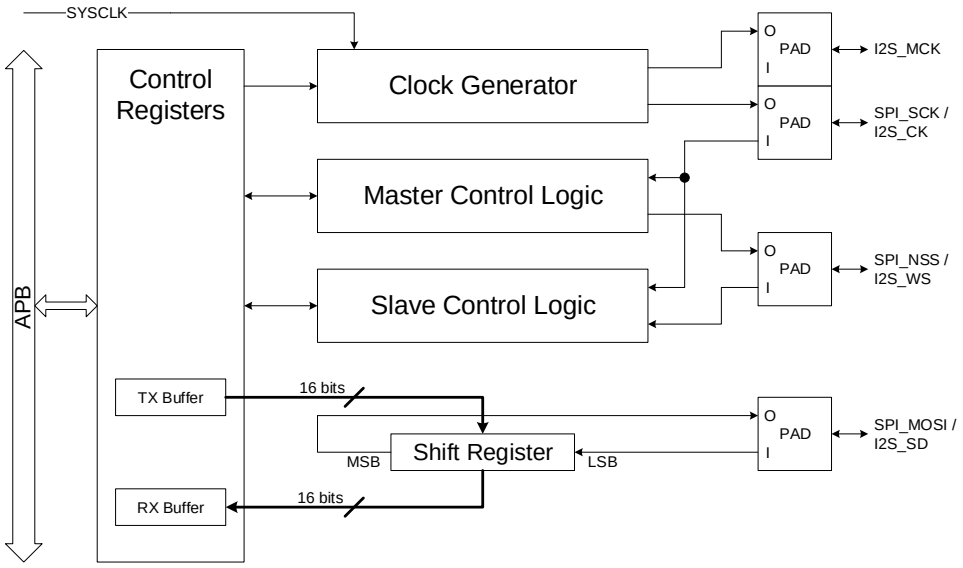
Table 21-6. SPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer empty	Write SPI_DATA register.	TBEIE
RBNE	Receive buffer not empty	Read SPI_DATA register.	RBNEIE
CONFERR	Configuration fault error	Read or write SPI_STAT register, then write SPI_CTL0 register.	ERRIE
RXORERR	Rx overrun error	Read SPI_DATA register, then read SPI_STAT register.	
CRCERR	CRC error	Write 0 to CRCERR bit	
FERR	TI mode format error	Write 0 to FERR bit	

21.4. I2S function overview

21.4.1. I2S block diagram

Figure 21-14. Block diagram of I2S



There are five sub modules to support I2S function, including control registers, clock generator, master control logic, slave control logic and shift register. All the user configuration registers are implemented in the control registers module, including the TX buffer and RX buffer. The clock generator is used to produce I2S communication clock in master mode. The master control logic is implemented to generate the I2S_WS signal and control the communication in master mode. The slave control logic is implemented to control the communication in slave mode according to the received I2S_CK and I2S_WS. The shift register handles the serial data transmission and reception on I2S_SD.

21.4.2. I2S signal description

There are four pins on the I2S interface, including I2S_CK, I2S_WS, I2S_SD and I2S_MCK. I2S_CK is the serial clock signal, which shares the same pin with SPI_SCK. I2S_WS is the frame control signal, which shares the same pin with SPI_NSS. I2S_SD is the serial data signal, which shares the same pin with SPI_MOSI. I2S_MCK is the master clock signal. It produces a frequency rate equals to 256 x Fs, and Fs is the audio sampling frequency.

21.4.3. I2S audio standards

The I2S audio standard is selected by the I2SSTD bits in the SPI_I2SCTL register. Four audio standards are supported, including I2S Philips standard, MSB justified standard, LSB justified standard, and PCM standard. All standards except PCM handle audio data time-multiplexedly on two channels (the left channel and the right channel). For these standards, the I2S_WS signal indicates the channel side. For PCM standard, the I2S_WS signal indicates frame synchronization information.

The data length and the channel length are configured by the DTLEN bits and CHLEN bit in the SPI_I2SCTL register. Since the channel length must be greater than or equal to the data length, four packet types are available. They are 16-bit data packed in 16-bit frame, 16-bit data packed in 32-bit frame, 24-bit data packed in 32-bit frame, and 32-bit data packed in 32-bit frame. The data buffer for transmission and reception is 16-bit wide. In the case that the data length is 24 bits or 32 bits, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In the case that the data length is 16 bits, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. When using 16-bit data packed in 32-bit frame, 16-bit 0 is inserted by hardware automatically to extend the data to 32-bit format.

For all standards and packet types, the most significant bit (MSB) is always sent first. For all standards based on two channels time-multiplexed, the channel left is always sent first followed by the channel right.

I2S Philips standard

For I2S Philips standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The timing diagrams for each configuration are shown below.

Figure 21-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

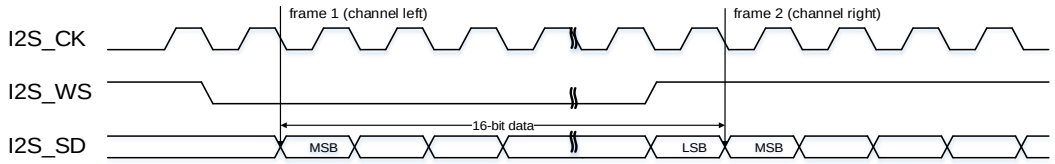
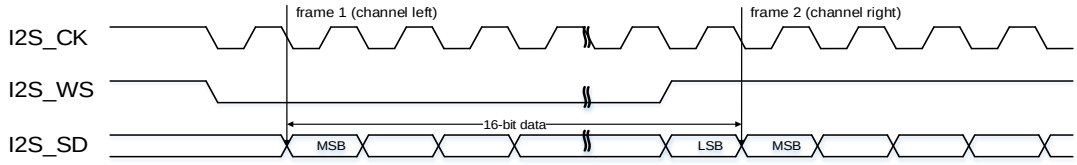


Figure 21-16. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)



When the packet type is 16-bit data packed in 16-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame.

Figure 21-17. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

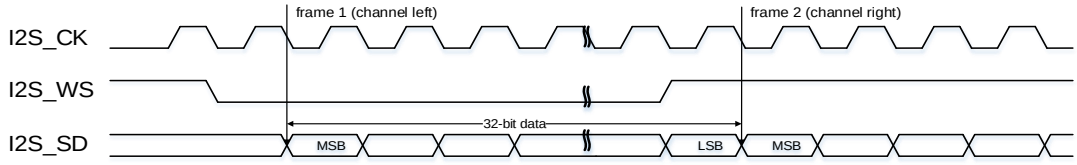
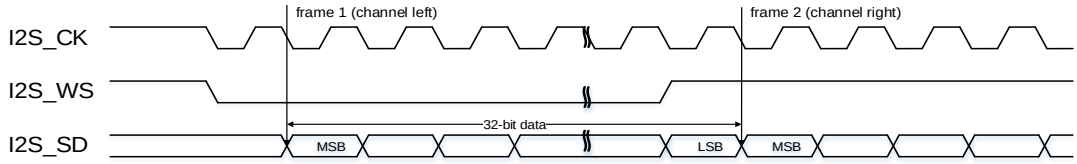


Figure 21-18. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)



When the packet type is 32-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 32-bit data is going to be sent, the first data written to the SPI_DATA register should be the higher 16 bits, and the second one should be the lower 16 bits. In reception mode, if a 32-bit data is received, the first data read from the SPI_DATA register should be the higher 16 bits, and the second one should be the lower 16 bits.

Figure 21-19. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

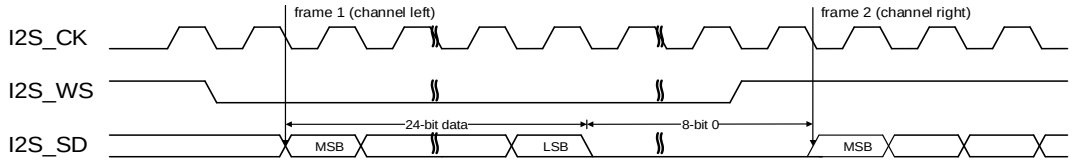
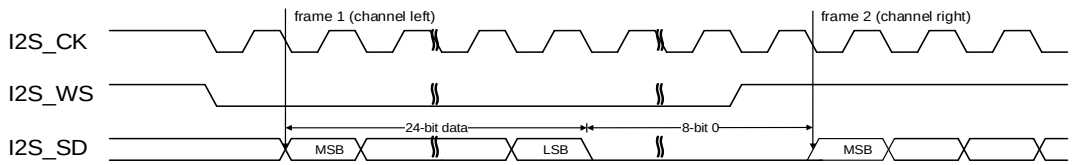


Figure 21-20. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 24-bit data D[23:0] is going to be sent, the first data written to the

SPI_DATA register should be the higher 16 bits D[23:8]. And the second one should be a 16-bit data, the higher 8 bits of this 16-bit data should be D[7:0] and the lower 8 bits can be any value. In reception mode, if a 24-bit data D[23:0] is received, the first data read from the SPI_DATA register is D[23:8]. And the second one is a 16-bit data, the higher 8 bits of this 16-bit data are D[7:0] and the lower 8 bits are zeros.

Figure 21-21. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

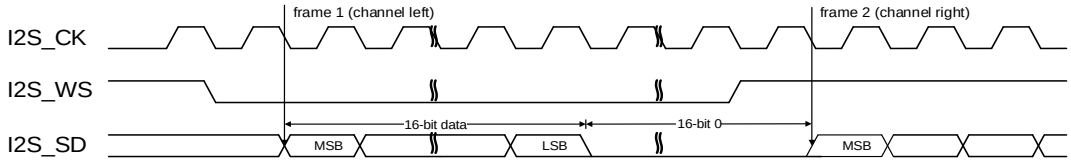
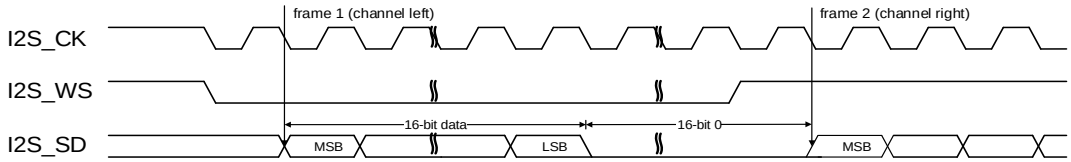


Figure 21-22. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. The remaining 16 bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

MSB justified standard

For MSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The SPI_DATA register is handled in the exactly same way as that for I2S Philips standard. The timing diagrams for each configuration are shown below.

Figure 21-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

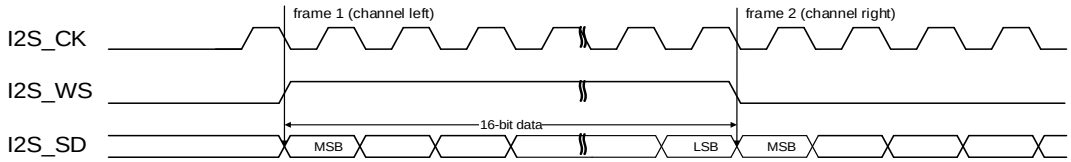


Figure 21-24. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

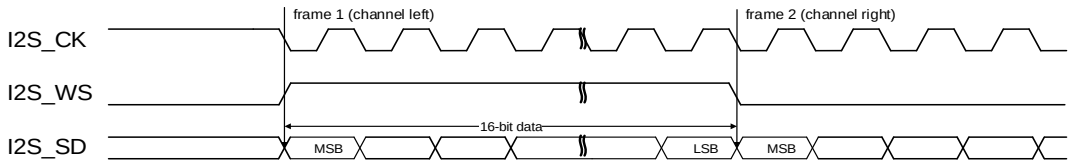


Figure 21-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

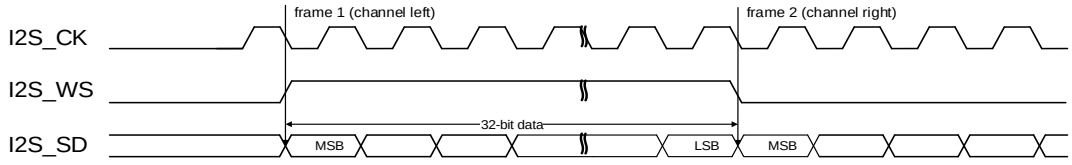


Figure 21-26. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

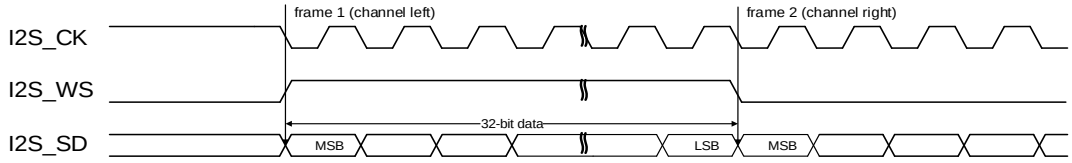


Figure 21-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

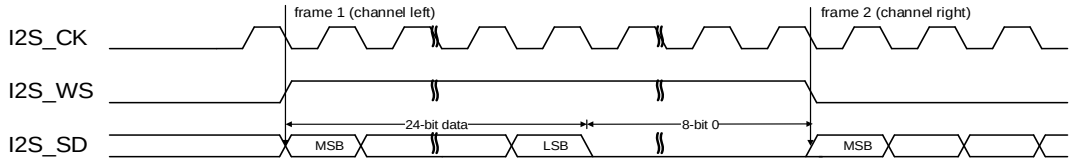


Figure 21-28. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

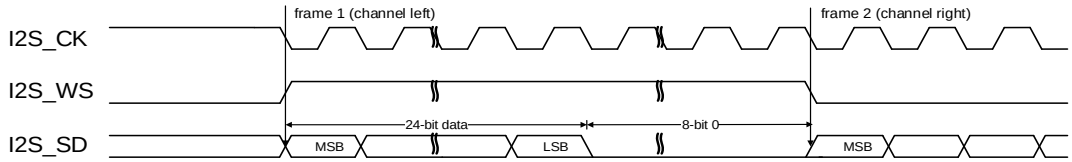


Figure 21-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

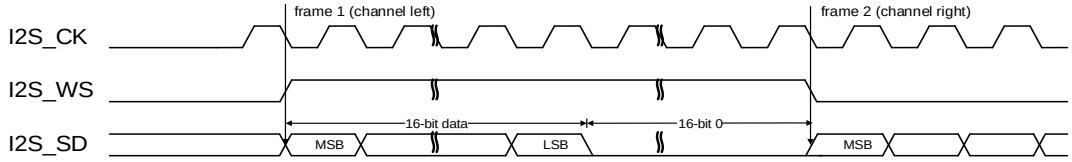
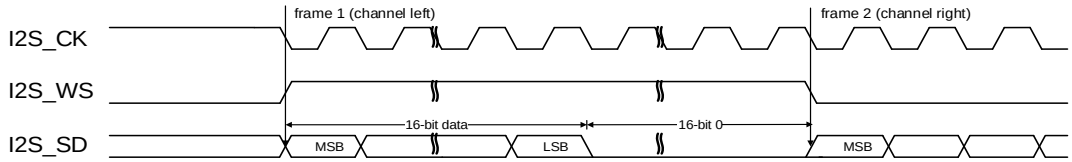


Figure 21-30. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



LSB justified standard

For LSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. In the case that the channel length is equal to the data length, LSB justified standard and MSB justified standard are exactly the same. In the case that the channel length is greater

than the data length, the valid data is aligned to LSB for LSB justified standard while the valid data is aligned to MSB for MSB justified standard. The timing diagrams for the cases that the channel length is greater than the data length are shown below.

Figure 21-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

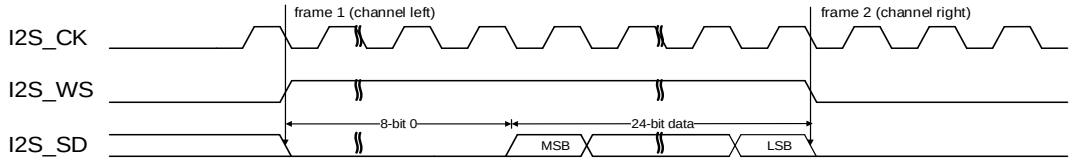
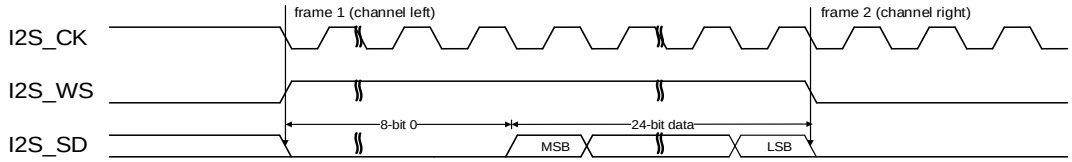


Figure 21-32. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 24-bit data D[23:0] is going to be sent, the first data written to the SPI_DATA register should be a 16-bit data. The higher 8 bits of the 16-bit data can be any value and the lower 8 bits should be D[23:16]. The second data written to the SPI_DATA register should be D[15:0]. In reception mode, if a 24-bit data D[23:0] is received, the first data read from the SPI_DATA register is a 16-bit data. The high 8 bits of this 16-bit data are zeros and the lower 8 bits are D[23:16]. The second data read from the SPI_DATA register is D[15:0].

Figure 21-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

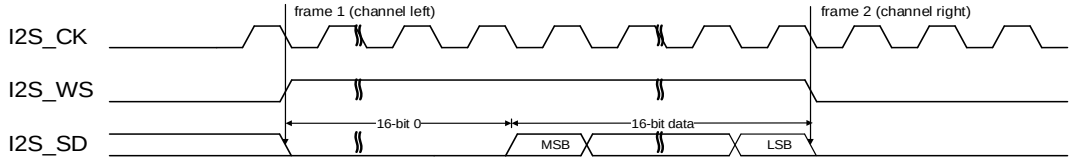
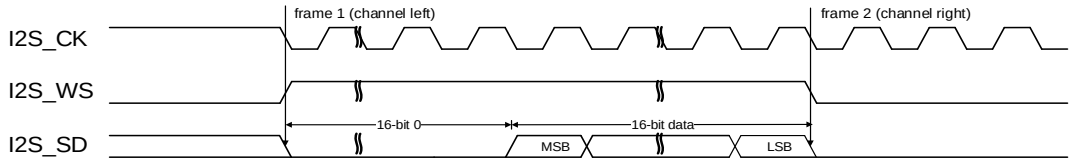


Figure 21-34. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. The remaining 16 bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

PCM standard

For PCM standard, I2S_WS and I2S_SD are updated on the rising edge of I2S_CK, and the I2S_WS signal indicates frame synchronization information. Both the short frame synchronization mode and the long frame synchronization mode are available and configurable using the PCMSMOD bit in the SPI_I2SCTL register. The SPI_DATA register is handled in the exactly same way as that for I2S Philips standard. The timing diagrams for each configuration of the short frame synchronization mode are shown below.

Figure 21-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

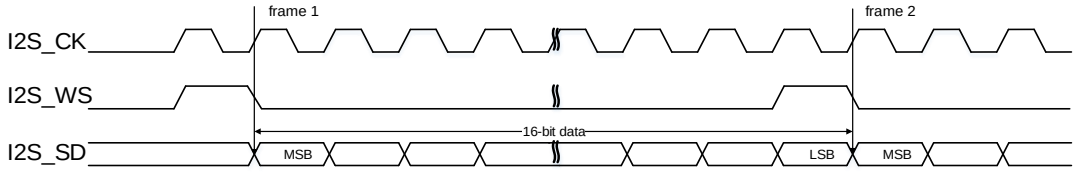


Figure 21-36. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

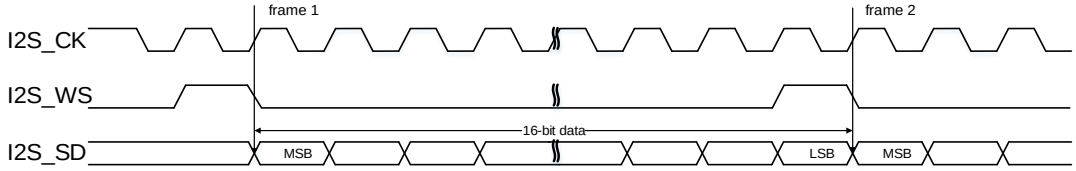


Figure 21-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

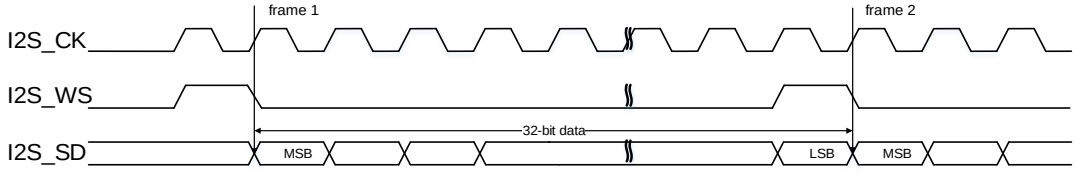


Figure 21-38. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

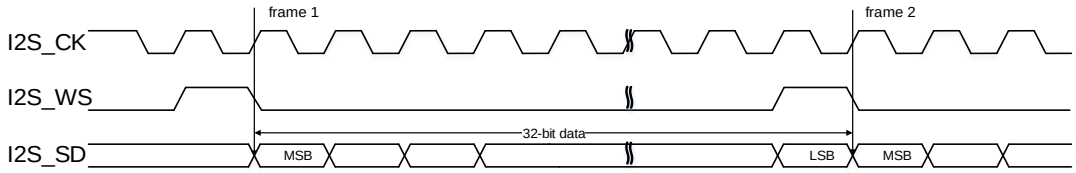


Figure 21-39. PCM standard short frame synchronization mode timing diagram

(DTLEN=01, CHLEN=1, CKPL=0)

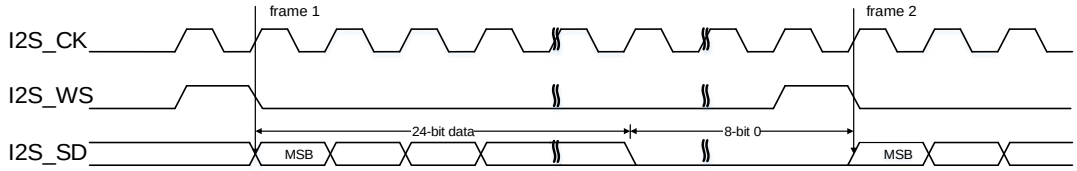


Figure 21-40. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

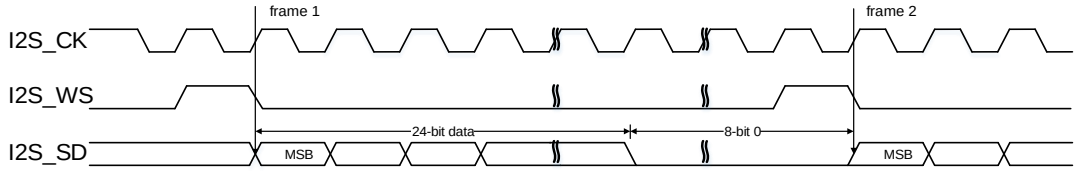


Figure 21-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

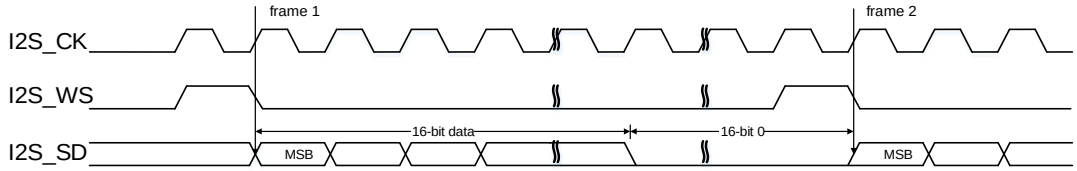
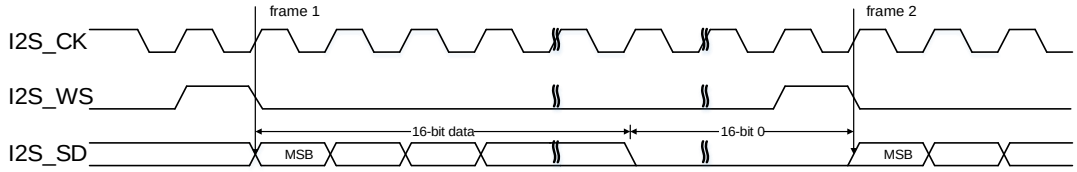


Figure 21-42. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 21-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

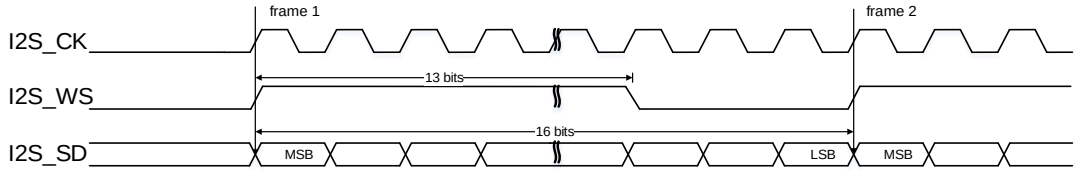


Figure 21-44. PCM standard long frame synchronization mode timing diagram

(DTLEN=00, CHLEN=0, CKPL=1)

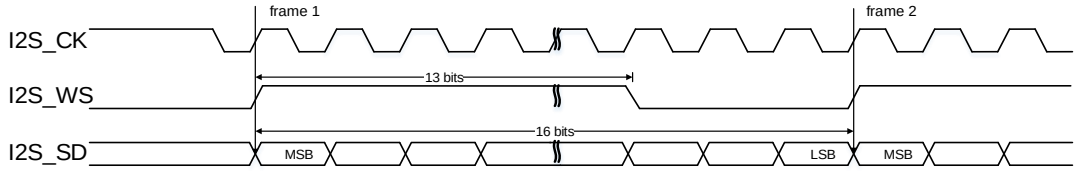


Figure 21-45. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

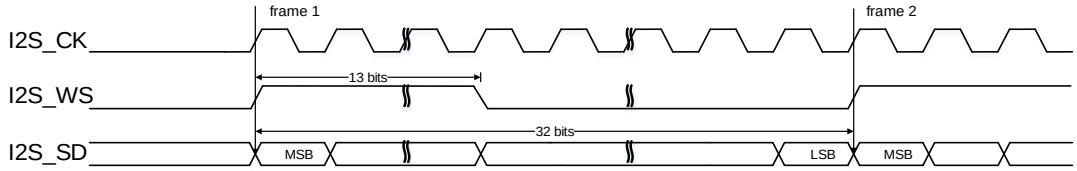


Figure 21-46. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

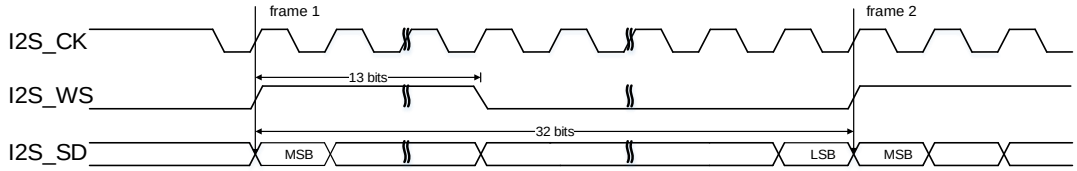


Figure 21-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

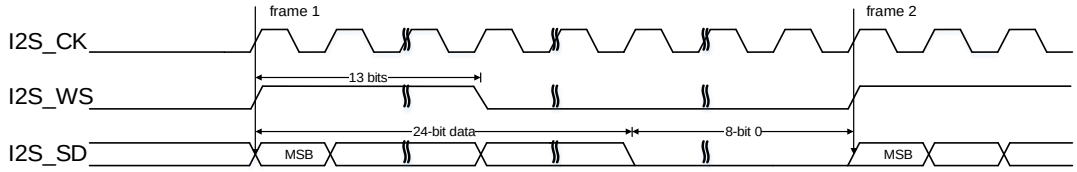


Figure 21-48. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

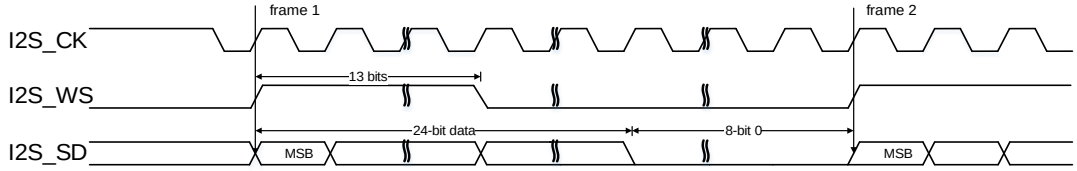


Figure 21-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

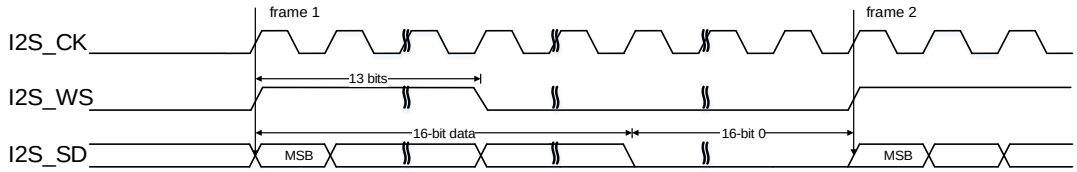
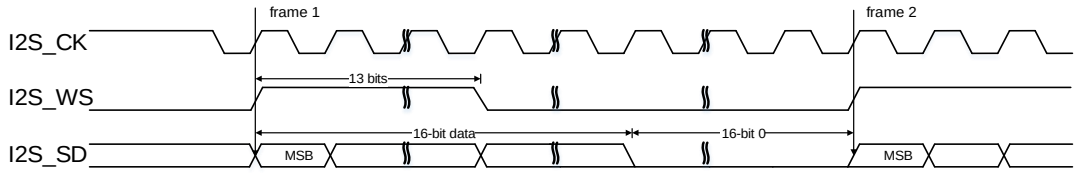
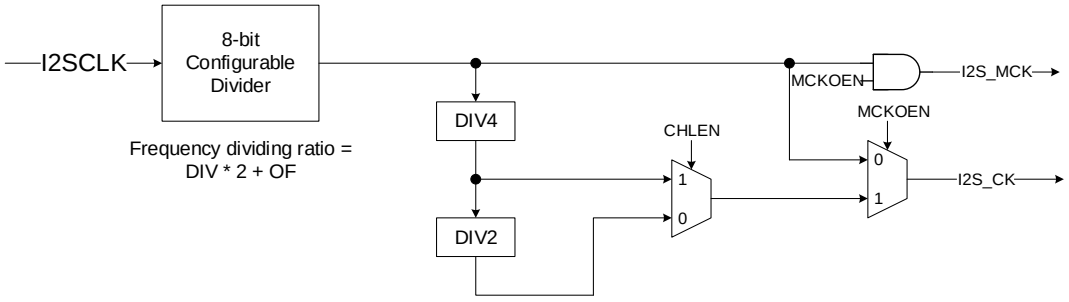


Figure 21-50. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



21.4.4. I2S clock

Figure 21-51. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as ***Figure 21-51. Block diagram of I2S clock generator***. The I2S interface clocks are configured by the DIV bits, the OF bit, the MCKOEN bit in the SPI_I2SPSC register and the CHLEN bit in the SPI_I2SCTL register. The source clock is the system clock(CK_SYS). The I2S bitrate can be calculated by the formulas shown in ***Table 21-7. I2S bitrate calculation formulas***.

Table 21-7. I2S bitrate calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

The relationship between audio sampling frequency (Fs) and I2S bitrate is defined by the following formula:

Fs = I2S bitrate / (number of bits per channel * number of channels)

So, in order to get the desired audio sampling frequency, the clock generator needs to be configured according to the formulas listed in ***Table 21-8. Audio sampling frequency calculation formulas.***

Table 21-8. Audio sampling frequency calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

Note: The configuration value of the I2S serial clock must be set to less than 1/6 of the PCLK clock (excluding 1/6).

21.4.5. Operation

Operation modes

The operation mode is selected by the I2SOPMOD bits in the SPI_I2SCTL register. There are four available operation modes, including master transmission mode, master reception mode, slave transmission mode, and slave reception mode. The direction of I2S interface signals for each operation mode is shown in the ***Table 21-9. Direction of I2S interface signals for each operation mode.***

Table 21-9. Direction of I2S interface signals for each operation mode

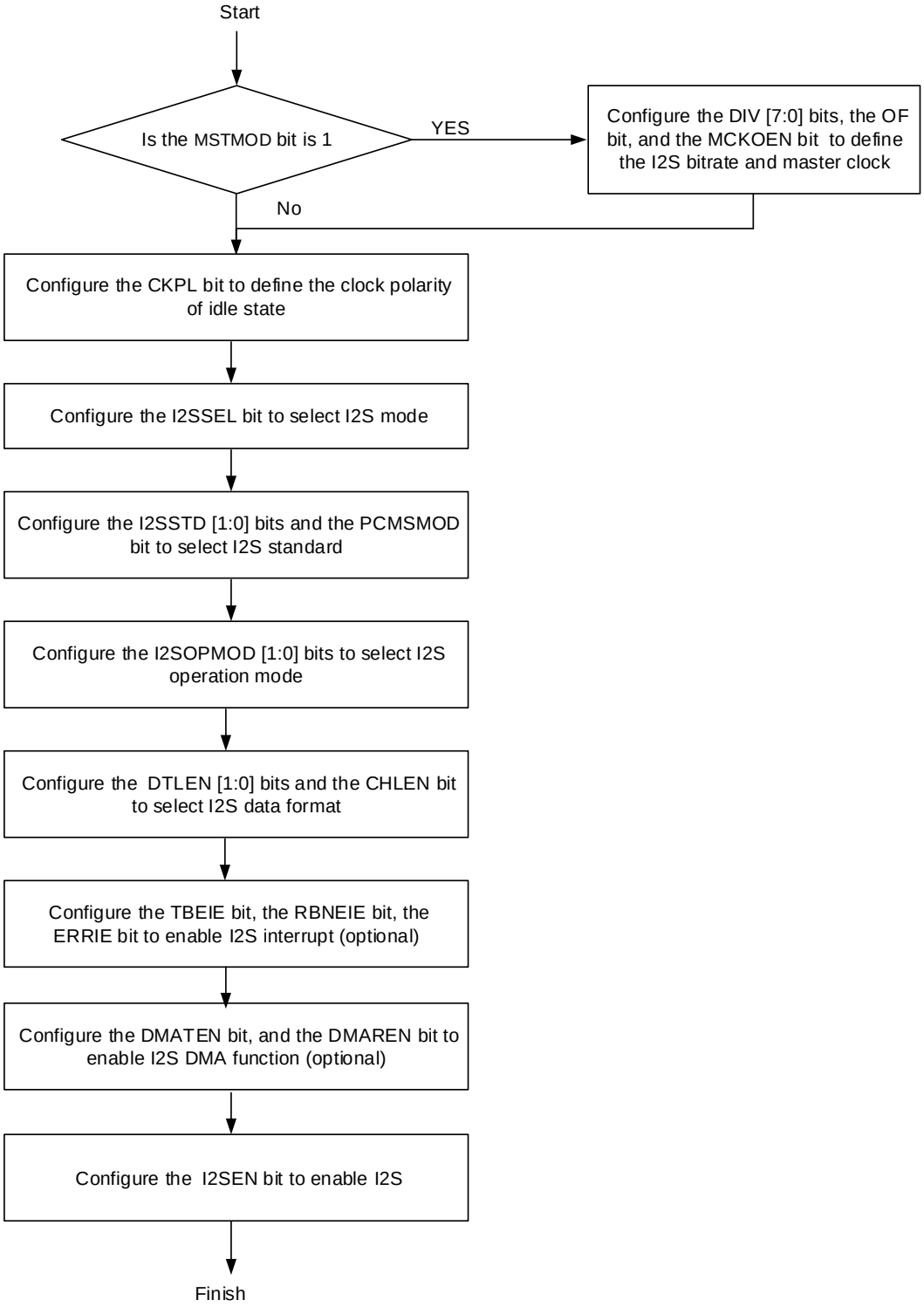
Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
Master transmission	Output or NU(1)	Output	Output	Output
Master reception	Output or NU(1)	Output	Output	Input
Slave transmission	Input or NU(1)	Input	Input	Output
Slave reception	Input or NU(1)	Input	Input	Input

1. NU means the pin is not used by I2S and can be used by other functions.

I2S initialization sequence

I2S initialization sequence is shown as ***Figure 21-52. I2S initialization sequence.***

Figure 21-52. I2S initialization sequence



I2S master transmission sequence

The TBE flag is used to control the transmission sequence. As is mentioned before, the TBE flag indicates that the transmit buffer is empty, and an interrupt will be generated if the TBEIE bit in the SPI_CTL1 register is set. At the beginning, the transmit buffer is empty (TBE is high) and no transmission sequence is processing in the shift register. When a half word is written

to the SPI_DATA register (TBE goes low), the data is transferred from the transmit buffer to the shift register (TBE goes high) immediately. At the moment, the transmission sequence begins.

The data is parallel loaded into the 16-bit shift register, and shifted out serially to the I2S_SD pin, MSB first. The next data should be written to the SPI_DATA register, when the TBE flag is high. After a write operation to the SPI_DATA register, the TBE flag goes low. When the current transmission finishes, the data in the transmit buffer is loaded into the shift register, and the TBE flag goes back high. Software should write the next audio data into SPI_DATA register before the current data finishes, otherwise, the audio data transmission is not continuous.

For all standards except PCM, the I2SCH flag is used to distinguish which channel support the data to transfer belongs to. The I2SCH flag is refreshed at the moment when the TBE flag goes high. At the beginning, the I2SCH flag is low, indicating the left channel data should be written to the SPI_DATA register.

In order to disable I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

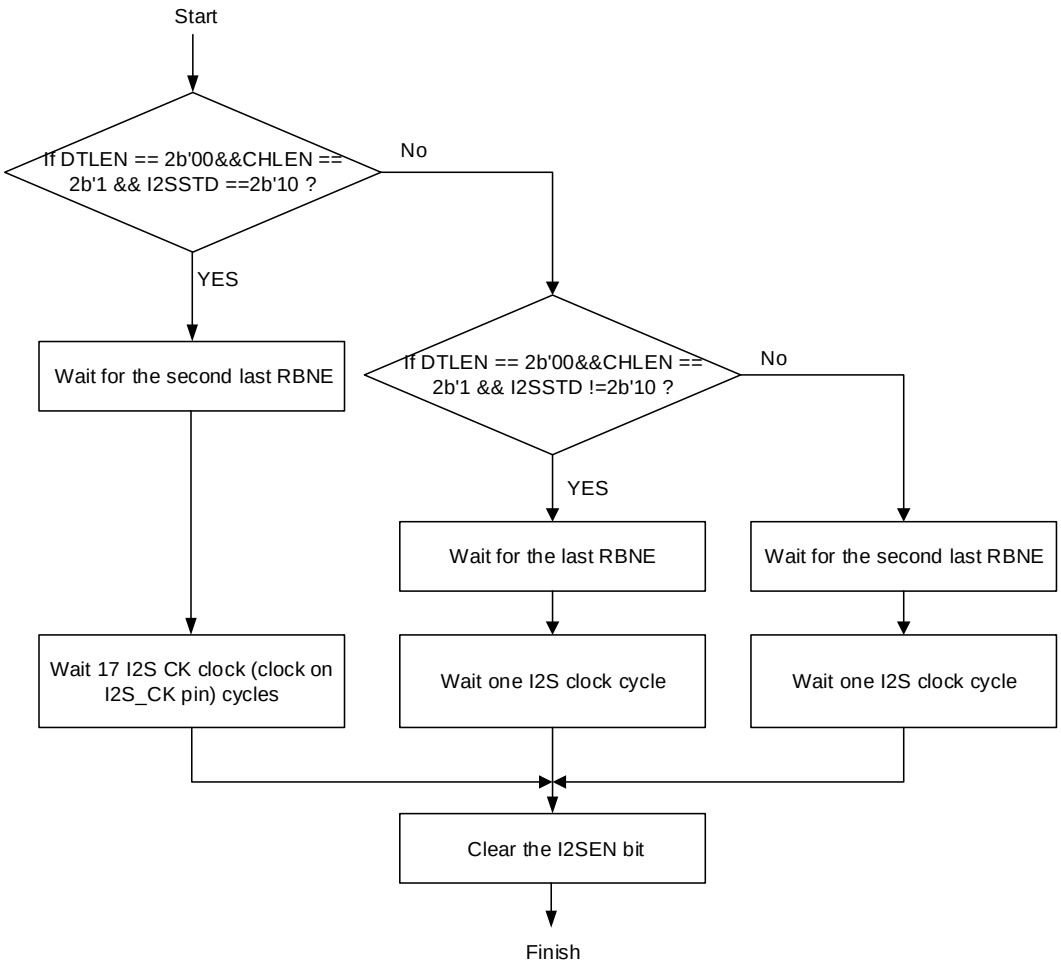
I2S master reception sequence

The RBNE flag is used to control the reception sequence. As is mentioned before, the RBNE flag indicates the receive buffer is not empty, and an interrupt will be generated if the RBNEIE bit in the SPI_CTL1 register is set. The reception sequence begins immediately when the I2SEN bit in the SPI_I2SCTL register is set. At the beginning, the receive buffer is empty (RBNE is low). When a reception sequence finishes, the received data in the shift register is loaded into the receive buffer (RBNE goes high). The data should be read from the SPI_DATA register, when the RBNE flag is high. After a read operation to the SPI_DATA register, the RBNE flag goes low. It is mandatory to read the SPI_DATA register before the end of the next reception. Otherwise, reception overrun error occurs. The RXORERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is necessary to disable and then enable I2S before resuming the communication.

For all standards except PCM, the I2SCH flag is used to distinguish which channel side the received data belongs to. The I2SCH flag is refreshed at the moment when the RBNE flag goes high.

Different sequences are used to disable the I2S in different standards, data length and channel length. The sequences for each case are described below ***Figure 21-53. I2S master reception disabling sequence***.

Figure 21-53. I2S master reception disabling sequence



I2S slave transmission sequence

The transmission sequence in slave mode is similar to that in master mode. The differences between them are described below.

In slave mode, the slave has to be enabled before the external master starts the communication. The transmission sequence begins when the external master sends the clock and when the I2S_WS signal requests the transfer of data. The data has to be written to the SPI_DATA register before the master initiates the communication. Software should write the next audio data into SPI_DATA register before the current data finishes. Otherwise, transmission underrun error occurs. The TXURERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is mandatory to disable and enable I2S to resume the communication. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to disable I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

I2S slave reception sequence

The reception sequence in slave mode is similar to that in master mode. The differences between them are described below.

In slave mode, the slave has to be enabled before the external master starts the communication. The reception sequence begins when the external master sends the clock and when the I2S_WS signal indicates a start of the data transfer. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to disable I2S, it is mandatory to clear the I2SEN bit immediately after receiving the last RBNE.

21.4.6. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

21.4.7. I2S interrupts

Status flags

There are four status flags implemented in the SPI_STAT register, including TBE, RBNE, TRANS and I2SCH. The user can use them to fully monitor the state of the I2S bus.

- **Transmit buffer empty flag (TBE)**

This bit is set when the transmit buffer is empty, the software can write the next data to the transmit buffer by writing the SPI_DATA register.

- **Receive buffer not empty flag (RBNE)**

This bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer, and software can read the data by reading the SPI_DATA register.

- **I2S transmitting ongoing flag (TRANS)**

TRANS is a status flag to indicate whether the transfer is ongoing or not. It is set and cleared by hardware and not controlled by software. This flag will not generate any interrupt.

- **I2S channel side flag (I2SCH)**

This flag indicates the channel side information of the current transfer and has no meaning in PCM mode. It is updated when TBE rises in transmission mode or RBNE rises in reception mode. This flag will not generate any interrupt.

Error conditions

There are three error flags:

- **Transmission underrun error flag (TXURERR)**

TXURERR will be set when the transmit buffer is empty and the valid SCK signal starts in slave transmission mode.

- **Reception overrun error flag (RXORERR)**

This situation occurs when the receive buffer is full and a newly incoming data has been completely received. When overrun occurs, the data in receive buffer is not updated and the newly incoming data is lost.

- **Format Error (FERR)**

In slave I2S mode, the I2S monitors the I2S_WS signal and an error flag will be set if I2S_WS toggles at an unexpected position.

I2S interrupt events and corresponding enable bits are summed up in the ***Table 21-10. I2S interrupt.***

Table 21-10. I2S interrupt

Interrupt flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer empty	Write SPI_DATA register	TBEIE
RBNE	Receive buffer not empty	Read SPI_DATA register	RBNEIE
TXURERR	Transmission underrun error	Read SPI_STAT register	ERRIE
RXORERR	Reception overrun error	Read SPI_DATA register and then read SPI_STAT register.	
FERR	I2S format error	Read SPI_STAT register	

21.5. Register definition

SPI0 base address: 0x4001 3000
SPI1/I2S1 base address: 0x4000 3800

21.5.1. Control register 0 (SPI_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

This register has no meaning in I2S mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSSEN	SWNSS	LF	SPIEN	PSC[2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The information transfers between the MOSI pin in master and the MISO pin in slave.
14	BDOEN	Bidirectional transmit output enable When BDEN is set, this bit determines the direction of transfer. 0: Work in receive-only mode 1: Work in transmit-only mode
13	CRCEN	CRC calculation enable 0: CRC calculation is disabled 1: CRC calculation is enabled
12	CRCNT	CRC next transfer 0: Next transfer is data 1: Next transfer is CRC value (TCRC) When the transfer is managed by DMA, CRC value is transferred by hardware. This bit should be cleared. In full-duplex or transmit-only mode, set this bit after the last data is written to SPI_DATA register. In receive-only mode, set this bit after the second last data is received.

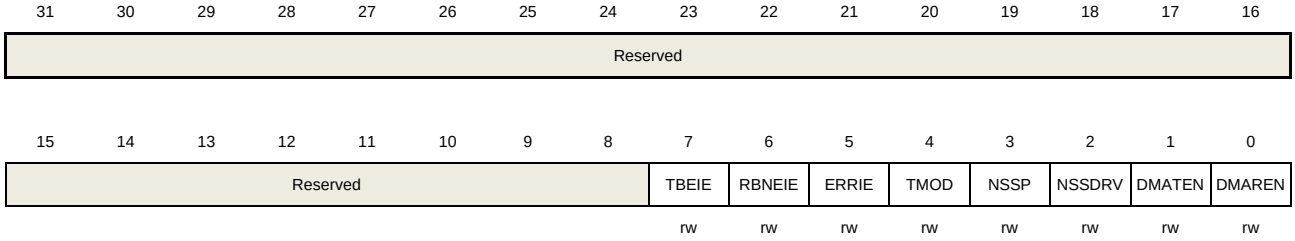
11	FF16	Data frame format 0: 8-bit data frame format 1: 16-bit data frame format
10	RO	Receive only mode When BDEN is cleared, this bit determines the direction of transfer. 0: Full-duplex mode 1: Receive-only mode
9	SWNSSEN	NSS software mode enable 0: NSS hardware mode. The NSS level depends on NSS pin. 1: NSS software mode. The NSS level depends on SWNSS bit. This bit has no meaning in SPI TI mode.
8	SWNSS	NSS pin selection in NSS software mode 0: NSS pin is pulled low 1: NSS pin is pulled high This bit effects only when the SWNSSEN bit is set. This bit has no meaning in SPI TI mode.
7	LF	LSB first mode 0: Transmit MSB first 1: Transmit LSB first This bit has no meaning in SPI TI mode.
6	SPIEN	SPI enable 0: SPI peripheral is disabled 1: SPI peripheral is enabled
5:3	PSC[2:0]	Master clock prescaler selection 000: PCLK/2 100: PCLK/32 001: PCLK/4 101: PCLK/64 010: PCLK/8 110: PCLK/128 011: PCLK/16 111: PCLK/256 PCLK means PCLK2 when using SPI0 or PCLK1 when using SPI1
2	MSTMOD	Master mode enable 0: Slave mode 1: Master mode
1	CKPL	Clock polarity selection 0: CLK pin is pulled low when SPI is idle 1: CLK pin is pulled high when SPI is idle
0	CKPH	Clock phase selection 0: Capture the first data at the first clock transition 1: Capture the first data at the second clock transition

21.5.2. Control register 1 (SPI_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TBEIE	Transmit buffer empty interrupt enable 0: TBE interrupt is disabled 1: TBE interrupt is enabled. An interrupt is generated when the TBE bit is set.
6	RBNEIE	Receive buffer not empty interrupt enable 0: RBNE interrupt is disabled 1: RBNE interrupt is enabled. An interrupt is generated when the RBNE bit is set.
5	ERRIE	Errors interrupt enable. 0: Error interrupt is disabled. 1: Error interrupt is enabled. An interrupt is generated when the CRCERR bit or the CONFERR bit or the RXORERR bit or the TXURERR bit is set.
4	TMOD	SPI TI mode enable 0: SPI TI Mode Disabled 1: SPI TI Mode Enabled
3	NSSP	SPI NSS pulse mode enable 0: SPI NSS Pulse Mode Disable 1: SPI NSS Pulse Mode Enable
2	NSSDRV	Drive NSS output 0: NSS output is disabled 1: NSS output is enabled If the NSS pin is configured as output, the NSS pin is pulled low in master mode when SPI is enabled. If the NSS pin is configured as input, the NSS pin should be pulled high in master mode, and this bit has no effect.
1	DMATEN	Transmit buffer DMA enable 0: Transmit buffer DMA is disabled 1: Transmit buffer DMA is enabled. When the TBE bit in SPI_STAT is set, it will

generate a DMA request at corresponding DMA channel.

0	DMAREN	Receive buffer DMA enable 0: Receive buffer DMA is disabled 1: Receive buffer DMA is enabled. When the RBNE bit in SPI_STAT is set, it will generate a DMA request at corresponding DMA channel.

21.5.3. Status register (SPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0002

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
							rc_w0	r	r	r	rc_w0	r	r	r	r

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	FERR	Format error SPI TI Mode: 0: No TI mode format error 1: TI mode format error occurs I2S Mode: 0: No I2S format error 1: I2S format error occurs This bit is set by hardware and cleared by writing 0.
7	TRANS	Transmitting ongoing bit 0: SPI or I2S is idle. 1: SPI or I2S is currently transmitting and/or receiving a frame This bit is set and cleared by hardware.
6	RXORERR	Reception overrun error bit 0: No reception overrun error occurs. 1: Reception overrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_DATA register followed by a read access to the SPI_STAT register.
5	CONFERR	SPI Configuration error 0: No configuration fault occurs. 1: Configuration fault occurred. (In master mode, the NSS pin is pulled low in NSS hardware mode or SWNSS bit is low in NSS software mode.)

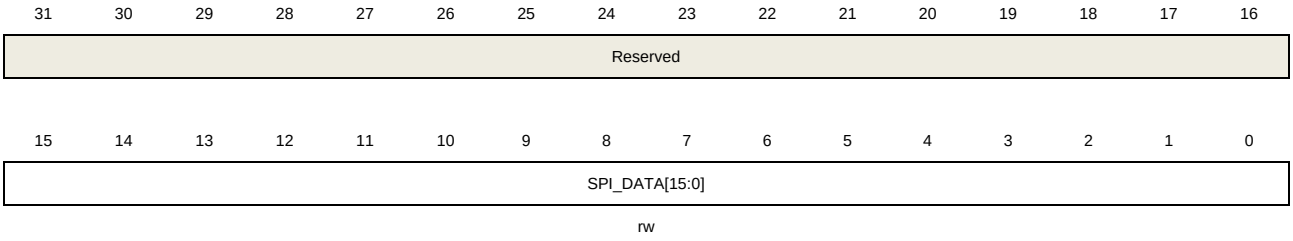
This bit is set by hardware and cleared by a read or write operation on the SPI_STAT register followed by a write access to the SPI_CTL0 register.
This bit is not used in I2S mode.

4	CRCERR	SPI CRC error bit 0: The SPI_RCRC value is equal to the received CRC data at last. 1: The SPI_RCRC value is not equal to the received CRC data at last. This bit is set by hardware and cleared by writing 0. This bit is not used in I2S mode.
3	TXURERR	Transmission underrun error bit 0: No transmission underrun error occurs. 1: Transmission underrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_STAT register. This bit is not used in SPI mode.
2	I2SCH	I2S channel side 0: The next data needs to be transmitted or the data just received is channel left. 1: The next data needs to be transmitted or the data just received is channel right. This bit is set and cleared by hardware. This bit is not used in SPI mode, and has no meaning in the I2S PCM mode.
1	TBE	Transmit buffer empty 0: Transmit buffer is not empty 1: Transmit buffer is empty
0	RBNE	Receive buffer not empty 0: Receive buffer is empty 1: Receive buffer is not empty

21.5.4. Data register (SPI_DATA)

Address offset: 0x0C
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	Data transfer register

The hardware has two buffers, including transmit buffer and receive buffer. Write data to SPI_DATA will save the data to transmit buffer and read data from SPI_DATA will get the data from receive buffer.

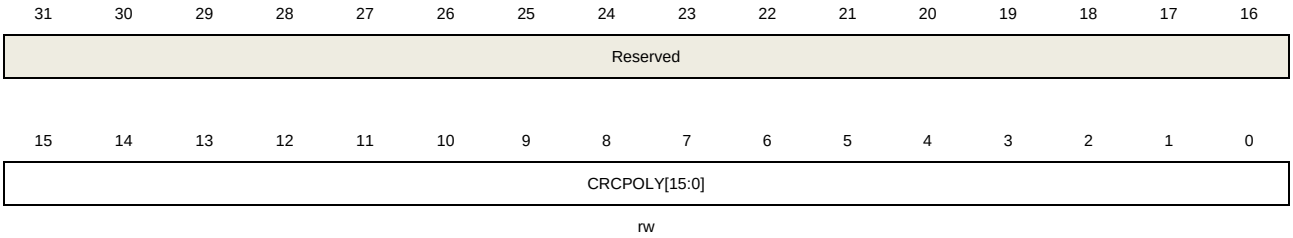
When the data frame format is set to 8-bit data, the SPI_DATA [15:8] is forced to 0 and the SPI_DATA[7:0] is used for transmission and reception, transmit buffer and receive buffer are 8-bit. If the data frame format is set to 16-bit data, the SPI_DATA[15:0] is used for transmission and reception, transmit buffer and receive buffer are 16-bit.

21.5.5. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

This register can be accessed by half-word (16-bit) or word (32-bit).



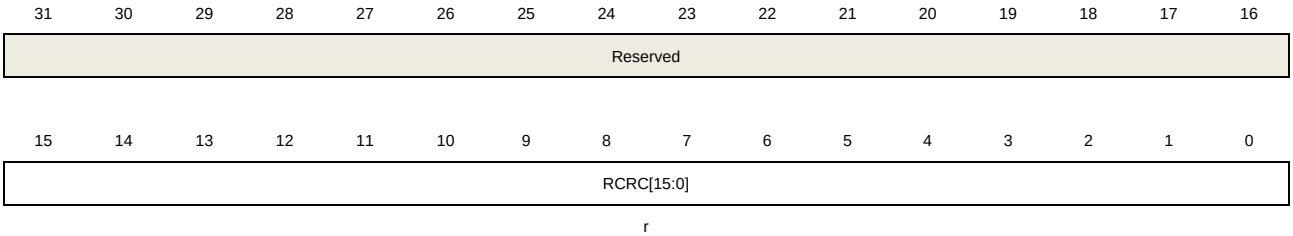
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRCPOLY[15:0]	CRC polynomial value These bits contain the CRC polynomial and they are used for CRC calculation. The default value is 0007h.

21.5.6. RX CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000

This register can be accessed by half-word (16-bit) or word (32-bit).



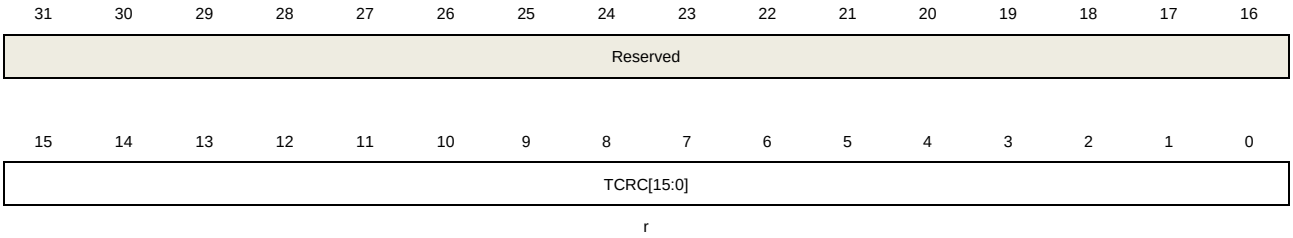
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15:0	RCRC[15:0]	RX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the received bytes and saves them in RCRC register. If the data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in RCRC[7:0], when the data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in RCRC[15:0]. The hardware computes the CRC value after each received bit, when the TRANS is set, a read to this register could return an intermediate value. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

21.5.7. TX CRC register (SPI_TCRC)

Address offset: 0x18
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	TX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the transmitted bytes and saves them in TCRC register. If the data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in TCRC[7:0], when the data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in TCRC[15:0]. The hardware computes the CRC value after each transmitted bit, when the TRANS is set, a read to this register could return an intermediate value. The different frame formats (LF bit of the SPI_CTL0) will get different CRC values. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

21.5.8. I2S control register (SPI_I2SCTL)

Address offset: 0x1C
Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	Reserved	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	I2SSEL	I2S mode selection 0: SPI mode 1: I2S mode This bit should be configured when SPI/I2S is disabled.
10	I2SEN	I2S enable 0: I2S is disabled 1: I2S is enabled This bit is not used in SPI mode.
9:8	I2SOPMOD[1:0]	I2S operation mode 00: Slave transmission mode 01: Slave reception mode 10: Master transmission mode 11: Master reception mode This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
7	PCMSMOD	PCM frame synchronization mode 0: Short frame synchronization 1: Long frame synchronization This bit has a meaning only when PCM standard is used. This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
6	Reserved	Must be kept at reset value.
5:4	I2SSTD[1:0]	I2S standard selection 00: I2S Philips standard 01: MSB justified standard 10: LSB justified standard 11: PCM standard These bits should be configured when I2S is disabled. These bits are not used in SPI mode.
3	CKPL	Idle state clock polarity 0: The idle state of I2S_CK is low level 1: The idle state of I2S_CK is high level

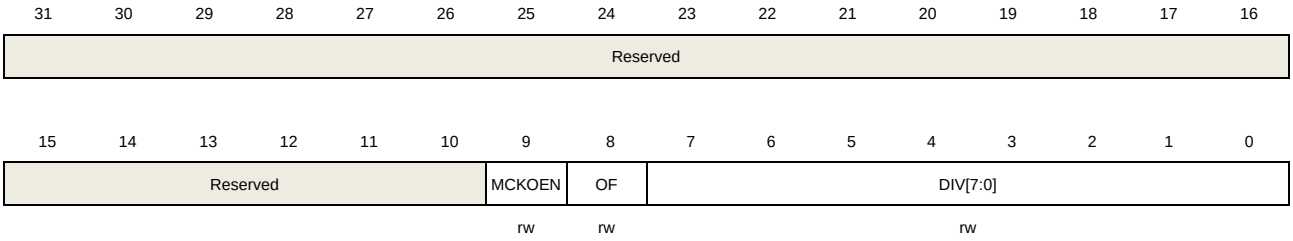
		This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
2:1	DTLEN[1:0]	Data length 00: 16 bits 01: 24 bits 10: 32 bits 11: Reserved These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
0	CHLEN	Channel length 0: 16 bits 1: 32 bits The channel length must be equal to or greater than the data length. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.

21.5.9. I2S clock prescaler register (SPI_I2SPSC)

Address offset: 0x20

Reset value: 0x0000 0002

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	MCKOEN	I2S_MCK output enable 0: I2S_MCK output is disabled 1: I2S_MCK output is enabled This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
8	OF	Odd factor for the prescaler 0: Real divider value is DIV * 2 1: Real divider value is DIV * 2 + 1 This bit should be configured when I2S is disabled. This bit is not used in SPI mode.
7:0	DIV[7:0]	Dividing factor for the prescaler

Real divider value is DIV * 2 + OF.

DIV must not be 0.

These bits should be configured when I2S is disabled.

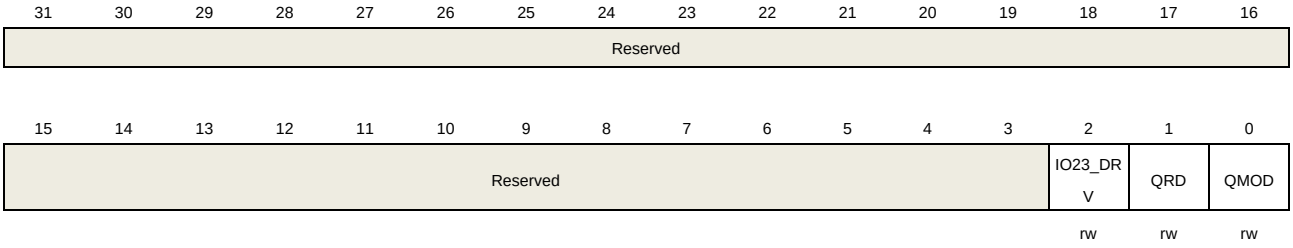
These bits are not used in SPI mode.

21.5.10. Quad-SPI mode control register (SPI_QCTL) of SPI0

Address offset: 0x80

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	IO23_DRV	Drive IO2 and IO3 enable 0: IO2 and IO3 are not driven in single wire mode 1: IO2 and IO3 are driven to high in single wire mode This bit is only available in SPI0.
1	QRD	Quad-SPI mode read select 0: SPI is in quad wire write mode 1: SPI is in quad wire read mode This bit should be only be configured when SPI is not busy (TRANS bit cleared). This bit is only available in SPI0.
0	QMOD	Quad-SPI mode enable 0: SPI is in single wire mode 1: SPI is in Quad-SPI mode This bit should only be configured when SPI is not busy (TRANS bit cleared). This bit is only available in SPI0.

22. Comparator (CMP)

22.1. Overview

The general purpose CMP can work either standalone (all terminal are available on I / Os) or together with the timers.

It can be used to wake up the MCU from low-power mode by an analog signal, provide a trigger source when an analog signal is in a certain condition, achieve some current control by working together with a PWM output of a timer and the DAC.

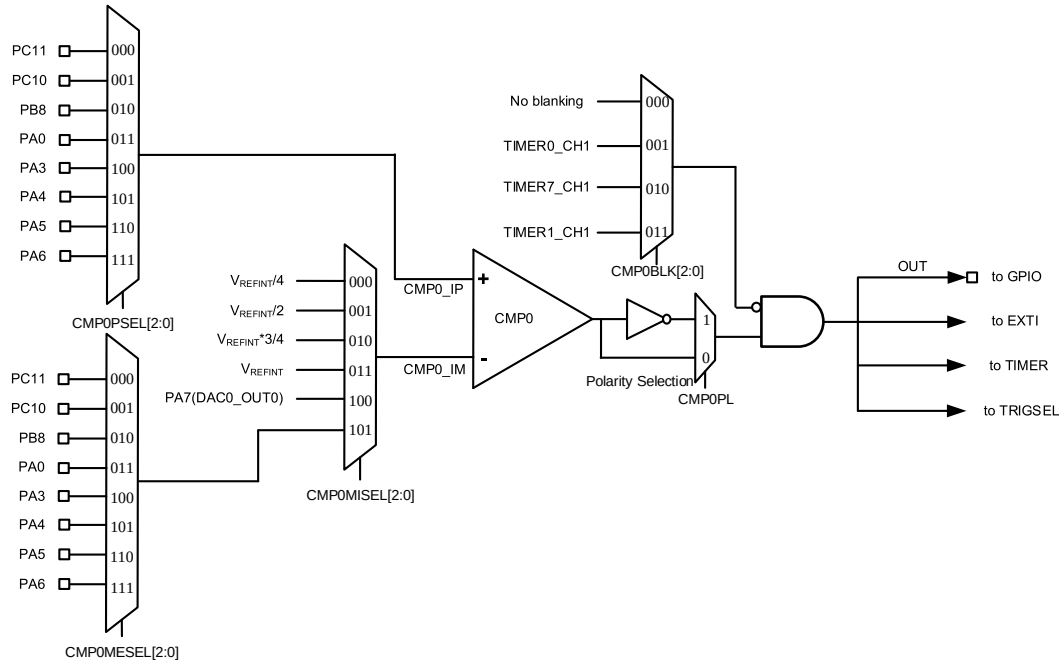
22.2. Characteristics

- Rail-to-rail comparators.
- Configurable hysteresis.
- Configurable speed and consumption.
- Configurable analog input source.
  - DAC output.
  - Multiplexed I / O pins.
  - The whole or sub-multiple values of internal reference voltage.
- Outputs with blanking source.
- Outputs to I / O.
- Outputs to timers for triggering.
- Outputs to EXTI.
- Outputs to TRIGSEL.

22.3. Function overview

The block diagram of CMP is shown below:

Figure 22-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

22.3.1. CMP clock

The clock of the CMP which is connected to APB bus, is synchronous with PCLK. It has private reset and clock enable bits.

22.3.2. CMP I / O configuration

These I / Os must be configured in analog mode in the GPIOs registers before they are selected as CMP inputs.

Refer to pin definitions in datasheet, and the CMP output can be connected to the corresponding I/O port via the alternate function of the GPIO.

The CMP output can be redirected internally and externally simultaneously.

CMP output internally connect to the TIMER and the connections between them are as follows:

- CMP output to the TIMER input channel.

In order to work even in Deep-sleep mode, the polarity selection logic and the output redirection to the port work independently from PCLK.

Table 22-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 22-1. CMP inputs and outputs summary

	CMP0
CMP non inverting	PA0

	CMP0
inputs connected to I / Os	PA3 PA4 PA5 PA6 PB8 PC10 PC11
CMP inverting inputs connected to I / Os	PA0 PA3 PA4 PA5 PA6 PB8 PC10 PC11
CMP inverting inputs connected to internal signals	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0
CMP outputs connected to I / Os	PB9 PF2
CMP outputs connected to EXTI	•
CMP outputs connected to TRIGSEL	•
CMP outputs connected to internal signals	TIMER0_CH0 TIMER7_CH0

22.3.3. CMP operating mode

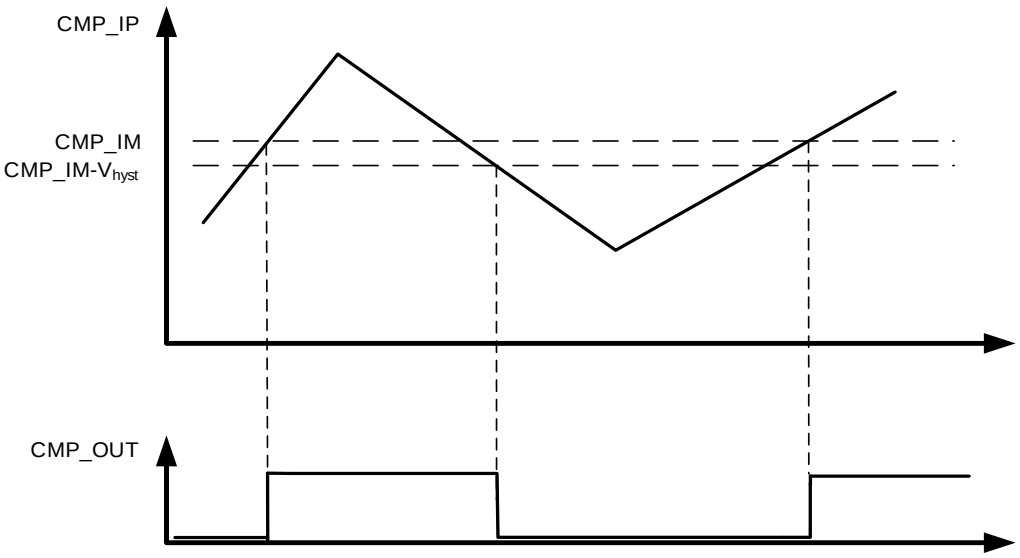
For a given application, there is a trade-off between the CMP power consumption versus propagation delay, which is adjusted by configuring bits CMPxM [1:0] in CMPx_CS register. The CMP works fastest with highest power consumption when CMPxM[1:0] = 2'b00, while works slowest with lowest power consumption when CMPxM[1:0] = 2'b11.

22.3.4. CMP hysteresis

In order to avoid spurious output transitions that caused by the noise signal, a programmable

hysteresis is designed to force the hysteresis value by configuring CMPx_CS register. This function could be shut down if it is unnecessary.

Figure 22-2. CMP hysteresis



22.3.5. CMP register write protection

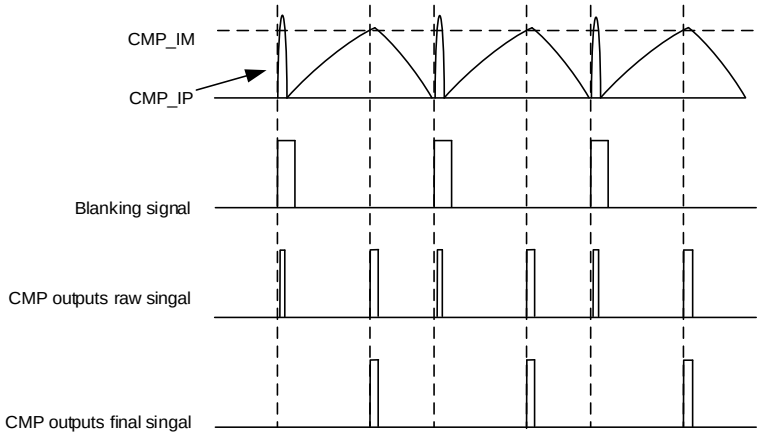
The CMP control and status register (CMPx_CS) can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the MCU reset.

22.3.6. CMP output blanking

CMP output blanking function can be used to avoid interference of short pulses in the input signal to CMP output signal. If the CMPxBLK[2:0] bits in the CMPx_CS register are setting to an available value, the CMP output final signal is obtained by ANDing the complementary signal of the selected blanking signal with the raw output of the comparator. The blanking function can be used for false overcurrent detection in motor control applications.

Figure 22-3. The CMP outputs signal blanking shows the comparator output blank function.

Figure 22-3. The CMP outputs signal blanking



22.3.7. CMP voltage scaler function

The voltage scaler function can provide selectable 1 / 4, 1 / 2, 3 / 4 reference voltage for CMP input. It is controlled by CMPxSEN and CMPxBEN bits in CMP control / status register. The CMPxSEN and CMPxBEN bits are used to enable the V_{REFINT} voltage output and the divider circuit, respectively, to generate the selected voltage.

22.3.8. CMP interrupt

The CMP output is connected to the EXTI and the EXTI line is exclusive to CMP. With this function, CMP can generate either interrupt or event which could be used to exit from low-power mode.

22.4. Register definition

CMP base address：0x4001 7C00

22.4.1. CMP Control/status register (CMPx_CS)

Address offset: 0x00
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP0LK	CMP0O	Reserved						CMP0SEN	CMP0BEN	Reserved	CMP0BLK[2:0]			CMP0HST[1:0]	
rwo	r							rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0PL	CMP0OSEL[3:0]		CMP0PSEL[2:0]			CMP0MISEL[2:0]			CMP0MESEL[2:0]			CMP0M[1:0]		Reserved	CMP0EN
rw	rw		rw			rw			rw			rw			rw

Bits	Fields	Descriptions
31	CMP0LK	CMP0 lock This bit can set all control bits of CMP0 as read-only. It can only be set once by software and cleared by a system reset. 0: CMPx_CS[31:0] bits are read-write 1: CMPx_CS[31:0] bits are read-only
30	CMP0O	CMP0 output state This bit is a copy of CMP output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
29:24	Reserved	Must be kept at reset value.
23	CMP0SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the CMP. 0: Disable bandgap scaler 1: Enable bandgap scaler
22	CMP0BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge 1: Enable scaler resistor bridge
21	Reserved	Must be kept at reset value.
20:18	CMP0BLK[2:0]	CMP0 output blanking source This bit is used to select which timer output controls the CMP0 output blanking. 000: No blanking

001: Select TIMER0_CH1 as blanking source
010: Select TIMER7_CH1 as blanking source
011: Select TIMER1_CH1 as blanking source
100~111: Reserved

17:16	CMP0HST[1:0]	CMP0 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
15	CMP0PL	Polarity of CMP0 output This bit is used to select the polarity of CMP0 output. 0: Output is not inverted 1: Output is inverted
14:13	CMP0OSEL[1:0]	CMP0 output selection These bits are used to select the destination of the CMP0 output. 00: no selection 01: TIMER0 CH0 input capture 10: TIMER7 CH0 input capture 11: Reserved Note: It is recommended to enable CMP first, and then configure the timer channel, when using TIMER to capture the output signal of the comparator.
12:10	CMP0PSEL[2:0]	CMP0_IP input selection These bits are used to select the source connected to the CMP0_IP input of the CMP. 000: PC11 001: PC10 010: PB8 011: PA0 100: PA3 101: PA4 110: PA5 111: PA6
9:7	CMP0MISEL[2:0]	CMP0_IM internal input selection These bits are used to select the internal source connected to the CMP0_IM input of the CMP. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT} * 3 / 4 011: V_{REFINT} 100: PA7(DAC0_OUT0) 101: Result of CMP0_IM external input selection

		110~111: Reserved
6:4	CMP0MESEL[2:0]	CMP0_IM external input selection These bits are used to select the external source connected to the CMP0_IM input of the CMP0. 000: PC11 001: PC10 010: PB8 011: PA0 100: PA3 101: PA4 110: PA5 111: PA6
3:2	CMP0M[1:0]	CMP0 mode These bits are used to control the operating mode of the CMP adjust the speed / consumption. 00: High speed / full power 01 / 10: Medium speed / medium power 11: Low speed / low power
1	Reserved	Must be kept at reset value.
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

23. Controller area network (CAN)

23.1. Overview

CAN bus (Controller Area Network) is a bus standard designed to allow microcontrollers and devices to communicate with each other without a host computer. The CAN interface supports the CAN 2.0A/B protocol, ISO 11898-1:2015 and BOSCH CAN FD specification.

The CAN module is a CAN Protocol controller with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system consists of a set of mailboxes that store configuration and control data, timestamp, message ID, and data. The space of up to 32 mailboxes can also be configured as Rx FIFO with ID filtering against up to 104 extended IDs or 208 standard IDs or 416 partial 8-bit IDs, and configure receive FIFO/mailbox private filter register for up to 32 ID filter table elements.

23.2. Characteristics

- Supports CAN protocol version 2.0A/B.
- Compliant with the ISO 11898-1:2015 standard.
- Supports CAN FD frame with up to 64 data bytes, baudrate up to 8 Mbit/s.
- Supports CAN classical frame with up to 8 data bytes, baudrate up to 1 Mbit/s.
- Supports time stamp based on 16-bit free running counter.
- Supports transmitter delay compensation for CAN FD frames at faster data rates.
- Maskable interrupts.
- Supports four communication mode: normal mode, Inactive mode, Loopback and silent mode, and Monitor mode.
- Supports two power saving modes: CAN_Disable mode, and Pretended Networking mode.
- Support two wakeup methods for waking up from Pretended Networking mode: wakeup matching event, and wakup timeout event.
- 32 mailboxes when configures with 8 bytes data length each, configurable as Rx or Tx mailbox.
- Global network time, synchronized by a specific message.
- The clock source of Protocol controller (CANCLK) is configurable(1).

Transmission

- Supports transmission abort.
- Tx mailbox status checkable.
- CRC for transmitted message.
- Supports priority of transmission message: lowest mailbox number, or highest priority.

Reception

- Receive private filter registers per Rx mailbox or Rx FIFO.
- Receive public filter register for Rx mailboxes and receive public filter register for Rx

FIFO.

- Supports priority of message reception between mailboxes and Rx FIFO during matching process.
- Rx FIFO identifier filtering, supports identifier matching against either 104 extended, 208 standard, or 416 partial (8 bit) identifiers.
- Rx FIFO up to 6 frames depth, with DMA support.

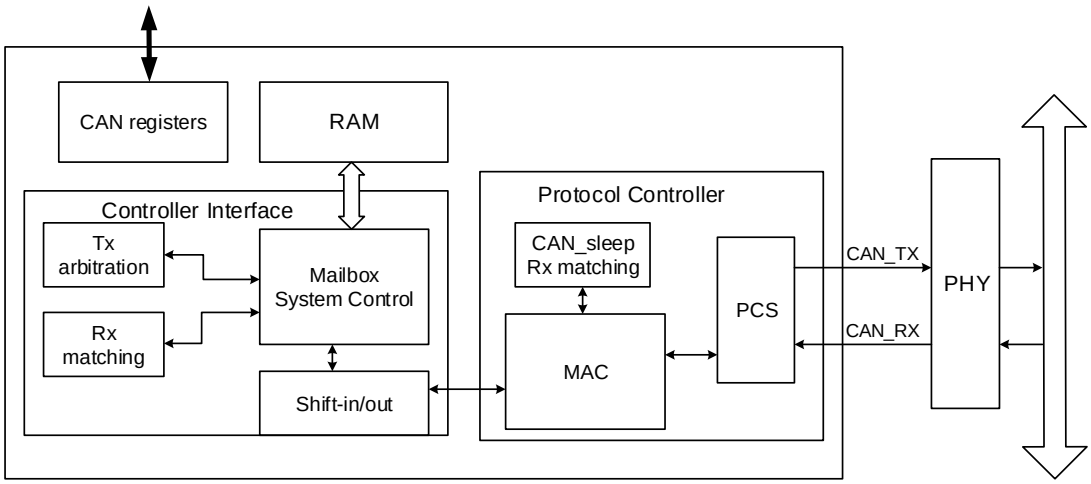
Note:

(1) The CANCLK source selection has restrictions. For details, please refer to the errata manual of the GD32A50x series.

23.3. Function overview

Figure 23-1. CAN module block diagram shows the CAN block diagram.

Figure 23-1. CAN module block diagram



As shown in *Figure 23-1. CAN module block diagram*, CAN module includes three main parts:

- The Protocol controller

  The Protocol controller manages the communication on the CAN bus, including:

  MAC (Media Access Control):

  - Bit-stuffing/de-stuffing.
  - Stuff bit count for FD Frames.
  - Add CRC.
  - Construction of MAC frame.
  - ACK check/transmission.

  PCS (Physical Coding Sub-layer):

  - Bit timing.
  - Synchronization.
  - TDC (Transmitter delay compensation).

Pretended Networking Rx matching:

- Process reception matching in Pretended Networking mode.

■ The Controller Interface

The Controller Interface manages RAM space selection for reception and transmission, including:

Tx arbitration:

- Find out the frame with the highest priority.

Rx matching:

- Compare the frame data received in the Rx shift buffer (an internal mailbox descriptor) with the fields in Rx mailbox or Rx FIFO according to the configured matching order.

Mailbox System Controller:

- Manage RAM space selection for reception and transmission, control the mailbox CODE, control the Rx FIFO pointer, and control the access requirement from the APB bus to the RAM space.

  The messages are stored in an embedded RAM dedicated to the CAN module. The dedicated RAM base address is module base address.

Shift in/out:

- Tranmit data between the selected mailbox / Rx FIFO descriptor and the Tx or Rx shift buffer.

■ CAN registers

The CAN registers is responsible for the CAN module communication with the APB bus.

23.3.1. Mailbox descriptor

The mailbox descriptor shown in ***Table 23-1. Mailbox descriptor with 64 byte payload*** can be used for both extended (29-bit identifier) and standard (11-bit identifier) frames. Each mailbox is formed by 16, 24, 40, or 72 bytes, depending on the data bytes allocation for the message payload: 8, 16, 32, or 64 data bytes, respectively. The memory area from offset 0x80 to 0x27F is used by the mailboxes.

Table 23-1. Mailbox descriptor with 64 byte payload

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDES0	FDF	BRS	ESI	Reserved	CODE[3:0]				Reserved	SRR	IDE	RTR	DLC[3:0]				TIMESTAMP[15:0]															
MDES1	PRIO[2:0]			ID_STD[10:0]													ID_EXD[17:0]															
MDES2	DATA_0[7:0]								DATA_1[7:0]								DATA_2[7:0]								DATA_3[7:0]							
...	...								...								...								...							
MDES17	DATA_60[7:0]								DATA_61[7:0]								DATA_62[7:0]								DATA_63[7:0]							

MDES0: Mailbox descriptor word 0

Address offset: 0x80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

FDF	BRS	ESI	Reserved	CODE[3:0]	Reserved	SRR	IDE	RTR	DLC[3:0]
rw	rw	rw		rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP[15:0]															
r															

Bits	Fields	Descriptions
31	FDF	FD format indicator This bit is used to distinguish between CAN and CAN FD format frames. For reception (Rx mailbox), no need to write this bit, it will be stored with the value received on the CAN bus.
30	BRS	Bit rate switch This bit defines whether the bit rate is switched for a CAN FD frame. For reception (Rx mailbox), no need to write this bit, it will be stored with the value received on the CAN bus.
29	ESI	Error state indicator This bit indicates if the transmitting node is error active or error passive. This bit does not exist in Classical frames. For transmission (Tx mailbox), it is transmitted dominant by error active nodes, and recessive by error passive nodes. For reception (Rx mailbox), no need to write this bit, it will be stored with the value received on the CAN bus.
28	Reserved	Must be kept at rest value.
27:24	CODE[3:0]	Mailbox Code (CODE) This bit field can be accessed by the CPU and by the CAN module, as part of the mailbox matching and arbitration process. The encoding is shown in ***Table 23-3. Mailbox Rx CODE*** and ***Table 23-4. Mailbox Tx CODE***.
23	Reserved	Must be kept at rest value.
22	SRR	Substitute remote request This bit is only used in extended format. For transmission (Tx mailbox), it must be set to ‘1’ (recessive), if the bus transmits this bit as ‘0’ (dominant), then it means an arbitration loss. For reception (Rx mailbox), it will be stored with the value received on the CAN bus. 0: Not valid for transmission in extended format frames. 1: Transmission in extended format frames.
21	IDE	ID extended bit This bit specifies whether the frame is standard or extended format. For reception (Rx mailbox), it will be stored with the value received on the CAN bus. 0: Frame format is standard.

1: Frame format is extended.

20 RTR

Remote transmission request

For transmission (Tx mailbox), if this bit is set to '1' (recessive), and the bus transmits this bit as '0' (dominant), then it means an arbitration loss. If this bit is set to '0' (dominant), and the bus transmits this bit as '1' (recessive), it is treated as a bit error. If the value configured matches the value transmitted, it is considered a successful bit transmission.

For reception (Rx mailbox), it will be stored with the value received on the CAN bus.

0: In Tx mailbox, the current mailbox has a data frame to be transmitted. In Rx mailbox, it may be considered in matching process.

1: In Tx mailbox, it means the current mailbox has a remote request frame to be transmitted. In Rx mailbox, incoming remote request frames may be stored.

Note: When configured as CAN FD frames, the RTR bit must be negated. This bit must be considered in classical frames only.

19:16 DLC[3:0]

Data length code in bytes

This bit field is the length (in bytes) of the Rx or Tx payload.

For reception (Rx mailbox), no need to write this bit field, they are written by the CAN module with the DLC field of the received frame.

For transmission (Tx mailbox), this bit field is written by the CPU with value of the frame to be transmitted. When RTR is 1, the frame to be transmitted is a remote request frame and does not include the data field, regardless of the DLC field.

15:0 TIMESTAMP[15:0]

Free-Running counter timestamp

This bit field is a copy of the free running counter, captured for Tx and Rx frames at the time when the beginning of the ID field appears on the CAN bus.

Table 23-2. Data bytes for DLC

DLC	Data size in bytes
i (0 ≤ i ≤ 8)	i (0 ≤ i ≤ 8)
9	12
10	16
11	20
12	24
13	32
14	48
15	64

Table 23-3. Mailbox Rx CODE

CODE	Meaning	CODE after reception	Serviced(1)	RRFRMS(2)	Description
0b0000	INACTIVE	-	-	-	Mailbox does not participate in the matching process.
0b0100	EMPTY	FULL	-	-	When a frame is received successfully (after the Move-

CODE	Meaning	CODE after reception	Serviced(1)	RRFRMS(2)	Description
					in process), the CODE field is automatically updated to FULL.
0b0010	FULL	FULL	Yes	-	It remains FULL. If a new frame is moved to the mailbox after the mailbox was serviced, the code still remains FULL.
		OVERRUN	No		If the mailbox is FULL and a new frame is moved to this mailbox before the CPU services it, the CODE field is automatically updated to OVERRUN.
0b0110	OVERRUN	FULL	Yes	-	If the CODE field indicates OVERRUN and CPU has serviced the mailbox, when a new frame is moved to the mailbox then the code returns to FULL.
		OVERRUN	No		If the CODE field already indicates OVERRUN, and another new frame must be moved, the mailbox will be overwritten again, and the code will remain OVERRUN.
0b1010	RANSWER(3)	TANSWER (0x1110)	-	0	A Remote Answer was configured to recognize a remote request frame reception. After reception, the mailbox is set to transmit a response frame when RRFRMS bit in CAN_CTL2 register is 0. The code is automatically changed to TANSWER.
		-		1	This code is ignored during matching and arbitration process.
CODE[0] = 1	BUSY(4)	FULL	-	-	Indicates that the mailbox is being updated.
		OVERRUN			

1. Serviced: Mailbox was serviced by CPU read, and was unlocked by reading CAN_TIMER register or other mailbox.
2. Remote Request Frame Stored bit, refer to ***Control register 2 (CAN_CTL2)***.
3. A mailbox with CODE 0b1010 should not be aborted. CODE 0b1010 must be used in mailbox which configured as CAN classical format, having the FDF bit reset.
4. If CODE[0] bit is set, the corresponding mailbox will not participate in the matching process. Notice that for Tx mailboxes, the BUSY bit should be ignored when read, except when MST bit in the CAN_CTL0 register is set.

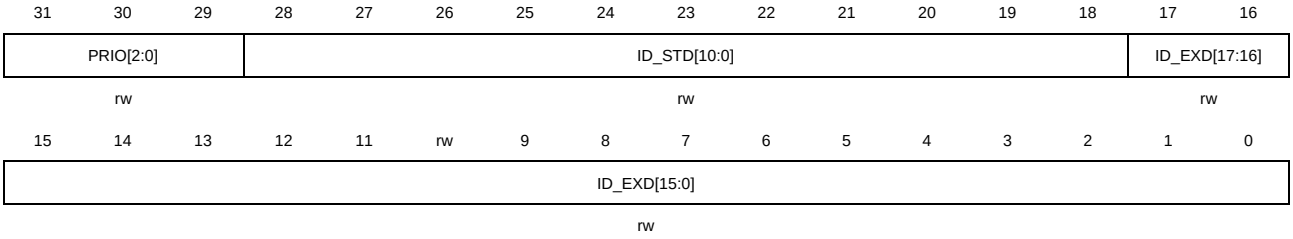
Table 23-4. Mailbox Tx CODE

CODE	Meaning	CODE after transmission	RTR	Description
0b1000	INACTIVE	-	-	Mailbox does not participate in arbitration process.
0b1001	ABORT	-	-	Mailbox does not participate in arbitration process.
0b1100	DATA	INACTIVE	0	Transmit data frame. After transmission, the mailbox automatically returns to the INACTIVE state.

CODE	Meaning	CODE after transmission	RTR	Description
	REMOTE	EMPTY	1	Transmit remote request frame. After transmission, the mailbox automatically becomes an Rx empty mailbox with the same ID.
0b1110	TANSWER	RANSWER	-	This is an intermediate code which is automatically written to the mailbox by the controller interface when a matching remote request frame is received. After transmitting the remote response frame, the mailbox will automatically return to RANSWER state. The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit.

MDES1: Mailbox descriptor word 1

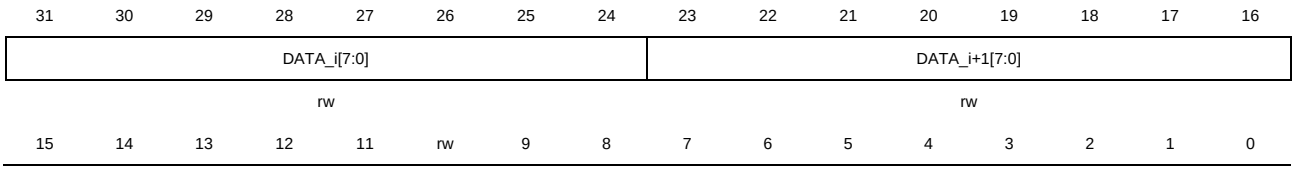
Address offset: 0x84



Bits	Fields	Descriptions
31:29	PRIO[2:0]	Local priority This bit field is only used when LAPRIOEN bit in CAN_CTL0 register is set. This bit filed is only used for Tx mailboxes, while these bits are not transmitted, they are appended to the regular ID to define the transmission priority.
28:18	ID_STD[10:0]	Identifier for standard frame In standard frame format, only these 11 most significant bits (28 to 18) are used for frame identification in both reception and transmission cases. The 18 least significant bits are ignored.
17:0	ID_EXD[17:0]	Identifier for extended frame In extended frame format, ID_STD[10:0] & these bits are used for frame identification in both reception and transmission cases.

MDESx: Mailbox descriptor word x (x = 2..17)

Address offset: 0x80 + 0x04 * x (x = 2..17)



DATA_i+2[7:0]	DATA_i+3[7:0]
rw	rw

Bits	Fields	Descriptions
31:24	DATA_i[7:0]	Data byte i (i = 4*x - 8) Refer to DATA_i+3[7:0] descriptions.
23:16	DATA_i+1[7:0]	Data byte i+1 (i = 4*x - 8) Refer to DATA_i+3[7:0] descriptions.
15:8	DATA_i+2[7:0]	Data byte i+2 (i = 4*x - 8) Refer to DATA_i+3[7:0] descriptions.
7:0	DATA_i+3[7:0]	Data byte i+3 (i = 4*x - 8) Up to 64 bytes can be used for a data frame, depending on the DLC value of the mailbox. For Rx frames, the data received from the CAN bus are stored in this bit field.

Mailbox number

When Rx FIFO is disabled, the dedicated RAM space is occupied by mailboxes only, so the mailbox number is the descriptor number which is incremented by one each time when across the complete mailbox descriptor length (with 8, 16, 32, or 64 data bytes).

When Rx FIFO is enabled (CAN FD mode disabled, data field is 8-byte length), the dedicated RAM space is occupied by both mailboxes and FIFO, so uniformly count the descriptor number by a mailbox descriptor length with 8 data bytes, then the mailbox number is the descriptor number which is occupied by mailbox.

Mailbox size for CAN FD

When CAN FD is enabled, the size of mailboxes that the CAN 512 bytes RAM can be partitioned is configured by MDSZ[1:0] bits in CAN_FDCTL register.

Table 23-5. Mailbox size

MDSZ[1:0]	Payload size in bytes	Mailbox size
0b00	8	32
0b01	16	21
0b10	32	12
0b11	64	7

23.3.2. Rx FIFO descriptor

The Rx FIFO descriptor is shown in ***Table 23-6. Rx FIFO*** .

When RFEN bit in CAN_CTL0 register is 1, the RAM space which normally occupied by mailbox 0–5 with 8 byte payload is used for the Rx FIFO. FDES0 – FDES3 contains the output

of the FIFO which is the oldest message that has been received but not yet read by the CPU. The RAM region 0x90-0xDF is reserved for internal use of the FIFO.

When RFEN bit in CAN_CTL0 register is 1, the RAM space which normally occupied by mailbox 6–31 with 8 byte payload is used for the ID filter table (configurable for 8 to up to 104 table elements) for receiving frames matching process into the FIFO. The ID filter table only contains 8 elements from FDES4 to FDES11 by default.

Table 23-6. Rx FIFO descriptor

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDES0	IDFMN[8:0]									SR R	IDE	RT R	DLC[3:0]				TIMESTAMP[15:0]															
FDES1	Reserved									ID_STD[10:0]							ID_EXD[17:0]															
FDES2	DATA_0[7:0]									DATA_1[7:0]							DATA_2[7:0]									DATA_3[7:0]					
FDES3	DATA_4[7:0]									DATA_5[7:0]							DATA_6[7:0]									DATA_7[7:0]					
0x90 - 0xDC	Reserved																															
FDES4	ID filter table element 0																															
…	…																															
FDES107	ID filter table element 103																															

FDES0: Rx FIFO descriptor word 0

Address offset: 0x80

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDFMN[8:0]									SRR	IDE	RTR	DLC[3:0]			
r									r	r	r	r			

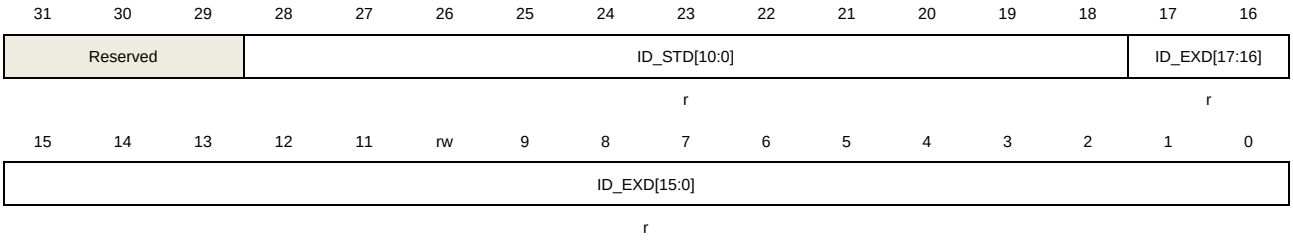
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP[15:0]															
r															

Bits	Fields	Descriptions
31:23	IDFMN[8:0]	Identifier filter matching number This bit field indicates which ID filter table element matches the received message that is in the output of the Rx FIFO.
22	SRR	Substitute remote request This bit is only used in extended format. It will be stored with the value received on the CAN bus.
21	IDE	ID extended bit This bit specifies whether the frame is standard or extended format. 0: Frame format is standard. 1: Frame format is extended.

20	RTR	Remote transmission request 0: Data frames are accepted 1: Remote frames are accepted
19:16	DLC[3:0]	Data length code in bytes This bit field is the length (in bytes) of the Rx payload. For reception, this bit field is written by the CAN module with the DLC field of the received frame.
15:0	TIMESTAMP[15:0]	Free-Running counter timestamp This bit field is a copy of the free running counter, captured for Rx frames at the time when the beginning of the ID field appears on the CAN bus.

FDES1: Rx FIFO descriptor word 1

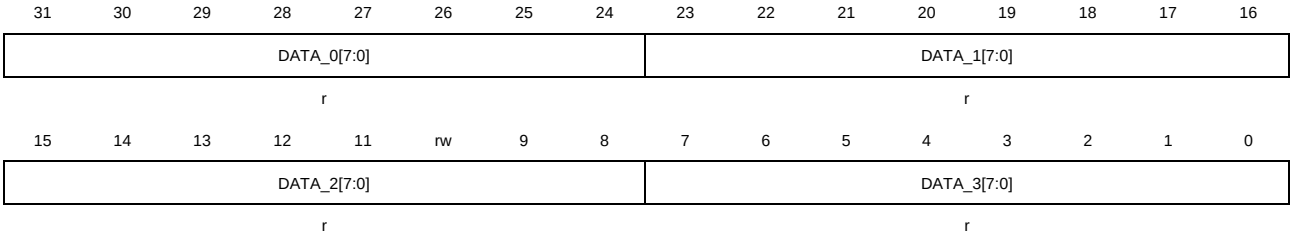
Address offset: 0x84



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at rest value.
28:18	ID_STD[10:0]	Identifier for standard frame In standard frame format, only these 11 most significant bits (28 to 18) are used for frame identification. The 18 least significant bits are ignored.
17:0	ID_EXD[17:0]	Identifier for extended frame In extended frame format, ID_STD[10:0] & these bits are used for frame identification.

FDES2: Rx FIFO descriptor word 2

Address offset: 0x88



Bits	Fields	Descriptions
31:24	DATA_0[7:0]	Data byte 0

		Refer to DATA_3[7:0] descriptions.
23:16	DATA_1[7:0]	Data byte 1 Refer to DATA_3[7:0] descriptions.
15:8	DATA_2[7:0]	Data byte 2 Refer to DATA_3[7:0] descriptions.
7:0	DATA_3[7:0]	Data byte 3 Up to 8 bytes can be used for a data frame, depending on the DLC value of the mailbox. FD frames is not supported to receive in Rx FIFO.

FDES3: Rx FIFO descriptor word 3

Address offset: 0x8C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA_4[7:0]								DATA_5[7:0]							
r								r							
15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
DATA_6[7:0]								DATA_7[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DATA_4[7:0]	Data byte 4 Refer to DATA_7[7:0] descriptions.
23:16	DATA_5[7:0]	Data byte 5 Refer to DATA_7[7:0] descriptions.
15:8	DATA_6[7:0]	Data byte 6 Refer to DATA_7[7:0] descriptions.
7:0	DATA_7[7:0]	Data byte 7 Up to 8 bytes can be used for a data frame, depending on the DLC value of the mailbox. FD frames is not supported to receive in Rx FIFO.

FDESx: Rx FIFO descriptor word x (x = 4..107)

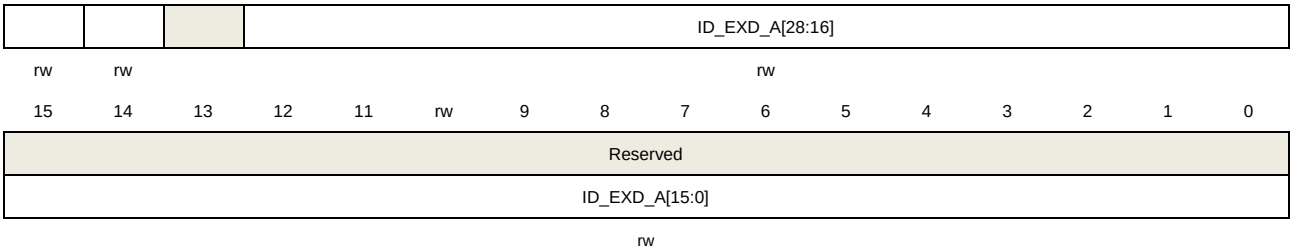
Address offset: 0xE0 + 4 * (x - 4)

This descriptor word shows the three different formats of the ID filter table elements, depending on the configuration of FS[1:0] bits in CAN_CTL0 register.

Note: The format is applied to all ID filter table elements. It is not possible to mix formats within the table.

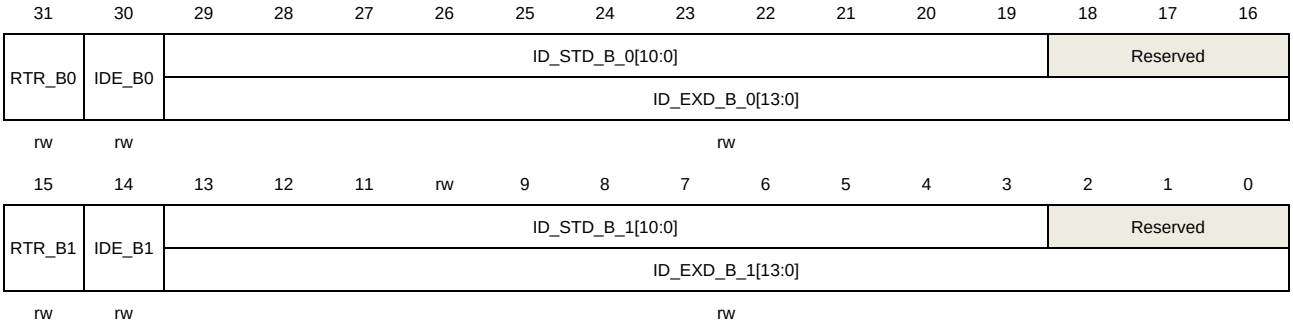
Format A mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTR_A	IDE_A	Reserved	ID_STD_A[10:0]											Reserved	



Bits	Fields	Descriptions
31	RTR_A	Remote frame for format A This bit specifies whether remote frames can be stored into the FIFO or not when matches. 0: It indicates that remote frames are rejected and data frames can be stored. 1: It indicates that remote frames can be stored and data frames are rejected.
30	IDE_A	ID Extended frame for format A This bit specifies whether extended frames can be stored into the FIFO or not when matches. 0: Extended frames are rejected and standard frames can be stored. 1: Extended frames can be stored and standard frames are rejected.
29	Reserved	Must be kept at rest value.
28:1	ID_STD_A[10:0]/ ID_EXD_A[28:0]	ID in format A This bit field specifies one full standard ID (standard or extended) for Rx FIFO matching process. If IDE_A is 0, the 18 to 28 bits are used for standard ID, and the rest bits are reserved; otherwise, all these bits are used for extended ID.

Format B mode:



Bits	Fields	Descriptions
31	RTR_B0	Remote frame 0 for format B This bit specifies whether remote frames can be stored into the FIFO or not when matches. 0: It indicates that remote frames are rejected and data frames can be stored. 1: It indicates that remote frames can be stored and data frames are rejected.
30	IDE_B0	ID Extended frame 0 for format B

		This bit specifies whether extended frames can be stored into the FIFO or not when matches. 0: Extended frames are rejected and standard frames can be stored. 1: Extended frames can be stored and standard frames are rejected.
29:16	ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0]	ID for frame 0 in format B This bit field specifies a full standard ID or partial 14-bit extended ID for Rx FIFO matching process. If IDE_B0 is 0, the 19 to 29 bits are used for standard ID, and the rest bits are reserved; otherwise, these bits are used for partial 14-bit extended ID, compared with the 14 most significant bits of the received ID.
15	RTR_B1	Remote frame 1 for format B Refer to RTR_B0 descriptions.
14	IDE_B1	ID Extended frame 1 for format B Refer to IDE_B0 descriptions.
13:0	ID_STD_B_1[10:0]/ ID_EXD_B_1[13:0]	ID for frame 1 in format B Refer to ID_STD_B_0[10:0]/ ID_EXD_B_0[13:0] descriptions.

Format C mode:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID_C_0[7:0]								ID_C_1[7:0]							
rw								rw							

15	14	13	12	11	rw	9	8	7	6	5	4	3	2	1	0
ID_C_2[7:0]								ID_C_3[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	ID_C_0[7:0]	ID for frame 0 in format C This bit field specifies a partial 8-bit standard ID or partial 8-bit extended ID for Rx FIFO matching process. In both standard and extended frame formats, the 8 bit field is compared with the 8 most significant bits of the received ID.
23:16	ID_C_1[7:0]	ID for frame 1 in format C Refer to ID_C_0[7:0] descriptions.
15:8	ID_C_2[7:0]	ID for frame 2 in format C Refer to ID_C_0[7:0] descriptions.
7:0	ID_C_3[7:0]	ID for frame 3 in format C Refer to ID_C_0[7:0] descriptions.

23.3.3. Communication modes

The CAN interface has four communication modes:

- Normal mode
- Inactive mode
- Loopback and silent mode
- Monitor mode

Normal mode

In normal mode, the message reception and transmission, and errors are all managed normally, and all CAN protocol functions are enabled.

Inactive mode

To enter Inactive mode, set INAMOD bit in CAN_CTL0 to 1 to enable Inactive mode, then set HALT bit in CAN_CTL0 register to 1 or put the chip into Debug mode.

When Inactive mode is requested, the following steps are performed before INAS bit asserted:

1. Wait for the bus 11 consecutive recessive bits.
2. Wait for the current transmission or reception processes being finished, it means all internal activities such as arbitration, matching, shift-in, and shift-out being finished. A pending shift-in does not prevent entering Inactive mode.
3. The Tx pin is driven as '1' (recessive).
4. Stop the prescaler.
5. Enable write access to the CAN_ERR0 register, which is read-only in other modes.
6. Set NRDY bit and INAS bit in CAN_CTL0 register.

When Inactive mode is entered, INAS bit in CAN_CTL0 register is set to 1 by CAN.

In Inactive mode, neither transmission nor reception is performed, and its prescaler is stopped, all registers are accessible.

To exit from Inactive mode, one of the following methods can meet:

- Clear INAMOD bit in CAN_CTL0.
- Clear HALT bit in CAN_CTL0 register, or the chip is removed from Debug mode.

If exiting from Inactive mode is requested, then INAS bit in CAN_CTL0 register is cleared after the CAN prescaler is running again. When out of Inactive mode, CAN module tries to resynchronize to the CAN bus by waiting for 11 consecutive recessive bits.

Note: When in Inactive mode, the CAN_Disable mode request will lead to INAS bit in CAN_CTL0 register be cleared and LPS bit in CAN_CTL0 register be set.

Loopback and silent mode

To enter this mode, set the LSCMOD bit in CAN_CTL1 register to 1. In this mode, the

messages are internally transmitted back to the receiver input, and the bit sent during the ACK slot in the frame acknowledge field is ignored to ensure reception transmitted by itself. Both transmit and receive interrupts are generated.

Loopback and silent mode is used for self-test. The Rx pin is ignored and the Tx pin holds in recessive state.

Monitor mode

To enter this mode, set MMOD bit in CAN_CTL1 register to 1.

When Monitor mode is entered, ERRSI[1:0] bit field in CAN_ERR1 register is set to 0b01 by CAN to indicate that the module works in an Error Passive state. In this mode, error counters are frozen for transmission and reception.

In Monitor mode, no transmission is performed and reception is performed only when messages are acknowledged by other CAN nodes, detection of a message that has not been acknowledged will lead to a bit dominant error flag (without changing the RECNT[7:0] or REFCNT[7:0] in CAN_ERR0 register).

23.3.4. Power saving modes

The CAN interface has two power saving modes:

- CAN_Disable mode.
- Pretended Networking mode.

In these two power saving modes, the dedicated RAM and the registers in SRAM can not be accessed.

CAN_Disable mode

The CAN module is enabled or disabled by configuring the CANDIS bit in CAN_CTL0 register.

For power saving, if CANDIS bit is set to 1 to disable CAN module, the CAN module will enter CAN_Disable mode after a delay when the LPS bit and NRDY bit in CAN_CTL0 register are changed to 1.

When CAN is disabled, the clocks to the Protocol controller and the Controller Interface are disabled. All registers except the CAN_RMPUBF, CAN_RFIFOPUBF, CAN_RFIFOIFMN, and CAN_RFIFOMPFx (x=0..31) registers are accessible. Also the dedicated RAM can not be accessed.

After CAN is enabled, you need to dalay a time to wait for LPS bit in CAN_CTL0 register to be cleared for Protocol controller recognization. When CAN is enabled, CAN module requests to resume the clocks to the Protocol controller and the Controller Interface.

Pretended Networking mode

Pretended Networking mode is used to receive wakeup messages with low power

consumption. This mode can work together with MCU deepsleep mode.

To enter Pretended Networking mode, set PNEN bit and PNMOD bit in CAN_CTL0 register to 1, and optionally put the MCU into deepsleep mode.

When Pretended Networking mode is requested, the following steps are performed:

1. Wait for the bus to be in Idle state, or else wait for the third bit of Intermission, and then checks it to be recessive.
2. Set PNS bit in CAN_CTL0 register.
3. Request to disable the Controller Interface clock, while keeping the Protocol controller clock running.

In Pretended Networking mode, Controller Interface clock is disabled and Protocol controller is kept clocked (if the MCU works in deepsleep mode, the clock of CAN Protocol Controller should be configured as HXTAL or IRC8M in advance, otherwise the the clock of CAN Protocol Controller will be lost), so that the reception process is still active to filter messages. The matching, arbitration, shift-in and shift-out processes are not performed in Pretended Networking mode.

To exit from Pretended Networking mode, the following method can meet:

- When a wakeup event is detected, and a wake up interrupt is occured. Clear PNEN bit and PNMOD bit in CAN_CTL0 register.
- Clear PNEN bit and PNMOD bit in CAN_CTL0 register.

If exiting from Pretended Networking mode is requested, CAN module will wait for the bus to be in Idle state or else wait for the third bit of Intermission to clear LPS bit and PNS bit in CAN_CTL0 register, and resume normal mode, CAN module will be synchronized to the CAN bus.

23.3.5. Data transmission

For transmission, an arbitration mechanism decides whether the Tx mailbox transmission priority is depending on the message ID (the PRIO field can also be configured to participate in arbitration), or on the mailbox number.

The quantity of mailboxes in CAN FD format is determined by MDSZ[1:0] bits in CAN_FDCTL register, refer to ***Mailbox size for CAN FD***.

Transmit process

To transmit a CAN frame, a Tx mailbox must be prepared for transmission in following steps:

1. Check whether the corresponding mailbox state MSx bit in CAN_STAT register is set and clear it.
2. If the mailbox is active (either Tx or Rx), inactivate the mailbox by method described in ***Tx mailbox inactivation*** or ***Rx mailbox inactivation***, when Tx mailbox inactivation is performed, do the following steps, when Rx mailbox inactivation is performed, go to step

6. While if the mailbox is inactive (either Tx or Rx), go to step 6.

3. Poll the the corresponding MSx bit in CAN_STAT register to be set, or by the interrupt when MIEx bit in CAN_INTEN register is set.
4. Read back the CODE field to get the state of the mailbox (aborted, or transmitted).
5. Clear the corresponding flag MSx in the CAN_STAT register.
6. Write mailbox ID field (plus the mailbox PRIO field if LAPRIOEN bit in CAN_CTL0 register is set to 1) of the MDES1 word.
7. Write payload data bytes in mailbox DATA field of MDESx (x = 2..17) word.
8. Configure the mailbox IDE, RTR, FDF, BRS, ESI, and DLC field to MDES0 word.
9. Activate the mailbox to transmit the frame by setting mailbox CODE field to 0b1100. When the mailbox is activated, it participates in the arbitration process and is eventually transmitted according to its priority. When the mailbox payload size is less than the mailbox DLC value, CAN adds the necessary number of bytes with constant 0xCC to meet the expected DLC.

Upon a successful transmission, the CODE field is automatically updated, and the TIMESTAMP field is automatically updated with the value of the free running counter; the CRC registers (CAN_CRCC and CAN_CRCCFD) are updated, and the corresponding flag MSx in the CAN_STAT register is set, if the interrupt enable bit MIEx in CAN_INTEN register is set, an interrupt will be generated.

Arbitration process

When more than one Tx mailbox is pending, the arbitration process which searching from the lowest number mailbox to the higher ones will give the transmission order. The arbitration algorithm is controlled by the MTO bit in CAN_CTL1 register.

The arbitration process starts when matching one of the following situations:

- The CRC field on CAN bus: number of ASD[4:0] (in CAN_CTL2 register) CAN bits delay after the first bit of the CRC field.
- The Error or Overload Delimiter field on CAN bus.
- CAN bus is recovering from Bus Off state: number of ASD[4:0] (in CAN_CTL2 register) CAN bits delay after the counter TECNT[7:0] counted to 124. Recovering from Bus Off state needs 128 times of 11 continuous recessive bits, which is counted by TECNT[7:0] in CAN_ERR0 register.
- Exit from Inactive mode, or power saving mode (including CAN_Disable mode and Pretended Networking mode).
- Rewrite of MDES0 word of arbitration winner (temporary winner or final winner).
- Rewrite to MDES0 word of the scanned mailbox (arbitration is on-going): if no arbitration winner is found when scan finished, arbitration will restart at soon; otherwise, the arbitration process is finished.
- Write to MDES0 word of a mailbox: when no arbitration is processing, and no arbitration winner exists, and the CAN bus is not in SOF-DATA field / SOF-Control field of a data / remote frame or Error / Overload flag field of an Error / Overload frame.
- CAN node enters Bus Integration state (refer to ***Bus integration state***): Number of

ASD[4:0] (in CAN_CTL2 register) CAN bits delay after the state.

Arbitration process stops when matching one of the following situations:

- All mailboxes are scanned.
- A Tx active mailbox is found when MTO bit in CAN_CTL1 register is set to 1 (lowest-number mailbox first).
- The Error or Overload flag field on CAN bus.
- The SOF field of the next frame on CAN bus.
- When Inactive mode, CAN_Disable mode or Pretended Networking mode is requested.

Lowest-number mailbox first

If MTO bit in CAN_CTL1 register is set to 1, the lowest number Tx mailbox is transmitted first, and LAPRIOEN bit in CAN_CTL0 register has no effect.

Highest-priority mailbox first

If MTO bit in CAN_CTL1 register is set to 0, then the Tx mailbox with the highest priority is transmitted first. The highest priority Tx mailbox is the one that has the lowest arbitration value (refer to ***Table 23-7. Mailbox arbitration value(32 bit) when local priority disabled*** and ***Table 23-8. Mailbox arbitration value(35 bit) when local priority enabled***) among all Tx mailboxes. If more than one mailboxes have equivalent arbitration values, the mailbox with the lowest number is the arbitration winner.

When LAPRIOEN bit in CAN_CTL0 register is set to 1, the local priority is disabled, the bits participate in the internal arbitration process are exactly what will be transmitted to the CAN bus, shown in ***Table 23-7. Mailbox arbitration value(32 bit) when local priority disabled***.

When LAPRIOEN bit in CAN_CTL0 register is set to 1, the local priority is enabled, then the mailbox PRIO field will participate in the internal arbitration process. Shown in ***Table 23-8. Mailbox arbitration value(35 bit) when local priority enabled***, the mailbox PRIO field is the most significant part of the arbitration value, thus mailboxes with low PRIO values have higher priority regardless of the rest of their arbitration values, while the PRIO field will not be transmitted to the CAN bus.

Table 23-7. Mailbox arbitration value(32 bit) when local priority disabled

IDE	31 30 29 28 27 26 25 24 23 22 21	20	19	18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	0
0	ID_STD[10:0]	RTR	IDE	Reserved	
1	ID_EXD[28:18]	SRR	IDE	ID_EXD[17:0]	RTR

Table 23-8. Mailbox arbitration value(35 bit) when local priority enabled

IDE	34 33 32	31 30 29 28 27 26 25 24 23 22 21	20	19	18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	0
0	PRIO[2:0]	ID_STD[10:0]	RTR	IDE	Reserved	
1	PRIO[2:0	ID_EXD[28:18]	S	ID	ID_EXD[17:0]	RT

]		R R	E		R

Arbitration start delay

Arbitration start delay is configured by ASD[4:0] bits in CAN_CTL2 register to optimize the transmission performance when the arbitration process ends too early to give a chance for the CPU to overwrite the winner Tx mailbox, thus the arbitration is restarted and may not be able to transmit in time.

Shift-out

The shift-out process is the copy operation of the content from a Tx mailbox to the Tx shift buffer (an internal mailbox descriptor) after the arbitration winner is found. The message on the Tx shift buffer is transmitted according to the CAN protocol rules.

When the shift-out process is done, write access to the MDES0 word of the corresponding mailbox is blocked even if MST bit in CAN_CTL0 register is set. The write access to MDES0 word of the corresponding mailbox is recoverd when matching one of the following situations:

- After the mailbox is transmitted and the corresponding flag MSx in the CAN_STAT register is cleared by the CPU.
- CAN node enters Inactive mode or Bus Off state.
- CAN node loses the bus arbitration or there is an error during the transmission.

The shift-out process starts when matching one of the following situations:

- The first bit of Intermission field on CAN bus.
- During Bus Idle state.
- During Wait For Bus Idle state.

During the shift-out process, the CPU has priority to access the corresponding memory in Bus Idle state, and the shift-out operation has the lowest priority to access the corresponding memory.

Abort

To request an abort of the transmission, the recommended operation is setting MST bit in the CAN_CTL0 register to 1, then writing ABORT (0b1001) to the CODE field of the mailbox.

The writing of ABORT (0b1001) to the mailbox MDES0 word is successfully when the mailbox is not the arbitration winner, or when the mailbox is the arbitration winner but the shift-out of the mailbox is not finished yet. In this situation, the corresponding MSx bit in CAN_STAT register will be set.

The writing of ABORT (0b1001) to the mailbox MDES0 word is blocked when shift-out of the mailbox is already finished, or when the mailbox is being transmitted. In this situation, the abort request is captured and kept pending until the frame is transmitted successfully or failed:

- The frame is transmitted successfully, the mailbox is not aborted: If the frame is

transmitted successfully, the pending abort request will be cleared automatically, the corresponding MSx bit in CAN_STAT register will be set, and an interrupt will occur when MIEx bit in CAN_INTEN register is set.

- The frame is transmitted failed, the mailbox is aborted: If the frame failed to be transmitted, the pending abort request is responsed, the write access to the mailbox is recoverd, with ABORT code written to the mailbox MDES0 word, the corresponding MSx bit in CAN_STAT register will be set, and an interrupt will occur when MIEx bit in CAN_INTEN register is set.
  When matching one of the following situations, the frame failed to be transmitted:
  - Lose the bus arbitration.
  - There is an error during the transmission.
  - Enter Bus Off state.
  - There is an overload frame.

Tx mailbox inactivation

The way to inactivate a Tx mailbox:

- Write the CODE field of the Tx mailbox MDES0 with ABORT. This is the recommended way for inactivation, which will not cause the unknown transmission.
  This operation must be done when MST bit in the CAN_CTL0 register is 1.

23.3.6. Data reception

For Classical CAN frames, reception through FIFO and mailbox are both supported.

For CAN FD frames, reception is only supported through mailbox.

Mailbox reception

For mailbox reception, a received frame will be stored into the mailbox only when the frame ID matches the mailbox ID programmed in the ID field (or the mailbox ID group when the filter registers are applied).

To receive a CAN frame into a mailbox, a mailbox must be prepared for reception in following steps:

1. If the mailbox is active (either Tx or Rx), inactivate the mailbox by method described in ***Tx mailbox inactivation*** or ***Rx mailbox inactivation***, when Tx mailbox inactivation is performed, do the following steps, when Rx mailbox inactivation is performed, go to step 4. While if the mailbox is inactive (either Tx or Rx), go to step 4.
2. Poll the the corresponding MSx bit in CAN_STAT register to be set, or by the interrupt when MIEx bit in CAN_INTEN register is set.
3. Read back the CODE field to make sure that the mailbox is aborted, or transmitted.
4. Clear the corresponding flag MSx in the CAN_STAT register.
5. Write the mailbox ID field of MDES1 word, and write IDE, RTR field of MDES0 word if needed.
6. Write the EMPTY (0b0100) to the CODE field of MDES0 word to activate the mailbox.

Upon a successful reception, all bits of the mailbox descriptor (DATA, ID, TIMESTAMP, SRR, IDE, RTR, FDF, BRS, ESI, DLC, CODE) are stored with the received data field or automatically updated, and the corresponding flag MSx in the CAN_STAT register is set, if the interrupt enable bit MIEx in CAN_INTEN register is set, an interrupt will be generated. The TIMESTAMP field is automatically updated with the value of the free running counter at the time of the second bit of frame's ID field.

To service (read) a Rx mailbox, the recommended steps are shown as below:

1. Poll the corresponding flag MSx in the CAN_STAT register to be set, or by the interrupt when MIEx bit in CAN_INTEN register is set.
2. Read the mailbox MDES0 word, and poll until the BUSY bit (in CODE field) is 0. When the BUSY bit is 0, the read operation of the mailbox will lock the mailbox, so that to prevent the mailbox being overwritten.
3. Read the contents of the mailbox.
4. Clear the corresponding flag MSx in the CAN_STAT register.
5. Read CAN_TIMER register to unlock the mailbox.

Rx mailbox locking

A locking mechanism is only applied for Rx mailbox: For CODE field with Rx FULL or Rx OVERRUN, a CPU read to the mailbox MDES0 word will lock the mailbox, thus the locking will prevent a new matching frame overwriting it.

The locking will be released when reading the CAN_TIMER register (global unlocking operation), or when MDES0 word of any other mailbox is read. When unlocked, a shift-in process will start with the pending message (the same in Inactive mode, while when LPS bit in CAN_CTL0 register is 1, the shift-in process will be delayed until LPS bit changes to 0).

If the mailbox is not unlocked in time, while a new matching frame is coming, then the new frame will overwrite the Rx shift buffer without a notification of a lost message, and no error is recorded.

Note: Mailbox inactivation (write CODE with Rx INACTIVE, or Tx ABORT) has higher priority than locking.

Rx mailbox inactivation

The way to inactivate a Rx mailbox:

- Write the CODE field of the Rx mailbox MDES0 with INACTIVE (Tx INACTIVE or Rx INACTIVE). But this operation may lead to a result that a matched message lost without notice.

Note: The Rx mailbox inactivation will automatically unlocks the mailbox. There is no write protection for Rx FIFO.

Rx FIFO reception

The Rx FIFO is 6-message deep. When RFEN bit in CAN_CTL0 register is 1, Rx FIFO is

enabled for reception. Rx FIFO can only be used for reception, and must not be enabled when CAN FD mode is enabled. The Rx FIFO descriptor refers to ***Rx FIFO descriptor***. There is a powerful filter system provided to filter a group of identifiers, reducing the interrupt servicing workload. The number of Rx FIFO filters is configured in RFFN[3:0] bits of CAN_CTL2 register, up to 32 filters are configured in CAN_RFIFOMPFx (x = 0..31) registers (if RPFQEN bit in CAN_CTL0 register is 1), or CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) registers (if RPFQEN bit in CAN_CTL0 register is 0).

Rx FIFO has unread messages: Only when MS5_RFNE bit in CAN_STAT register is 1, the FDES0-FDES3 words are valid to be read to get the received message. When MS5_RFNE bit in CAN_STAT register is 1, it means there is at least one frame available to be read from Rx FIFO. If the interrupt enable bit MIEx in CAN_INTEN register is set, an interrupt will be generated; while if DMAEN bit in CAN_CTL0 register is set, the MS5_RFNE flag will generate the DMA request and no Rx FIFO interrupt is generated.

- To service (read) Rx FIFO by CPU, the recommended steps are shown as below:

  1. Poll the flag MS5_RFNE in the CAN_STAT register to be set, or by the interrupt when MIE5 bit in CAN_INTEN register is set.
  2. Read the Rx FIFO FDES0-FDES3 words, and if needed read CAN_RFIFOIFMN register.
  3. Clear the flag MS5_RFNE in the CAN_STAT register. If there are more than one messages in the Rx FIFO, the act of clearing the flag will update the Rx FIFO FDES0-FDES3 words with the next message, and CAN_RFIFOIFMN register will be updated at the same time, the flag MS5_RFNE remains set, and an interrupt occurs again if enabled, repeat step 2 and 3 to get the received messages.

- To service (read) Rx FIFO by DMA controller, the recommended operation are shown as below :

  1. Configure and enable the DMA controller for Rx FIFO reception.
  2. Service (read) Rx FIFO by CPU until the flag MS5_RFNE in the CAN_STAT register is cleared, to avoid an additional DMA request after DMA mode is enabled.
  3. Set DMAEN bit in CAN_CTL0 register to 1 to enable DMA mode.
  4. Wait for the DMA request. When the flag MS5_RFNE in the CAN_STAT register is set, a DMA request is generated.
  5. Upon receiving the DMA request, the DMA will read the Rx FIFO FDES0 to FDES3. FDES3 word must be read to clear the flag MS5_RFNE in the CAN_STAT register, if there are more than one messages in the Rx FIFO, the act of reading FDES3 word will update the Rx FIFO FDES0-FDES3 words with the next message, and CAN_RFIFOIFMN register (should be read before FDES3) will be updated at the same time, the flag MS5_RFNE remains set, and a DMA request is generated again. Steps 4 and 5 are repeated.

DMA mode

DMA mode is supported for Rx FIFO reception when RFEN bit and DMAEN bit in CAN_CTL0 register are both set. When the DMA mode is enabled, the CPU must not read the Rx FIFO.

When DMA mode is enabled, Rx FIFO FDES0 to FDES3 words will be read by DMA controller when there is unread message in Rx FIFO, to get the received message. In this mode, Rx FIFO warning flag MS6_RFW bit and Rx FIFO overflow flag MS7_RFO bit in CAN_STAT register are reserved.

Before disabling DMA mode by clearing DMAEN bit in CAN_CTL0 register, a clear FIFO operation (when RFEN bit in CAN_CTL0 register is 1, set MS0 in the CAN_STAT register to 1 in Inactive mode) must be performed to clear the Rx FIFO contents. The act of clearing FIFO will clear MS5_RFNE bit in the CAN_STAT register, and cancel the DMA request.

Clear FIFO

when Rx FIFO is enabled (RFEN bit in CAN_CTL0 register is 1), set MS0 bit in the CAN_STAT register to 1 in Inactive mode will clear the Rx FIFO contents, while the Rx FIFO flags will not be cleared (except in DMA mode), thus before clearing FIFO operation, the Rx FIFO must be serviced until the flag MS5_RFNE in the CAN_STAT register is cleared.

Flag

Rx FIFO not empty

When MS5_RFNE bit in CAN_STAT register is 1, it means there is at least one frame available to be read from Rx FIFO.

Rx FIFO warning

When MS6_RFW bit in CAN_STAT register is 1, it means the number of unread messages within the Rx FIFO is increased to five from four due to the reception of a new one, the Rx FIFO is almost full.

Rx FIFO overflow

When MS7_RFO bit in CAN_STAT register is 1, it means there is an incoming message lost because the Rx FIFO is full.

Matching process

The matching process is searching for a Rx mailbox or Rx FIFO (when enabled) with an ID matching with the frame ID on CAN bus, also, the IDE and RTR field will participate in the matching.

The matching process starts when completes the DLC field reception.

Searching process

- When the Rx FIFO is enabled, the RFO bit in CAN_CTL2 register gives the searching order.
  - If RFO bit is set to 1, matching process starts from Rx mailbox to Rx FIFO. The Rx mailbox is searched from the lowest number mailbox to the higher ones.
    Firstly, searching for a matching mailbox that is available for receiving. If the

RPFQEN bit is 0, the first mailbox that matches is the winner, regardless of whether it is free-to-receive or non-free-to-receive. If the RPFQEN bit is 1, the first free-to-receive mailbox matched is the match winner. Rx FIFO is no longer searched in any of the above cases.

Secondly, If the RPFQEN bit is 1 and no free-to-receive mailbox is matched, but a matching non-free-to-receive mailbox is found, then the Rx FIFO is also searched to determine the winner. when the Rx FIFO is matched and is not full, the Rx FIFO is the matching winner; otherwise, the matching winner is the last non-free-to-receive matched Rx mailbox(leading to mailbox CODE OVERRUN).

Thirdly, if no matched Rx mailbox is found (means no free-to-receive matched mailbox, nor non-free-to-receive matched mailbox), then matching is processed on Rx FIFO. In this case, if the Rx FIFO is matched but it is full, it will leads to Rx FIFO overflow, while if the Rx FIFO is not matched (no matter it is full or not), the message will not be received.

- If RFO bit is set to 0, matching process starts from Rx FIFO to Rx mailbox. If the Rx FIFO matches the searching conditions, and is not full, then the Rx FIFO is the matching winner, regardless of searching for mailboxes. If Rx FIFO does not match or it is full, then matching is processed on Rx mailbox. The matching of mailboxes is affected by the RPFQEN bit. If the RPFQEN bit is 0, the first mailbox to be matched is the match winner, regardless of whether it is free-to-receive or non-free-to-receive. If the RPFQEN bit is 1, the first free-to-receive mailbox matched is the winner. If no free-to-receive mailbox is found, the last non-free-to-receive mailbox matched is the winner.

- When the Rx FIFO is disabled, the matching process only searches the Rx mailboxes. Refer to the mailbox matching description above.

A free-to-receive Rx mailbox can be:

- For a data frame reception, or a remote frame reception when RRFRMS bit in CAN_CTL2 register is 1, it can be: A mailbox with CODE field EMPTY; A mailbox with CODE field FULL or OVERRUN, which has already been serviced (read) and unlocked.
- For a remote frame reception when RRFRMS bit in CAN_CTL2 register is 0, it can be a mailbox with CODE field RANSWER.

Searching conditions for matched Rx mailbox

Searching conditions for matched Rx mailbox, refers to ***Table 23-9. Rx mailbox matching***:

- When the frame in Rx shift buffer is a data frame (RTR field is 0), Rx mailbox with CODE EMPTY, FULL, and OVERRUN will be searched:
  - When IDERTR_RMF bit in CAN_CTL2 register is 0, it means the IDE field will be compared and RTR field will not be compared (regardless of bit 30 and bit 31 in related filter register). The ID field will be compared, using bit 0 to bit 28 filter data configurations in related filter register.
  - When IDERTR_RMF bit in CAN_CTL2 register is 1, it means all the IDE, RTR and ID fields will be compared, using bit 0 to bit 28, bit 30, bit 31 filter data configurations in related filter register.

- When the frame in Rx shift buffer is a remote frame (RTR field is 1):
  - If RRFRMS bit in CAN_CTL2 register is 0, it indicates that the Rx mailbox with CODE RANSWER will be searched, and the IDE, ID field will be compared, using bit 0 to bit 28, bit 30 filter data configurations in related filter register.
  - If RRFRMS bit in CAN_CTL2 register is 1, it indicates that the matching process is the same as a data frame, so Rx mailbox with CODE EMPTY, FULL, and OVERRUN will be searched:
    When IDERTR_RMF bit in CAN_CTL2 register is 0, it means the IDE field will be compared and RTR field will not be compared (regardless of bit 30 and bit 31 in related filter register). The ID field will be compared, using bit 0 to bit 28 filter data configurations in related filter register.
    When IDERTR_RMF bit in CAN_CTL2 register is 1, it means all the IDE, RTR and ID fields will be compared, using bit 0 to bit 28, bit 30, bit 31 filter data configurations in related filter register.

Table 23-9. Rx mailbox matching

Received bit	Configuration bit		Field in mailbox descriptor for matching			
RTR	IDERTR_RMF (in CAN_CTL2 register)	RRFRMS (in CAN_CTL2 register)	IDE	RTR	ID	CODE
0	0	-	Compared(1)	Never(2)	Filtered(3)	EMPTY / FULL / OVERRUN
	1		Filtered			EMPTY / FULL / OVERRUN
1	-	0	Compared	Never	Compared	RANSWER
	0	1	Compared	Never	Filtered	EMPTY / FULL / OVERRUN
	1		Filtered			EMPTY / FULL / OVERRUN

1. Compared: This field in Rx mailbox descriptor is always compared with the received bit, regardless of the filter data configurations in related filter register.
2. Never: This field in Rx mailbox descriptor is not compared with the received bit, regardless of the filter data configurations in related filter register.
3. Filtered: This field in Rx mailbox descriptor is compared with the received bit, using the filter data configurations in related filter register.

Searching conditions for matched Rx FIFO

Searching conditions for matched Rx FIFO, refers to ***Table 23-10. Rx FIFO matching***:

- If the FS[1:0] bits in CAN_CTL0 register is 0 or 1, it means A or B format of filter table is adopted, then all the IDE, RTR and ID fields will be compared, using bit 0 to bit 31 filter data configurations in related filter register.
- If the FS[1:0] bits in CAN_CTL0 register is 2, it means C format of filter table is adopted,

then the IDE, RTR will not be compared (no these fields in FIFO descriptor) and ID fields will be compared, using bit 0 to bit 31 filter data configurations in related filter register.

- If the FS[1:0] bits in CAN_CTL0 register is 3, it means D format of filter table is adopted, then all frames are rejected.

Table 23-10. Rx FIFO matching

Configuration bit	Field in Rx FIFO descriptor for matching		
FS[1:0] (in CAN_CTL0 register)	IDE	RTR	ID
0	Filtered		
1	Filtered		
2	Never		Filtered
3	Not match(1)		

1. Not match: All frames are rejected.

Shift-in

The shift-in process is the copy operation of the content from a Rx shift buffer (an internal mailbox descriptor) to a Rx mailbox or Rx FIFO that matched it.

When there is a matching descriptor found in the FIFO or in the Rx mailboxes, a shift-in process will be pending. The pending shift-in process starts to transfer when meets all of the following conditions:

- There is a matching winner for the frame in the Rx shift buffer.
- The CAN bus is in:
  - The second bit of Intermission field.
  - The first bit of an Overload frame.
- The target mailbox is not locked.

When the target mailbox of a pending shift-in process is unlocked during Inactive mode, the pending shift-in process starts to transfer. While if the unlocking occurs when LPS bit in CAN_CTL0 register is 1, the pending shift-in process will still be delayed until LPS bit changes to 0.

When the shift-in process is on-going, the BUSY bit (CODE[0]) of the target mailbox is set to indicate that the mailbox is being updated.

The shift-in process can be cancelled for a Rx mailbox, but can not be cancelled for the Rx FIFO. The shift-in process will be cancelled when matching one of the following situations:

- The target mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished.
- The Rx shift buffer receives a message transmitted by itself while the self reception is disabled by setting SRDIS bit in CAN_CTL0 register.
- There is a CAN protocol error.

When the shift-in process is done, Rx mailbox descriptor or Rx FIFO descriptor (if Rx FIFO is enabled) will be updated with the received message, and CAN_RFIFOIFMN will be updated if shift-in to the Rx FIFO, CODE field of Rx mailbox descriptor will be updated if shift-in to the Rx mailbox.

Filter data configuration

When Rx FIFO is disabled:

- If RPFQEN bit in CAN_CTL0 register is 0, then CAN_RMPUBF is used for all mailboxes.
- If RPFQEN bit in CAN_CTL0 register is 1, then CAN_RFIFOMPFx (x = 0..31) is used for mailboxes individually.

When Rx FIFO is enabled:

- If RPFQEN bit in CAN_CTL0 register is 0, then CAN_RMPUBF is used for all mailboxes, CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) are used for all the Rx FIFO ID filter table elements, and the value of these registers must be all the same.
- If RPFQEN bit in CAN_CTL0 register is 1, then CAN_RFIFOMPFx (x = 0..31) is used for the Rx FIFO ID filter table elements defined by RFFN[3:0] bits in CAN_CTL2 register and the mailboxes individually (because the Rx FIFO descriptor and the Rx mailbox descriptors can not occupy the same RAM space at the same time), CAN_RFIFOPUBF is used for the Rx FIFO ID filter table elements of the rest.

Self reception

When SRDIS bit in CAN_CTL0 register is 1, self reception is disabled, thus the frames transmitted by itself will not be received even if there is a matched Rx mailbox or Rx FIFO is matched, and no flag or interrupt will be generated. When the SRDIS bit is 0, it is allowed to receive a matching frame sent by itself.

23.3.7. Data reception in Pretended Networking mode

When PNEN bit and PNMOD bit in CAN_CTL0 register are configured to 1,the Pretended Networking mode is enabled, then the CAN is able to process received messages in MCU sleep mode. A wakeup event will wake up the CAN module from the Pretended Networking mode.

There are four groups of registers used for matched message storage: CAN_PN_RWMxCS, CAN_PN_RWMxI, CAN_PN_RWMxD0 and CAN_PN_RWMxD1 registers, group x from 0 to 3. Therefore, four messages can be stored at most (when NMM[7:0] bits in CAN_PN_CTL0 register is larger than or equal to 4), and only the latest messages will be stored. The group x indicates the message arrival order. If NMM[7:0] is less than 4, only NMM[7:0] value of messages can be stored, at groups from 0 to NMM[7:0] minus 1.

When the data length of the frame to be stored is less than 8 bytes, the padding values which continued to the received DATA field to be written to the CAN_PN_RWMxD0 and CAN_PN_RWMxD1 registers (x = 0..3) are zeroes. No timestamp is stored for wakeup

matched frames.

Note: When in Pretended Networking mode, CAN FD format messages are ignored.

Wakeup interrupt

There are two types of wakeup interrupt events, including wakeup match event, and wakeup timeout event. Each interrupt event has a dedicated flag bit in the CAN_PN_STAT register, and a dedicated enable bit in CAN_PN_CTL0 register. The relationship is described in the ***Table 23-11. Interrupt events***.

An wakeup interrupt can be generated when any type of the wakeup interrupt event occurs and enabled.

Wakeup timeout event

When CAN reaches the timeout value, a wakeup timeout event will occur. The timeout is configured by WTO[15:0] bits in CAN_PN_TO register.

Note: Even if the timeout value is reached, CAN module will not stop the message filtering process until the CPU wakes up.

Wakeup match event

When CAN receives the matched wakeup frame/frames within the timeout, the wakeup match event will occur. MMCNT[7:0] bits in CAN_PN_STAT register reflects the number of matched messages from the time of entering Pretended Networking mode to the time the CPU wakes up.

Note: Even if CAN receives the matched wakeup frame/frames, the timeout counter will not stop counting until the CPU wakes up.

Frame matching

The fields of frame participate in the wakeup matching process are IDE, RTR, ID, DLC, and DATA field.

- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 0, it means a wakeup match event occurs when a frame is received with all fields except DATA field, DLC field (that is IDE, RTR, and ID field matched) matched.
- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 1, it means a wakeup match event occurs when a frame is received with all fields (that is IDE, RTR, ID, DLC, and DATA field matched) matched.
- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means a wakeup match event occurs when a specified number (configured by NMM[7:0] bits in CAN_PN_CTL0 register) of frames are received with all fields except DATA field, DLC field (that is IDE, RTR, and ID field matched) matched.
- When FFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means a wakeup match event occurs when a specified number (configured by NMM[7:0] bits in CAN_PN_CTL0 register) of frames are received with all fields (that is IDE, RTR, ID, DLC,

and DATA field matched) matched.

IDE field matching

The IDE field of a matched message is the same as the configured expected IDE field in CAN_PN_EID0 register, using filter data in CAN_PN_IFEID1 register.

RTR field matching

The RTR field of a matched message is the same as the configured expected RTR field in CAN_PN_EID0 register, using filter data in CAN_PN_IFEID1 register.

ID field matching

- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 0, it means the ID field of a matched message is the same as the configured expected ID field in CAN_PN_EID0 register, using filter data in CAN_PN_IFEID1 register.
- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 1, it means the ID field of a matched message is larger than or equal to the configured expected ID field in CAN_PN_EID0 register. IDFD_EHT[28:0] bits in CAN_PN_IFEID1 register are not used.
- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means the ID field of a matched message is smaller than or equal to the configured expected ID field in CAN_PN_EID0 register. IDFD_EHT[28:0] bits in CAN_PN_IFEID1 register are not used.
- When IDFT[1:0] bit field in CAN_PN_CTL0 register is configured to 3, it means the ID field of a matched message is larger than or equal to the configured expected ID field in CAN_PN_EID0 register, and is smaller than or equal to the configured expected ID field in CAN_PN_IFEID1 register.

DLC field matching

- The DLC field of a matched message is larger than or qual to the configured expected DLC low threshold DLCELT[3:0] in CAN_PN_EDLC register, and lower than or qual to the configured expected DLC high threshold DLCEHT[3:0] in CAN_PN_EDLC register.

DATA field matching

- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 0, it means the DATA field of a matched message is the same as the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers, using filter data in CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers.
- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 1, it means the DATA field of a matched message is larger than or equal to the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers. CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers are reserved.
- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 2, it means the DATA field of a matched message is smaller than or equal to the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers. CAN_PN_DF0EDH0 and

CAN_PN_DF1EDH1 registers are reserved.

- When DATAFT[1:0] bit field in CAN_PN_CTL0 register is configured to 3, it means the DATA field of a matched message is larger than or equal to the configured expected DATA field in CAN_PN_EDLx (x = 0,1) registers, and is smaller than or equal to the configured expected DATA field in CAN_PN_DF0EDH0 and CAN_PN_DF1EDH1 registers.

  Note: In this case, all the two 8 bytes of the expected data register should be configured, when the DLC of the received message (DLC field matched) is less than 8 bytes, then in DATA field matching, the data that matching with the expected data is the received DATA field plus the padding value zeros.

23.3.8. CAN FD operation

Both ISO CAN FD (ISO11898-1 specification) and non-ISO (Bosch CAN FD Specification V1.0) CAN FD protocols are supported, but they are incompatible with each other, so select the protocol by ISO bit in CAN_CTL2 register. In comparison to the non-ISO CAN FD protocol, a 3-bit counter and a parity bit are introduced in ISO CAN FD protocol. Thus the failure detection capability is improved for ISO CAN FD.

CAN FD mode supports both CAN classical frames and CAN FD frames. The FDF bit (the reserved bit in CAN classical frames) is used to distinguish between CAN classical and CAN FD format frames. When the FDF bit is recessive ‘1’, it is recognized as a CAN FD frame; otherwise, it is a classical frame. Compared with CAN classical frame, CAN FD frame does not support Rx FIFO, Rx FIFO DMA, and Pretended Networking function.

To enable CAN FD mode, set FDEN bit in CAN_CTL0 register to 1.

CAN FD BRS

In CAN FD mode, the data byte length is allowed up to 64 bytes for a CAN FD frame, and the bit time can switch to a higher speed of 8 Mbit/s for the Data Phase (from BRS bit to the first sample point of CRC Delimiter or to the starting of an error frame when an error condition is detected) of a CAN FD frame with BRS bit set (refer to ISO11898-1 or Bosch CAN FD Specification V1.0).

When BRSEN bit in CAN_FDCTL register is set to 1 (takes effect at the next message), and BRS bit in Tx mailbox is written as recessive ‘1’, then higher bit time (called as data bit time) will be used for the Data Phase of CAN FD frame, the nominal bit time will be used for the rest of the bits. The bit time is changed at the sample point of the BRS bit. The data bit time is configured in CAN_FDBT register. The nominal bit time is configured in CAN_BT register.

When BRSEN bit in CAN_FDCTL register is set to 0, or when BRS bit in Tx mailbox is written as 0, then nominal bit time will be used for the entire CAN FD frame.

Note: The length of time quantum should be the same for the entire CAN FD frame, to reduce the possibilities of phase error frames on the CAN bus.

In FD frames, all nodes shall accept an up to two bit long dominant phase of overlapping ACK slot bits as a valid ACK, to compensate for phase shifts between the receivers. (Refer to ISO11898-1 specification)

CAN FD ESI

The transmission of ESI bit (the bit before DLC bits, refer to ISO11898-1 or Bosch CAN FD Specification V1.0) is defined by ESI field in MDES0 word of Tx mailbox and ERRSI[1:0] bits in CAN_ERR1 register. If ESI field in MDES0 is 0, it will transmit the dominant bit by error active nodes and transmit the recessive bit by error passive nodes according to ERRSI[1:0] bits in CAN_ERR1 register. If ESI field in MDES0 is 1, it will transmit ESI field in MDES0 word.

CAN FD CRC

Different CRC polynomials are used for different frame formats, results in a Hamming distance of 6:

- The CRC_15 polynomial is used for frames in CAN classical format: 0xC599
  $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- The CRC_17 polynomial is used for frames in CAN FD format with DATA field no more than 16 bytes: 0x3685B
  $x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1$
- The CRC_21 polynomial is used for frames in CAN FD format with DATA field more than 16 bytes: 0x302899
  $x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1$

For transmission, these three types of CRC will all be calculated at the start of the frame, and the final CRC to be transmitted is determined by the FDF field and DLC field of the frame. After a successful transmission, when corresponding MSx bit of CAN_STAT register is set, the CAN_CRCCFD register is updated at the same time, with the calculated CRC for both CAN FD and non-FD messages. The CAN_CRCC register only stores the calculated CRC for CAN classical format frames.

For reception, the CRC polynomial used for CRC check is determined by the received FDF and DLC field.

Note: In Classical frames, the CRC delimiter is one single recessive bit. In FD frames, the CRC delimiter may consist of one or two recessive bits. A transmitter shall send only one recessive bit as CRC delimiter, but it shall accept two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. A receiver will send its acknowledge bit after the first CRC delimiter bit. Refer to ISO11898-1 specification.

Bit stuff

The bit stuffing in CAN FD format frames is different from that in CAN classical format frames.

For transmission of CAN FD format frames, a fixed stuff bit is inserted before the first bit of the CRC field (regardless of the bit stuff conditions), and other fixed stuff bits are inserted

after each 4 bits of the CRC field (fixed stuff bits are not included). The value of these fixed stuff bits are the inverse value of their preceding bit. Refer to ISO11898-1 specification.

For reception of CAN FD format frames, the fixed stuff bits will be descarded. When the value of the fixed stuff bit is the same as the value of its preceding bit, a stuff error is detected.

Note: For CAN FD format frames, fixed stuff bits are included in CRC calculation. For CAN classical format frames, stuff bits are not included.

Resynchronization

Resynchronization and hard synchronization occur in CAN FD frames in the same way as in CAN classical frames. Resynchronization is not performed in transmitting the CAN FD data phase.

Transmitter delay compensation

The transmitter delay compensation is used for the data phase of CAN FD frames with BRS set, for the reason that in CAN FD frames with BRS bit set, the length of the CAN bit time in the data phase is shorter than the transmitter delay, thus the bit error check is influenced, the transmitter cannot receive its own transmitted bit latest at the sample point of that bit. The transmitter delay is measured from the falling edge of FDF bit of transmitted frame to the falling edge of FDF bit of received frame, shown in ***Figure 23-2. Transmitter delay***.

The transmitter delay compensation mechanism defines a secondary sample point SSP. When it is used, the transmitter shall ignore bit errors detected at the sample point. When TDCEN bit in CAN_FDCTL register is 1, this feature is enabled, then the bit check will be done between the actually received bit and the delayed transmitted bit (the delay is calculated based on the measured transmitter delay).

The transmitter delay compensation value is calculated in the equation follows:

$$t_{compensation} = t_{measure} + t_{offset} \quad (23\text{-}1)$$

with

$$t_{offset} = TDCO[4:0] \times t_{CANCLK} \quad (23\text{-}2)$$

$$t'_{offset} = t_{PBS1_FD} + t_{PTS_FD} + t_{SYNC_SEG} \quad (23\text{-}3)$$

$$t_{PBS1_FD} = (DPBS1[2:0] + 1) \times t_{q_FD} \quad (23\text{-}4)$$

$$t_{PTS_FD} = DPTS[4:0] \times t_{q_FD} \quad (23\text{-}5)$$

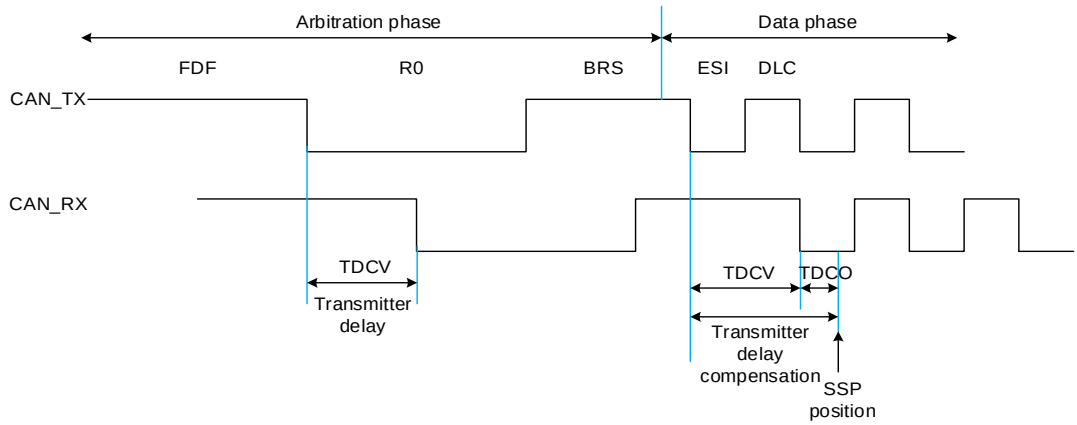
$$t_{q_FD} = (DBAUDPSC[9:0] + 1) \times t_{CANCLK} \quad (23\text{-}6)$$

where the $t_{measure}$ is the measured transmitter delay, t_{offset} is the transmitter delay compensation offset which is saved in the TDCO[4:0] bits of CAN_FDCTL register in unit of t_{CANCLK}, t_{offset} should not be larger than the CAN data bit time. t'_{offset} is the theoretical transmitter delay compensation offset, users can set t_{offset} refer to t'_{offset}. $t_{compensation}$ is the transmitter delay compensation value saved in TDCV[5:0] bits of CAN_FDCTL register in unit

of t_{CANCLK}.

In the equations, the DPBS1[2:0] bits, DPTS[4:0] bits, DBAUDPSC[9:0] bits are all configured in CAN_FDBT register.

Figure 23-2. Transmitter delay



The maximum $t_{compensation}$ is $(3 \times \text{data bit time} - 2 \times t_{q_FD})$. When exceed this value, it is unable to compensate the transmitter delay, then TDCS bit in CAN_FDCTL register will be set. The implementation shall be able to compensate transmitter delays of at least two data bit times.

23.3.9. Errors and states

Transmit Error Counter (TECNT[7:0] bits in CAN_ERR0 register) and Receive Error Counter (RECNT[7:0] bits in the CAN_ERR0 register) take into account all errors in both CAN FD and non-FD messages, which get incremented or decremented according to the error condition. For detailed information about TECNT[7:0] and RECNT[7:0] management, please refer to the CAN standard.

For CAN FD format frames, a Receive Error Counter for data phase of CAN FD messages (REFCNT[7:0] bits in the CAN_ERR0 register) and a Transmit Error Counter for data phase of CAN FD messages (TEFCNT[7:0] bits in the CAN_ERR0 register) are used additionally only when the BRS field of the frame is set. They stop counting and keep their values when in Bus off state, and they restart counting after returned to error active state by Bus off recovery.

Note: When in Pretended Networking mode, receive error counters keep counting and error flags are saved, transmit error counters stop counting and save their values. When returns to normal mode, the CAN_ERR0 and CAN_ERR1 registers will be updated with the counter value and saved error flags.

States

Error Passive State

If the value of TECNT[7:0] or RECNT[7:0] in CAN_ERR0 register increments to greater than

127, ERRSI[1:0] in CAN_ERR1 register is updated to 1 (error passive state).

Error Active state

If the node is in Error Passive state, and the value of either TECNT[7:0] or RECNT[7:0] in CAN_ERR0 register decrements to less than or equal to 127 when the other already satisfies this condition, ERRSI[1:0] in CAN_ERR1 register is updated to 0 (error active state).

Bus off state

If the value of TECNT[7:0] in CAN_ERR0 register becomes greater than 255, ERRSI[1:0] in CAN_ERR1 register is updated to 0b1x (Bus off state), and BOF bit in CAN_ERR1 register will be set, when BOIE bit in CAN_CTL1 register is set, an interrupt will be generated. The value of TECNT[7:0] will be reset to 0.

Bus off recovery:

To exit from Bus off state, the CAN has to wait for the recovery sequence specified in the CAN standard (128 occurrences of 11 consecutive recessive bits monitored on CAN RX). When TECNT[7:0] in CAN_ERR0 register reaches 128, ERRSI[1:0] in CAN_ERR1 register is updated to 0 (error active state) and both TECNT[7:0] and RECNT[7:0] in CAN_ERR0 register are reset to 0.

Depending on ABORDIS bit in CAN_CTL1 register, CAN will recover from Bus off automatically or remain in Bus off state.

When ABORDIS is 0, enable automatic bus off recovery, CAN wil recover from Bus off automatically after the recovery sequence. If the ABORDIS is changed to 0 after the recovery sequence, then CAN will resynchronize to the bus by detecting 11 consecutive recessive bits.

When ABORDIS is 1, not enable automatic bus off recovery. If the ABORDIS is changed to 1 after the CAN entered Bus off state, automatic bus off recovery will be disabled at the next time the CAN entered Bus off state.

Bus integration state

If the node starts the protocol operation during Bus off recovery, or detects the protocol exception event (the event occurs when FDEN bit in CAN_CTL0 register is set to 0, and a FDF bit of a FD frame is received), the node enters the bus integration state. In this state, the synchronicity to the CAN bus is lost. CAN node can leave the bus integration state when the bus idle condition (the sequence of 11 consecutive recessive bits) is detected. Refer to the CAN Protocol standard (ISO 11898-1).

The protocol exception detection is controlled by PREEN bit in CAN_CTL2 register.

The edge filtering can be configured by EFDIS bit in CAN_CTL2 register, which is used during the bus integration state. When the edge filtering is enabled, two consecutive nominal time quanta with dominant bus state are required to detect an edge that causes synchronization. When synchronization occurs, the counting for bus idle condition (the sequence of 11 consecutive recessive bits) is restarted. When edge filtering is performed, dominant bus-

states shorter than a nominal bit time (the bits in data phase of a FD frame) will be ignored, stopped from being mistaken for an idle condition. Refer to the CAN Protocol standard (ISO 11898-1).

Note: Recommend to always enable edge filtering by reset EFDIS bit, to avoid mistakenly detection of bus idle condition.

Errors

If at least one of the error flags (ACKERR, BRERR, BDERR, CRCERR, FMERR, and STFERR bit in CAN_ERR1 register) is set, ERRSF bit in CAN_ERR1 register will be set. If ERRSIE bit in CAN_CTL1 register is set, an error interrupt will be generated.

If at least one of the error flags (BRFERR, BDFERR, CRCFERR, FMFERR, and STFFERR bit in CAN_ERR1 register) is set, ERRFSF bit in CAN_ERR1 register will be set. If ERRFSIE bit in CAN_CTL2 register is set, an error interrupt will be generated for errors detected in CAN FD frame data phase with BRS bit set.

Acknowledge error

If there is only one node operating, then it will lead to TECNT[7:0] in CAN_ERR0 register incrementing (to 128 at most by acknowledge error) in each message transmission, and an acknowledge error will occur, which is indicated by ACKERR bit in the CAN_ERR1 register.

Bit recessive error

When at least one bit sent as recessive ‘1’ is received as dominant ‘0’, a bit recessive error occurs. Refers to BRFERR and BRERR bit in CAN_ERR1 register.

Bit dominant error

When at least one bit sent as dominant ‘0’ is received as recessive ‘1’, a bit dominant error occurs. Refers to BDFERR and BDERR bit in CAN_ERR1 register.

CRC error

When the calculated CRC is different from the received CRC field of the frame, a CRC error occurs. Refers to CRCFERR and CRCERR bit in CAN_ERR1 register.

Form error

When a fixed-form bit field contains at least one illegal bit, a form error occurs. Refers to FMFERR and FMERR bit in CAN_ERR1 register.

Stuff error

Refers to STFFERR and STFERR bit in CAN_ERR1 register.

23.3.10. Communication parameters

Bit time

The CAN bit time from the CAN protocol has three segments as follows:

Synchronization segment (SYNC_SEG): A bit change is expected to occur within this time segment. It has a fixed length of one time quantum ($1 \times t_q$).

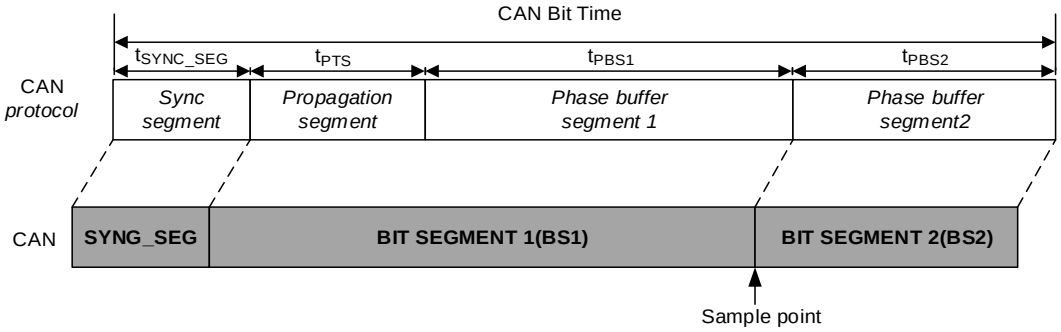
Bit segment 1 (BS1): It defines the location of the sample point. It includes the Propagation segment and Phase buffer segment 1 in the CAN standard. It may be automatically lengthened to compensate for positive phase drifts due to different frequency of the various nodes of the network.

Bit segment 2 (BS2): It defines the location of the sample point. It may also be automatically shortened to compensate for negative phase drifts. Its duration should programmed no less than 2 time quanta.

Note: The bit time configuration ranges must be in compliance with the CAN Protocol standard (ISO 11898-1).

CAN bit time is shown as in the ***Figure 23-3. CAN bit time***.

Figure 23-3. CAN bit time



Synchronization Jump Width (SJW): It can be lengthened or shortened to compensate for the Synchronization error of the CAN network node. It is configured by SJW[4:0] bits in CAN_BT register for nomainal bit time, or configured by DSJW[2:0] bits in CAN_FDBT register for data bit time.

A valid edge is defined as the first toggle in a bit time from dominant to recessive bus level before the controller sends a recessive bit.

If a valid edge is detected in BS1, not in SYNC_SEG, BS1 is added with up to SJW maximumly, so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2, not in SYNC_SEG, BS2 is cut down with up to SJW maximumly, so that the transmit point is moved earlier.

Bit sampling

BSPMOD in CAN_CTL1 register defines the sampling mode of CAN bits at the Rx input pin.

When BSPMOD is 0, only one sample (the sample point) is used.

When BSPMOD is 1, three samples are used to determine the received bit value, that is the one on the sample point, and the two preceding samples.

Note: This bit cannot be set when CAN FD is enabled.

Baudrate

CAN module has two clock domains:

- The clock of Control Interface and CAN registers derives from the APB2 clock.
- The clock of Protocol controller (CANCLK) can be configured by CANxSEL[1:0] bit in RCU_CFG2 register, to derive from oscillator clock, or APB2 clock, or APB2 clock devided by 2, or IRC8M internal clock.

The CAN calculates its baudrate as follows:

$$\text{BaudRate} = \frac{1}{\text{CAN Bit Time}} \tag{23-7}$$

$$\text{CAN Bit Time} = t_{SYNC_SEG} + t_{PTS} + t_{PBS1} + t_{PBS2} \tag{23-8}$$

with

$$t_{SYNC_SEG} = 1 \times t_q \tag{23-9}$$

$$t_{PTS} = (N_{PTS} + 1) \times t_q \text{ or } t_{PTS} = N_{DPTS} \times t_q \tag{23-10}$$

$$t_{PBS1} = (N_{PBS1} + 1) \times t_q \tag{23-11}$$

$$t_{PBS2} = (N_{PBS2} + 1) \times t_q \tag{23-12}$$

$$t_q = (N_{BAUDPSC} + 1) \times t_{CANCLK} \tag{23-13}$$

In the equations, for nominal bit rate:

N_{PTS}, N_{PBS1}, N_{PBS2}, and $N_{BAUDPSC}$ are configured by the PTS[5:0] bits, PBS1[4:0] bits, PBS2[4:0] bits, and BAUDPSC[9:0] bits respectively in CAN_BT register.

For data bit rate:

N_{DPTS}, N_{PBS1}, N_{PBS2}, and $N_{BAUDPSC}$ are configured by the DPTS[4:0] bits, DPBS1[2:0] bits, DPBS2[2:0] bits, and DBAUDPSC[9:0] bits respectively in CAN_FDBT register.

Note: In the reception matching process and transmission arbitration process, CAN needs to complete the search of all mailboxes in one CAN frame time. In order to meet the search time requirement of this part, the CAN bit time is calculated as CAN Bit Time = NUM × t_{CANCLK}, and the requirement of Min_{NUM}=16 must be met.

Timestamp

A 16-bit internal counter of the CAN hardware in CAN_TIMER register is used to generate the timestamp value. The value of the internal counter is sampled at SOF field on the CAN bus, and is written into the TIMESTAMP field of MDES0 or FDES0 word after a successful reception or transmission of a message.

The counter does not count in Inactive mode, or when LPS bit in CAN_CTL0 register is 1.

Counter clock source

When ITSRC bit in CAN_CTL2 register is 1, the TRIGSEL output CANx_EX_TIME_TICK is selected as the increasing condition of internal counter. The CANx_EX_TIME_TICK and APB2 bus clock are in the same clock domain, to ensure the internal counter be increased effectively, the pulse width of CANx_EX_TIME_TICK signal must be greater than or equal to the APB2 bus clock period.

When ITSRC bit in CAN_CTL2 register is 0, the CAN baudrate is selected as the increasing condition of internal counter, internal counter increments by one for each bit that is received or transmitted. When there is no message on the bus, it is counted with the baudrate programmed previously.

Time synchronization

If TSYNC bit in CAN_CTL1 register is 1, a SYNC message reception in the first mailbox descriptor will reset the internal counter for network time synchronization.

23.3.11. Interrupts

The CAN interrupt events and flags are list in ***Table 23-11. Interrupt events***.

Table 23-11. Interrupt events

Interrupt event		Flag			Enable control			
		Bit		Register	Enable bit	Control bit	Enable register	Control register
Bus off		BOF		CAN_ERR1	BOIE		CAN_CTL1	
Bus off recovery		BORF			BORIE		CAN_CTL2	
Error summary	Bit recessive error	ERRSF	BRERR		ERRSIE		CAN_CTL1	
	Bit dominant error		BDERR					
	ACK error		ACKERR					
	CRC error		CRCERR					
	Form error		FMERR					
	Stuff error		STFERR					
Error summary for FD frames with data bit time	Bit recessive error	ERRFSF	BRFERR		ERRFSIE		CAN_CTL2	
	Bit dominant error		BDFERR					
	CRC error		CRCFERR					
	Form error		FMFERR					
	Stuff error		STFFERR					
Tx error warning		TWERRIF			TWERRIE	WERREN	CAN_CTL1	CAN_CTL0
Rx error warning		RWERRIF			RWERRIE			
Wakeup match		WMS		CAN_PN_STAT	WMIE		CAN_PN_CTL0	
Wakeup timeout		WTOS			WTOIE			
Mailbox successful transmission or reception		All bits		CAN_STAT	All bits	RFEN = 0	CAN_INTEN	CAN_CTL0
		MSx			MIEx	RFEN = 1		
Rx FIFO not empty		MS5_RFNE			MIE5	RFEN = 1 & DMAEN = 0		
Rx FIFO warning		MS6_RFW			MIE6			
Rx FIFO overflow		MS7_RFO			MIE7			

23.4. Example for a typical configuration flow of CAN

After power-on reset or system reset, the following operation flow is a typical process for application to configure and run CAN:

- Configure CAN module clock source CANCLK, and enable CAN clock
  Configure CANxSEL[1:0] bits in CAN_CFG2 register to select the CAN module clock source. Program the RCU_APB2EN register to enable the CAN module clock.
- Setup the communication interface
  Configure GPIO and AFIO module to select PADs to alternate functions.
- Enter Inactive mode
  Because INAMOD bit, HALT bit, NRDY bit and INAS bit are default set after power-on reset or system reset, so CAN will automatically enters Inactive mode for configuration

of CAN registers.

- Service the flags in CAN_STAT register
  Read the Rx mailbox or Rx FIFO descriptor contents, clear the corresponding asserted flag bit in CAN_STAT register, then read the CAN_TIMER register at last for a complete flag bit service. If Rx FIFO is enabled, do a clearing FIFO operation by setting MS0 bit in CAN_STAT register to 1. Also clear the asserted flags by Tx mailboxes.
- Initialize the physical memory space for mailbox and Rx FIFO descriptors
  Configure memory space for mailbox and Rx FIFO descriptors totally by MSZ[4:0] bits in CAN_CTL0 register.
- Configure the communication parameters
  1) Configure the CAN nominal bit rate by PTS[5:0] bits, PBS1[4:0] bits, PBS2[4:0] bits, SJW[4:0] bits and BAUDPSC[9:0] bits in CAN_BT register.
  2) Configure bit sampling mode by BSPMOD bit in CAN_CTL1 register if needed.
  3) Configure PREEN bit and EFDIS bit for bus integration state if needed.
- Configure the control parameters for transmission
  1) Configure arbitration priority by MTO bit in CAN_CTL1 register and LAPRIOEN bit in CAN_CTL0 register.
  2) Configure arbitration start delay by ASD[4:0] bits of CAN_CTL2 register if needed.
  3) Enable transmission abort function for Tx mailbox descriptor configuration by MST bit in CAN_CTL0 register.
- Configure the control parameters for reception
  1) Choose whether use Rx FIFO and Rx FIFO DMA for reception or not by RFEN bit and DMAEN bit in CAN_CTL0 register.
  2) Configure Rx private filter & Rx mailbox queue feature by RPFQEN bit in CAN_CTL0 register.
  3) Configure receive filter related parameters by RFO bit, RRFRMS bit and IDERTR_RMF bit of CAN_CTL2 register.
  4) Configure filter data of the Rx mailbox and Rx FIFO by CAN_RMPUBF, CAN_RFIFOPUBF and CAN_RFIFOMPFx (x = 0..31) registers. If Rx FIFO is enabled, configure Rx FIFO ID filter table element format by FS[1:0] bits of CAN_CTL0 register, configure Rx FIFO ID filter table element number by RFFN[3:0] bits of CAN_CTL2 register.
- If CAN FD operation is needed
  1) Select CAN FD operation protocol by ISO bit in CAN_CTL2 regitser.
  2) Enable CAN FD operation by FDEN bit in CAN_CTL0 register.
  3) Initialize the mailbox data size by MDSZ[1:0] bits of CAN_FDCTL register.
  4) Configure CAN FD related transmitter delay compensation feature by TDCEN bit and TDCO[4:0] bits of CAN_FDCTL register if needed.
  5) Configure the CAN data bit rate by DPTS[4:0] bits, DPBS1[2:0] bits, DPBS2[2:0] bits DSJW[2:0] bits and DBAUDPSC[9:0] bits in CAN_FDBT register.
- Configure interrupts
  Enable the needed interrupts in CAN_CTL0, CAN_CTL1, CAN_CTL2 and CAN_INTEN registers.
- Initialize the Tx / Rx mailbox descriptors

1) If message transmission is needed, initialize the Tx mailbox descriptors.

2) If message reception is needed, initialize the Rx mailbox descriptors, if Rx FIFO is enabled, also initialize the Rx FIFO descriptors including the ID filter table elements.

- If Pretended Networking mode is required, set PNEN bit and PNMOD bit in CAN_CTL0 register and configure the necessary registers for wakeup.
- Exit Inactive mode

  Clear HALT bit in CAN_CTL0 register to exit Inactive mode, and CAN starts to synchronize to the CAN bus.

23.5. CAN registers

CAN0 base address: 0x4001 A000

CAN1 base address: 0x4001 B000

23.5.1. Control register 0 (CAN_CTL0)

Address offset: 0x00
Reset value: 0x5900 000F

All bits except bit 30, 28, 25, 19 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

All bits except bit 31, 27, 24, 20 of this register will be reset by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CANDIS	INAMOD	RFEN	HALT	NRDY	Reserved	SWRST	INAS	Reserved		WERREN	LPS	PNEN	PNS	SRDIS	RPFQEN
rw	rw	rw	rw	r		rw	r			rw	r	rw	r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAEN	PNMOD	LAPRIOEN	MST	FDEN	Reserved	FS[1:0]		Reserved			MSZ[4:0]				
rw	rw	rw	rw	rw		rw					rw				

Bits	Fields	Descriptions
31	CANDIS	CAN disable 0: Enable CAN module 1: Disable CAN module
30	INAMOD	Inactive mode enable 0: Disable Inactive mode 1: Enable Inactive mode
29	RFEN	Rx FIFO enable 0: Disable Rx FIFO 1: Enable Rx FIFO
28	HALT	Halt CAN 0: No enter Inactive mode request 1: Enter Inactive mode if the INAMOD bit in CAN_CTL0 register is set
27	NRDY	Not ready This bit indicates the state of whether the Protocol controller clock is disabled or not. When in Inactive mode, or in CAN_Disable mode, the Protocol controller clock is disabled, and CAN is not ready. 0: CAN is ready

		1: CAN is not ready
26	Reserved	Must be kept at reset value.
25	SWRST	Software reset When this bit is set, CAN internal state machines and CAN registers will be reset. This bit is automitically cleared by hardware when software reset is completed. 0: No effect 1: Software reset request
24	INAS	Inactive mode state 0: Not in Inactive mode 1: In Inactive mode
23:22	Reserved	Must be kept at reset value.
21	WERREN	Error warning enable When this bit is set, the warning interrupt flag TWERRIF and RWERRIF bit in CAN_ERR1 register will be enabled to reflect the state change of TWERRF and RWERRF bit in CAN_ERR1 register respectively. 0: Disable Tx and Rx error warining 1: Enable Tx and Rx error warining
20	LPS	Low power state 0: Not in low power state 1: In low power state
19	PNEN	Pretended Networking mode enable 0: Disable Pretended Networking mode 1: Enable Pretended Networking mode
18	PNS	Pretended Networking state 0: Not in Pretended Networking state 1: In Pretended Networking state
17	SRDIS	Self reception disable 0: Enable self reception 1: Disable self reception
16	RPFQEN	Rx private filters enable & Rx mailbox queue enable 0: Disable Rx private filters & disable Rx mailbox queue 1: Enable Rx private filters & enable Rx mailbox queue
15	DMAEN	DMA enable 0: DMA feature for RX FIFO disabled. 1: DMA feature for RX FIFO enabled.
14	PNMOD	Pretended Networking mode selection 0: Not select Pretended Networking mode 1: Select Pretended Networking mode

13	LAPRIOEN	Local arbitration priority enable 0: Disable local arbitration priority 1: Enable local arbitration priority
12	MST	Mailbox stop transmission 0: Disable transmission abort 1: Enable transmission abort
11	FDEN	CAN FD operation enable 0: Disable CAN FD operation 1: Enable CAN FD operation
10	Reserved	Must be kept at reset value.
9:8	FS[1:0]	Format selection This bit field defines the format of the Rx FIFO ID filter table elements. 00: Format A: One full ID (standard and extended) per ID filter table element 01: Format B: Two full standard IDs or two partial 14-bit extended IDs per ID filter table element 10: Format C: Four partial 8-bit IDs (standard and extended) per ID filter table element 11: Format D: All frames rejected
7:5	Reserved	Must be kept at reset value.
4:0	MSZ[4:0]	Memory size This bit field defines the maximum size of memory for message transmission and reception. The size is counted in uint of 4 words (equals to the size of a mailbox descriptor with 8-byte data), including mailbox and Rx FIFO. Before configuring this bit field, the flags in CAN_STAT register must be serviced. 00000: 1 unit 00001: 2 units … 11111: 32 units

23.5.2. Control register 1 (CAN_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

The bits 12, 7, 5, 4, 3 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

All bits of this register are not affected by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOIE	ERRSIE	Reserved	LSCMOD	TWERRIE	RWERRIE	Reserved		BSPMOD	ABORDIS	TSYNC	MTO	MMOD	Reserved		
rw	rw		rw	rw	rw			rw	rw	rw	rw	rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BOIE	Bus off interrupt enable 0: Disable Bus off interrupt 1: Enable Bus off interrupt
14	ERRSIE	Error summary interrupt enable 0: Disable error summary interrupt 1: Enable error summary interrupt
13	Reserved	Must be kept at reset value.
12	LSCMOD	Loopback and silent communication mode 0: Disable loopback and silent communication mode 1: Enable loopback and silent communication mode **Note:** In this mode, SRDIS bit in CAN_CTL0 register, and TDCEN in CAN_FDCTL register cannot be set.
11	TWERRIE	Tx error warning interrupt enable This bit can be written only when WERREN in CAN_CTL0 register is 1. This bit is read as zero when WERREN in CAN_CTL0 register is 0. 0: Disable Tx error warning interrupt 1: Enable Tx error warning interrupt
10	RWERRIE	Rx error warning interrupt enable This bit can be written only when WERREN in CAN_CTL0 register is 1. This bit is read as zero when WERREN in CAN_CTL0 register is 0. 0: Disable Rx error warning interrupt 1: Enable Rx error warning interrupt
9:8	Reserved	Must be kept at reset value.
7	BSPMOD	Bit sampling mode 0: One sample for the received bit 1: Three samples for the received bit
6	ABORDIS	Automatic Bus off recovery not enable 0: Enable automatic Bus off recovery 1: Not enable automatic Bus off recovery
5	TSYNC	Time synchronization enable 0: Disable time synchronization 1: Enable time synchronization

4	MTO	Mailbox transmission order 0: Highest priority mailbox is transmitted first 1: Lowest number mailbox is transmitted first
3	MMOD	Monitor mode 0: Disable Monitor mode 1: Enable Monitor mode
2:0	Reserved	Must be kept at reset value.

23.5.3. Timer register (CAN_TIMER)

Address offset: 0x08
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Counter value This bit field contains the internal counter value used for timestamp generation.

23.5.4. Receive mailbox public filter register (CAN_RMPUBF)

Address offset: 0x10
Reset value: 0xXXXX XXXX

This register is located in RAM.

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MFD31	MFD30	MFD29	MFD28	MFD27	MFD26	MFD25	MFD24	MFD23	MFD22	MFD21	MFD20	MFD19	MFD18	MFD17	MFD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFD15	MFD14	MFD13	MFD12	MFD11	MFD10	MFD9	MFD8	MFD7	MFD6	MFD5	MFD4	MFD3	MFD2	MFD1	MFD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	MFDx	Mailbox filter data MFD31 bit is used to filter the mailbox descriptor RTR field. MFD30 bit is used to filter the mailbox descriptor IDE field. MFDx (x = 0..28) bits are used to filter the mailbox descriptor ID field. 0: The bit is "don't care" 1: The bit is checked

23.5.5. Error register 0 (CAN_ERR0)

Address offset: 0x1C
Reset value: 0x0000 0000

All bits of this register are read-only except in Inactive mode.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFCNT[7:0]								TEFCNT[7:0]							
rw0								rw0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RECNT[7:0]								TECNT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	REFCNT[7:0]	Receive error counter for data phase of FD frames with BRS bit set This bit field can only be written as zero in Inactive mode.
23:16	TEFCNT[7:0]	Transmit error count for the data phase of FD frames with BRS bit set This bit field can only be written as zero in Inactive mode.
15:8	RECNT[7:0]	Receive error count defined by the CAN standard
7:0	TECNT[7:0]	Transmit error count defined by the CAN standard

23.5.6. Error register 1 (CAN_ERR1)

Address offset: 0x20
Reset value: 0x000X 000X

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRFERR	BDFERR	Reserved	CRCFERR	FMFERR	STFFERR	Reserved				ERROVR	ERRFSF	BORF	SYN	TWERRIF	RWERRIF
rc	rc		rc	rc	rc					rc_w1	rc_w1	rc_w1	r	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRERR	BDERR	ACKERR	CRCERR	FMERR	STFERR	TWERRF	RWERRF	IDLEF	TS	ERRSI[1:0]		RS	BOF	ERRSF	Reserved
rc	rc	rc	rc	rc	rc	r	r	r	r	r		r	rc_w1	rc_w1	

Bits	Fields	Descriptions
31	BRFERR	Bit recessive error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: At least one bit sent as recessive is received as dominant
30	BDFERR	Bit dominant error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: At least one bit sent as dominant is received as recessive
29	Reserved	Must be kept at reset value.
28	CRCFERR	CRC error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: A CRC error occurred
27	FMFERR	Form error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: A form error occurred
26	STFFERR	Stuff error in data phase of FD frames with the BRS bit set 0: No error occurrence 1: A stuff error occurred
25:22	Reserved	Must be kept at reset value.
21	ERROVR	Error overrun This bit indicates that an error condition occurred when any error flag is already set. 0: Error overrun not occurred 1: Error overrun occurred
20	ERRFSF	Error summary flag for data phase of FD frames with BRS bit set This bit is logical ORed by the following bits: CAN_ERR1[31]: Bit recessive error CAN_ERR1[30]: Bit dominant error CAN_ERR1[28]: CRC error CAN_ERR1[27]: Form error CAN_ERR1[26]: Stuff error
19	BORF	Bus off recovery flag This bit is set when the the recovery sequence specified in the CAN standard on the CAN bus is detected and CAN is ready to recovery from Bus off. 0: No event occurrence 1: Bus off recovery sequence event occurs
18	SYN	Synchronization flag 0: Not synchronized to the CAN bus 1: Synchronized to the CAN bus
17	TWERRIF	Tx error warning interrupt flag

This bit is not used during Bus off state.
0: No event occurrence
1: TWERRF bit in CAN_ERR1 register changes from 0 to 1

16 RWERRIF
Rx error warning interrupt flag
This bit is updated when exiting from Pretended Networking mode.
0: No event occurrence
1: RWERRF bit in CAN_ERR1 register changes from 0 to 1

15 BRERR
Bit recessive error for all format frames
This bit is updated when exiting from Pretended Networking mode.
0: No error occurrence
1: At least one bit sent as recessive is received as dominant

14 BDERR
Bit dominant error for all format frames
This bit is updated when exiting from Pretended Networking mode.
0: No error occurrence
1: At least one bit sent as dominant is received as recessive

13 ACKERR
ACK error
This bit is updated when exiting from Pretended Networking mode.
0: No error occurrence
1: An ACK error occurred

12 CRCERR
CRC error
This bit is updated when exiting from Pretended Networking mode.
0: No error occurrence
1: A CRC error occurred

11 FMERR
Form error
This bit is updated when exiting from Pretended Networking mode.
0: No error occurrence
1: A form error occurred

10 STFERR
Stuff error
This bit is updated when exiting from Pretended Networking mode.
0: No error occurrence
1: A stuffing error occurred

9 TWERRF
Tx error warning flag
0: No event occurrence
1: TECNT[7:0] in CAN_ERR0 register is greater than or equal to 96

8 RWERRF
Rx error warning flag
This bit is updated when exiting from Pretended Networking mode.
0: No event occurrence.
1: RECNT[7:0] in CAN_ERR0 register is greater than or equal to 96

7 IDLEF
IDLE flag
0: No event occurrence

		1: In Bus idle state
6	TS	Transmitting state 0: CAN is not working in transmitting state 1: CAN is working in transmitting state
5:4	ERRSI[1:0]	Error state indicator When MMOD bit in CAN_CTL1 register and SWRST bit in CAN_CTL0 register are both set to 1, this bit will be reset for one CAN bit time, and then changes to 0b01 to reflect Monitor mode state. 00: Error active 01: Error passive 1x: Bus off
3	RS	Receiving state 0: CAN is not working in receiving state 1: CAN is working in receiving state
2	BOF	Bus off flag 0: No event occurrence 1: In Bus off state
1	ERRSF	Error summary flag This bit is logical ORed by the following bits: CAN_ERR1[15]: Bit recessive error CAN_ERR1[14]: Bit dominant error CAN_ERR1[13]: ACK error CAN_ERR1[12]: CRC error CAN_ERR1[11]: Form error CAN_ERR1[10]: Stuff error
0	Reserved	Must be kept at reset value.

23.5.7. Interrupt enable register (CAN_INTEN)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MIE31	MIE30	MIE29	MIE28	MIE27	MIE26	MIE25	MIE24	MIE23	MIE22	MIE21	MIE20	MIE19	MIE18	MIE17	MIE16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIE15	MIE14	MIE13	MIE12	MIE11	MIE10	MIE9	MIE8	MIE7	MIE6	MIE5	MIE4	MIE3	MIE2	MIE1	MIE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions

31:0	MIEx	Message transmission and reception interrupt enable When Rx FIFO is disabled, these bits are used for mailbox number x (refers to ***Mailbox number***) interrupt configuration. When Rx FIFO is enabled, MIE5 to MIE7 are used for Rx FIFO interrupt configuration, and mailbox interruption configuration bits are the bits x that are the same with the mailbox number x (refers to ***Mailbox number***). 0: Disable the corresponding interrupt 1: Enable the corresponding interrupt

23.5.8. Status register (CAN_STAT)

Address offset: 0x30
Reset value: 0x0000 0000

The bits 1 to 7 of this register will be cleared by configuration change of RFEN bit in CAN_CTL0 register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MS31	MS30	MS29	MS28	MS27	MS26	MS25	MS24	MS23	MS22	MS21	MS20	MS19	MS18	MS17	MS16
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS15	MS14	MS13	MS12	MS11	MS10	MS9	MS8	MS7_RFO	MS6_RFW	MS5_RFNE	MS4_RES	MS3_RES	MS2_RES	MS1_RES	MS0_RFC
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:8	MSx	Mailbox x state x is the mailbox number, refers to ***Mailbox number***. 0: No successful transmission or reception has occurred in the mailbox descriptor 1: A successful transmission or reception has occurred in the mailbox descriptor
7	MS7_RFO	Mailbox 7 state / Rx FIFO overflow 0: No successful transmission or reception has occurred in the mailbox descriptor 7 when Rx FIFO is disabled. / Rx FIFO is not overflow when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 7 when Rx FIFO is disabled. / Rx FIFO is overflow when Rx FIFO is enabled.
6	MS6_RFW	Mailbox 6 state / Rx FIFO warning 0: No successful transmission or reception has occurred in the mailbox descriptor 6 when Rx FIFO is disabled. / Rx FIFO has no warning when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 6 when Rx FIFO is disabled. / Rx FIFO almost full warning when Rx FIFO is enabled.
5	MS5_RFNE	Mailbox 5 state / Rx FIFO not empty 0: No successful transmission or reception has occurred in the mailbox descriptor

		5 when Rx FIFO is disabled. / Rx FIFO is empty when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 5 when Rx FIFO is disabled. / Rx FIFO is not empty when Rx FIFO is enabled.
4	MS4_RES	Mailbox 4 state / Reserved Similar to MS1_RES description.
3	MS3_RES	Mailbox 3 state / Reserved Similar to MS1_RES description.
2	MS2_RES	Mailbox 2 state / Reserved Similar to MS1_RES description.
1	MS1_RES	Mailbox 1 state / Reserved 0: No successful transmission or reception has occurred in the mailbox descriptor 1 when Rx FIFO is disabled. / Reserved when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 1 when Rx FIFO is disabled. / Reserved when Rx FIFO is enabled.
0	MS0_RFC	Mailbox 0 state / Clear Rx FIFO bit 0: No successful transmission or reception has occurred in the mailbox descriptor 0 when Rx FIFO is disabled. / No effect when Rx FIFO is enabled. 1: A successful transmission or reception has occurred in the mailbox descriptor 0 when Rx FIFO is disabled. / Clear Rx FIFO when Rx FIFO is enabled, only allowed to written in Inactive mode, refers to ***Clear FIFO***.

23.5.9. Control register 2 (CAN_CTL2)

Address offset: 0x34
Reset value: 0x00A0 0000

All bits except bit 31, 30 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

All bits of this register are not reset by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRFSIE	BORIE	Reserved		RFFN[3:0]				ASD[4:0]					RFO	RRFRMS	IDERTR_RMF
rw	rw			rw				rw					rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITSRC	PREEN	Reserved	ISO	EFDIS	Reserved										
rw	rw		rw	rw											

Bits	Fields	Descriptions
31	ERRFSIE	Error summary interrupt enable bit for data phase of FD frames with BRS bit set 0: Disable error summary interrupt for data phase of FD frames with BRS bit set

1: Enable error summary interrupt for data phase of FD frames with BRS bit set

30 BORIE

Bus off recovery interrupt enable
0: Disable Bus off recovery interrupt
1: Enable Bus off recovery interrupt

29:28 Reserved

Must be kept at reset value.

27:24 RFFN[3:0]

Rx FIFO filter number

Table 23-12. Rx FIFO filter element number

RFFN[3:0]	Rx FIFO filter element number	Rx FIFO occupied space	Available mailboxes
0000	8	Mailbox descriptor 0 - 7	Mailbox 8 - 31
0001	16	Mailbox descriptor 0 - 9	Mailbox 10 - 31
0002	24	Mailbox descriptor 0 - 11	Mailbox 12 - 31
0003	32	Mailbox descriptor 0 - 13	Mailbox 14 - 31
0004	40	Mailbox descriptor 0 - 15	Mailbox 16 - 31
0005	48	Mailbox descriptor 0 - 17	Mailbox 18 - 31
0006	56	Mailbox descriptor 0 – 19	Mailbox 20 - 31
0007	64	Mailbox descriptor 0 – 21	Mailbox 22 - 31
0008	72	Mailbox descriptor 0 – 23	Mailbox 24 - 31
0009	80	Mailbox descriptor 0 – 25	Mailbox 26 - 31
000A	88	Mailbox descriptor 0 – 27	Mailbox 28 - 31
000B	96	Mailbox descriptor 0 – 29	Mailbox 30 - 31
000C	104	Mailbox descriptor 0 – 31	none
others	104	Mailbox descriptor 0 - 31	none

This bit field must not be programmed with values that cause memory occupied by Rx FIFO to exceed the available memory size which is defined by MSZ[4:0] bits in CAN_CTL0 register, otherwise the exceeding ones will not be functional.

23:19 ASD[4:0]

Arbitration start delay
This bit field defines how many CAN bits the Tx arbitration process start point can be delayed.

18 RFO

Receive filter order
0: Rx FIFO is filtered first
1: Mailboxes are filtered first

17 RRFRMS

Remote request frame is stored
0: Remote response frame is generated when a mailbox with CODE RANSWER is found with the same ID
1: Remote request frame is stored as a data frame without automatic remote response frame transmitted

16 IDERTR_RMF

IDE and RTR field filter type for Rx mailbox reception
This bit defines the matching of IDE and RTR field in Rx mailbox descriptor with the

		received bit. 0: IDE field is always compared, and RTR is never compared. Regardless of the filter data configurations in related filter register. 1: Filtering of IDE and RTR fields are enabled, by filter data configurations in related filter register.
15	ITSRC	Internal counter source 0: CAN baudrate 1: External trigger CANx_EX_TIME_TICK from TRIGSEL output
14	PREEN	Protocol exception detection enable by CAN standard 0: Disable protocol exception detection 1: Enable protocol exception detection
13	Reserved	Must be kept at reset value.
12	ISO	ISO CAN FD 0: Non-ISO CAN FD protocol operation is applied 1: ISO CAN FD protocol operation is applied
11	EFDIS	Edge filtering disable 0: Enable edge filtering 1: Disable edge filtering
10:0	Reserved	Must be kept at reset value.

23.5.10. CRC for classical frame register (CAN_CRCC)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											ANTM[4:0]				
											r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CRCTC[14:0]														
	r														

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	ANTM[4:0]	Associated number of mailbox for transmitting the CRCTC[14:0] value This bit field contains the number of the mailbox which transmits the CRCTC[14:0] value.
15	Reserved	Must be kept at reset value.
14:0	CRCTC[14:0]	Transmitted CRC value for classical frames

This bit field contains the CRC value of the last successfully transmitted message in classical format.

23.5.11. Receive FIFO public filter register (CAN_RFIFOPUBF)

Address offset: 0x48
Reset value: 0xXXXX XXXX

This register is located in RAM.

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FFD31	FFD30	FFD29	FFD27	FFD27	FFD26	FFD25	FFD24	FFD23	FFD22	FFD21	FFD20	FFD19	FFD18	FFD17	FFD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFD15	FFD14	FFD13	FFD12	FFD11	FFD10	FFD9	FFD8	FFD7	FFD6	FFD5	FFD4	FFD3	FFD2	FFD1	FFD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FFDx	Rx FIFO filter data Each bit is used for filtering the corresponding ID filter table element bit, except the reserved bit in ID filter table element. 0: The bit is "don't care" 1: The bit is checked

23.5.12. Receive FIFO identifier filter matching number register (CAN_RFIFOIFMN)

Address offset: 0x4C
Reset value: 0xXXXX XXXX

This register has to be accessed by word(32-bit).

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

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.

8:0	IDFMN[8:0]	Identifier filter matching number This field is valid only when MS5_RFNE bit in CAN_STAT register is 1. This bit field indicates which ID filter table element matches the received message that is in the output of the Rx FIFO. If more than one element is matched, the ID filter table element with the lowest number is stored.

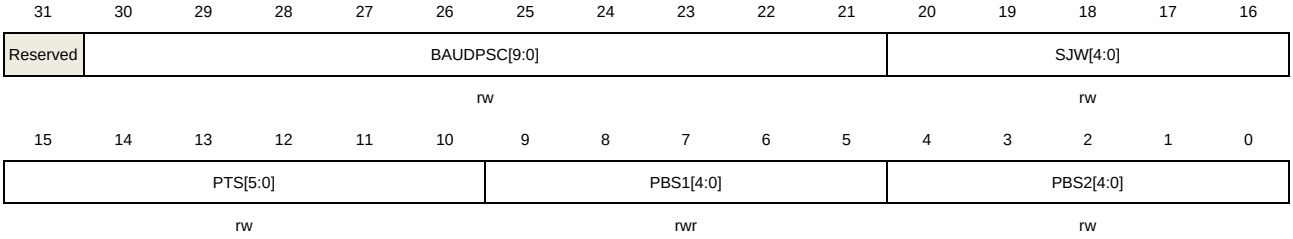
23.5.13. Bit timing register (CAN_BT)

Address offset: 0x50
Reset value: 0x0100 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:21	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler. Baud rate prescaler = BAUDPSC[9:0] + 1
20:16	SJW[4:0]	Resynchronization jump width Resynchronization jump width time quantum = SJW[4:0] + 1
15:10	PTS[5:0]	Propagation time segment Propagation time segment time quantum = PTS[5:0] + 1
9:5	PBS1[4:0]	Phase buffer segment 1 Phase buffer segment 1 time quantum = PBS1[4:0] + 1
4:0	PBS2[4:0]	Phase buffer segment 2 Phase buffer segment 2 time quantum = PBS2[4:0] + 1

23.5.14. Receive FIFO/mailbox private filter x register (CAN_RFIFOMPFx)(x=0..31)

Address offset: 0x880 + 4 × x
Reset value: 0xXXXX XXXX

These register is located in RAM.

All bits of these registers should be configured in Inactive mode only, because they are blocked by hardware in other modes.

These registers are not affected by software reset bit SWRST in CAN_CTL0 register.

These registers have to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FMFD31	FMFD30	FMFD29	FMFD27	FMFD27	FMFD26	FMFD25	FMFD24	FMFD23	FMFD22	FMFD21	FMFD20	FMFD19	FMFD18	FMFD17	FMFD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMFD15	FMFD14	FMFD13	FMFD12	FMFD11	FMFD10	FMFD9	FMFD8	FMFD7	FMFD6	FMFD5	FMFD4	FMFD3	FMFD2	FMFD1	FMFD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FMFDx	FIFO/mailbox filter data If used as mailbox filters, refer to the MFDx bits in CAN_RMPUBF register. If used as Rx FIFO filters, refer to the FFDx bits in CAN_RFIFOPUBF register. 0: The bit is "don't care" 1: The bit is checked

23.5.15. Pretended Networking mode control register 0 (CAN_PN_CTL0)

Address offset: 0xB00
Reset value: 0x0000 0100

All bits except bit 17, 16 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														WTOIE	WMIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NMM[7:0]								Reserved		DATAFT[1:0]		IDFT[1:0]		FFT[1:0]	
rw										rw		rw		rw	

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	WTOIE	Wakeup timeout interrupt enable 0: Disable wakeup timeout interrupt 1: Enable wakeup timeout interrupt
16	WMIE	Wakeup match interrupt enable 0: Disable wakeup match interrupt

		1: Enable wakeup match interrupt
15:8	NMM[7:0]	Number of messages matching times An event counter is used in the wakeup message filter, in which a transistion on the output of event after N input matching events. 00000001: N = 1 00000010: N = 2 …… 11111111: N = 255
7:6	Reserved	Must be kept at reset value.
5:4	DATAFT[1:0]	DATA field filtering type in Pretended Networking mode 00: Only messages with DATA field equal to the expected data field through data filter are matched 01: Messages with DATA field greater than or equal to the expected data low threshold are matched 10: Messages with DATA field smaller than or equal to the expected data high threshold are matched 11: Messages with DATA field greater than or equal to the expected data low threshold, and smaller than or equal to the expected data high threshold are matched
3:2	IDFT[1:0]	ID field filtering type in Pretended Networking mode 00: Only messages with ID field equal to the expected identifier through identifier filter are matched 01: Messages with ID field greater than or equal to the expected identifier low threshold are matched 10: Messages with ID field smaller than or equal to the expected identifier high threshold are matched 11: Messages with ID field greater than or equal to the expected identifier low threshold, and smaller than or equal to the expected identifier high threshold are matched
1:0	FFT[1:0]	Frame filtering type in Pretended Networking mode 00: All fields except DATA field, DLC field are filtered 01: All fields are filtered 10: All fields except DATA field, DLC field are filtered with NMM[7:0] matching times 11: All fields are filtered with NMM[7:0] matching times

23.5.16. Pretended Networking mode timeout register (CAN_PN_TO)

Address offset: 0xB04
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WTO[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	WTO[15:0]	Wakeup timeout The timeout is counted by step of 64 times the CAN Bit Time. Wakeup timeout is default disabled.

23.5.17. Pretended Networking mode status register (CAN_PN_STAT)

Address offset: 0xB08

Reset value: 0x0000 0080

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														WTOS	WMS
														rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMCNT[7:0]								MMCNTS	Reserved						
r								r							

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	WTOS	Wakeup timeout flag status 0: No wakeup timeout event occured 1: Wakeup timeout event occured
16	WMS	Wakeup match flag status 0: No wakeup match event occured 1: Wakeup match event occured
15:8	MMCNT[7:0]	Matching message counter in Pretended Networking mode This bit field indicates the matching message number during Pretended Networking mode. These bits are cleared when node enters Pretended Networking mode, they are not affected by software reset.
7	MMCNTS	Matching message counter state This bit is set to 1 to show the value of MMCNT[7:0] is valid. 0: Matching message counter MMCNT[7:0] is updating

		1: Matching message counter MMCNT[7:0] is valid
6:0	Reserved	Must be kept at reset value.

23.5.18. Pretended Networking mode expected identifier 0 register (CAN_PN_EID0)

Address offset: 0xB0C
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	EIDE	ERTR	EID_ELT[28:16]												
	rw	rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EID_ELT[15:0]															
rw															

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	EIDE	Expected IDE in Pretended Networking mode 0: Standard frame format 1: Extended frame format
29	ERTR	Expected RTR in Pretended Networking mode 0: Data frame 1: Remote frame
28:0	EIDF_ELT[28:0]	Expected ID field / expected ID low threshold in Pretended Networking mode This bit field is used as expected ID field when IDFT[1:0] bit field in CAN_PN_CTL0 register is 0 / 1 / 2, or is used as expected ID low threshold when IDFT[1:0] bit field is 3. For extended frame format, all 29 bits are used. For standard frame format, bits 18 to 28 are used.

23.5.19. Pretended Networking mode expected DLC register (CAN_PN_EDLC)

Address offset: 0xB10
Reset value: 0x0000 0008

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												DLCELT[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DLCEHT[3:0]			
												rw			

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	DLCELT[3:0]	DLC expected low threshold in Pretended Networking mode
15:4	Reserved	Must be kept at reset value.
3:0	DLCEHT[3:0]	DLC expected high threshold in Pretended Networking mode

23.5.20. Pretended Networking mode expected data low 0 register (CAN_PN_EDL0)

Address offset: 0xB14
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0ELT[7:0]								DB1ELT[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2ELT[7:0]								DB3ELT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB0ELT[7:0]	Data byte 0 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
23:16	DB1ELT[7:0]	Data byte 1 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
15:8	DB2ELT[7:0]	Data byte 2 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
7:0	DB3ELT[7:0]	Data byte 3 expected low threshold in Pretended Networking mode This bit field is used as expected DATA field when DATAFT[1:0] bit field in CAN_PN_CTL0 register is 0 / 1 / 2, or is used as expected DATA low threshold

when DATAFT[1:0] bit field is 3.

23.5.21. Pretended Networking mode expected data low 1 register (CAN_PN_EDL1)

Address offset: 0xB18
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB4ELT[7:0]								DB5ELT[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6ELT[7:0]								DB7ELT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB4ELT[7:0]	Data byte 4 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
23:16	DB5ELT[7:0]	Data byte 5 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
15:8	DB6ELT[7:0]	Data byte 6 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.
7:0	DB7ELT[7:0]	Data byte 7 expected low threshold in Pretended Networking mode Refer to DB3ELT[7:0] descriptions.

23.5.22. Pretended Networking mode identifier filter / expected identifier 1 register (CAN_PN_IFEID1)

Address offset: 0x B1C
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	IDEFD	RTRFD	IDFD_EHT[28:16]												
	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IDFD_EHT[15:0]
rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	IDEFD	IDE filter data in Pretended Networking mode 0: The bit is "don't care" 1: The bit is checked
29	RTRFD	RTR filter data in Pretended Networking mode 0: The bit is "don't care" 1: The bit is checked
28:0	IDFD_EHT[28:0]	ID filter data / ID expected high threshold in Pretended Networking mode **ID filter data** (when IDFT[1:0] bit field in CAN_PN_CTL0 register is 0): 0: The bit is "don't care" 1: The bit is checked **ID expected high threshold** (when IDFT[1:0] bit field is 3). **Bits reserved** (when IDFT[1:0] bit field is 1 or 2). For extended frame format, all 29 bits are used. For standard frame format, bits 18 to 28 are used.

23.5.23. Pretended Networking mode data 0 filter / expected data high 0 register (CAN_PN_DF0EDH0)

Address offset: 0xB20
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB0FD_EHT[7:0]								DB1FD_EHT[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB2FD_EHT[7:0]								DB3FD_EHT[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB0FD_EHT[7:0]	Data byte 0 filter data / Data byte 0 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
23:16	DB1FD_EHT[7:0]	Data byte 1 filter data / Data byte 1 expected high threshold in Pretended

		Networking mode Refer to DB3FD_EHT[7:0] descriptions.
15:8	DB2FD_EHT[7:0]	Data byte 2 filter data / Data byte 2 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
7:0	DB3FD_EHT[7:0]	Data byte 3 filter data / Data byte 2 expected high threshold in Pretended Networking mode **Data byte 3 filter data** (when DATAFT[1:0] bit field in CAN_PN_CTL0 register is 0): 0: The bit is "don't care" 1: The bit is checked **Data byte 3 expected high threshold** (when DATAFT[1:0] bit field is 3). **Bits reserved** (when DATAFT[1:0] bit field is 1 or 2).

23.5.24. Pretended Networking mode data 1 filter / expected data high 1 register (CAN_PN_DF1EDH1)

Address offset: 0xB24
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB4FD_HTF[7:0]								DB5FD_HTF[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB6FD_HTF[7:0]								DB7FD_HTF[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB4FD_HTF[7:0]	Data byte 4 filter data / Data byte 4 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
23:16	DB5FD_HTF[7:0]	Data byte 5 filter data / Data byte 5 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
15:8	DB6FD_HTF[7:0]	Data byte 6 filter data / Data byte 6 expected high threshold in Pretended Networking mode Refer to DB3FD_EHT[7:0] descriptions.
7:0	DB7FD_HTF[7:0]	Data byte 7 filter data / Data byte 7 expected high threshold in Pretended

Networking mode

Refer to DB3FD_EHT[7:0] descriptions.

23.5.25. Pretended Networking mode received wakeup mailbox x control status information register (CAN_PN_RWMxCS)(x=0..3)

Address offset: 0xB40 + 16 * x

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									RSRR	RIDE	RRTR	RDLC[3:0]			
									r	r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	RSRR	Received SRR bit
21	RIDE	Received IDE bit 0: Frame format is standard 1: Frame format is extended
20	RRTR	Received RTR bit 0: Frame is data frame 1: Frame is remote frame
19:16	RDLC[3:0]	Received DLC bits The bit field indicates the valid data byte length.
15:0	Reserved	Must be kept at reset value.

23.5.26. Pretended Networking mode received wakeup mailbox x identifier register (CAN_PN_RWMxI)(x=0..3)

Address offset: 0xB44 + 16 * x

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

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

RID[15:0]

r

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:0	RID[28:16]	Received ID bits For extended frame format, all 29 bits are used for ID storage. For standard frame format, bits 18 to 28 are used for ID storage.

23.5.27. Pretended Networking mode received wakeup mailbox x data 0 register (CAN_PN_RWMxD0)(x=0..3)

Address offset: 0xB48 + 16 * x
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB0[7:0]								RDB1[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB2[7:0]								RDB3[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	RDB0[7:0]	Received data byte 0
23:16	RDB1[7:0]	Received data byte 1
15:8	RDB2[7:0]	Received data byte 2
7:0	RDB3[7:0]	Received data byte 3

23.5.28. Pretended Networking mode received wakeup mailbox x data 1 register (CAN_PN_RWMxD1)(x=0..3)

Address offset: 0xB4C + 16 * x
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB4[7:0]								RDB5[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB6[7:0]								RDB7[7:0]							

r r

Bits	Fields	Descriptions
31:24	RDB4[7:0]	Received data byte 4
23:16	RDB5[7:0]	Received data byte 5
15:8	RDB6[7:0]	Received data byte 6
7:0	RDB7[7:0]	Received data byte 7

23.5.29. FD control register (CAN_FDCTL)

Address offset: 0xC00
Reset value: 0x8000 0101

Bits 17:16, 15, 12:8 of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRSEN	Reserved													MDSZ[1:0]	
rw														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDCEN	TDCS	Reserved	TDCO[4:0]					Reserved		TDCV[5:0]					
rw	rc_w1		rw							r					

Bits	Fields	Descriptions
31	BRSEN	Bit rate of data switch enable 0: Bit rate not switch 1: The bit rate shall be switched from the nominal bit rate to the preconfigured data bit rate during the data phase when BRS bit in Tx mailbox is recessive ‘1’
30:18	Reserved	Must be kept at reset value.
17:16	MDSZ[1:0]	Mailbox data size 00: 8 bytes per mailbox 01: 16 bytes per mailbox 10: 32 bytes per mailbox 11: 64 bytes per mailbox
15	TDCEN	Transmitter delay compensation enable **Note:** Transmitter delay compensation must be disabled when loopback and silent mode is enabled. 0: Transmitter delay compensation is disabled 1: Transmitter delay compensation is enabled

14	TDCS	Transmitter delay compensation status When this bit is set, the transmitter delay is out of compensation range, it is unable to compensate the transmitter delay for bit check. 0: Transmitter delay is in compensation range 1: Transmitter delay is out of compensation range
13	Reserved	Must be kept at reset value.
12:8	TDCO[4:0]	Transmitter delay compensation offset These bits are set to the transmitter delay compensation offset value which defines the distance between the measured delay from CANTX to CANRX and the second sample point for CAN FD frames with BRS bit set.
7:6	Reserved	Must be kept at reset value.
5:0	TDCV[5:0]	Transmitter delay compensation value These bits are set by hardware to display the summary of the measured transmitter delay value and the transmitter delay compensation offset.

23.5.30. FD bit timing register (CAN_FDBT)

Address offset: 0xC04
Reset value: 0x0000 0000

All bits of this register should be configured in Inactive mode only, because they are blocked by hardware in other modes.

This register is not affected by software reset bit SWRST in CAN_CTL0 register.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DBAUDPSC[9:0]										Reserved	DSJW[2:0]		
		rw											rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DPTS[4:0]					Reserved		DPBS1[2:0]			Reserved		DPBS2[2:0]		
	rw							rw					rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:20	DBAUDPSC[9:0]	Baud rate prescaler for data bit time The CAN data bit time baud rate prescaler. Baud rate prescaler for data bit time = DBAUDPSC[9:0] + 1
19	Reserved	Must be kept at reset value.
18:16	DSJW[2:0]	Resynchronization jump width for data bit time Resynchronization jump width time quantum = DSJW[2:0] + 1

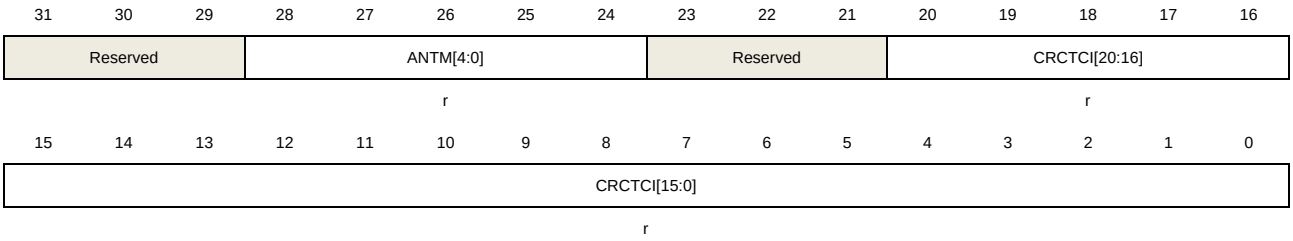
15	Reserved	Must be kept at reset value.
14:10	DPTS[4:0]	Propagation time segment for data bit time Propagation time segment time quantum = DPTS[4:0]
9:8	Reserved	Must be kept at reset value.
7:5	DPBS1[2:0]	Phase buffer segment 1 for data bit time Phase buffer segment 1 time quantum = DPBS1[2:0] + 1
4:3	Reserved	Must be kept at reset value.
2:0	DPBS2[2:0]	Phase buffer segment 2 for data bit time Phase buffer segment 2 time quantum = DPBS2[2:0] + 1

23.5.31. CRC for classical and FD frame register (CAN_CRCCFD)

Address offset: 0xC08

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:24	ANTM[4:0]	Associated number of mailbox for transmitting the CRCTCI[20:0] value This bit field contains the number of the mailbox which transmits the CRCTCI[20:0] value for both classical and FD frames.
23:21	Reserved	Must be kept at reset value.
20:0	CRCTCI[20:0]	Transmitted CRC value for classical and ISO / non-ISO FD frames For CRC_15, bits 0 to 14 are used, the other bits are zeros, and the value is the same as the value of CRCTC[14:0] in CAN_CRCC register. For CRC_17, bits 0 to 16 are used, the other bits are zeros. For CRC_21, all 21 bits are used.

24. Appendix

24.1. List of abbreviations used in register

Table 24-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read/write (rw)	Software can read and write to this bit.
read-only (r)	Software can only read this bit.
write-only (w)	Software can only write to this bit. Reading this bit returns the reset value.
read/clear write 1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
read/clear write 0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
toggle (t)	The software can toggle this bit by writing 1. Writing 0 has no effect.
read/set (rs)	Software can read as well as set this bit to 1. Writing ‘0’ has no effect on the bit value.
read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to ‘0’. Writing ‘0’ has no effect on the bit value.

24.2. List of terms

Table 24-2. List of terms

Glossary	Descriptions
Word	Data of 32-bit length.
Half-word	Data of 16-bit length.
Byte	Data of 8-bit length.
IAP (in-application programming)	Writing 0 has no effect IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
ICP (in-circuit programming)	ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the boot loader while the device is mounted on the user application board.
Option bytes	Product configuration bits stored in the Flash memory.
AHB	Advanced high-performance bus.
APB	Advanced peripheral bus.
RAZ	Read-as-zero.
WI	Writes ignored.
RAZ/WI	Read-as-zero, writes ignored.

24.3. Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

25. Revision history

Table 25-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Sep.15, 2022
1.1	1. Modify the bit field description of ***1.6.9. System configuration register 3 (SYSCFG_CFG3)***. 2. Modify the reset value of ***2.4.1. Wait state register (FMC_WS)***. 3. Modify the data flash size of different flash sizes in section ***2.3.1. Flash memory architecture***. 4. Explain the reset value of the ***2.4.12. EEPROM counter register (FMC_EPCNT)*** in different situations. 5. Modify the description of the ***7.5.4. Trigger selection for ADC1 register (TRIGSEL_ADC1)***.	Dec.19, 2022
1.2	1. Modify V_{REF+} range of the DAC pin from 2.6V to 2.7V in ***Table 15 1. DAC I/O description***. 2. Modify the start address of system memory in ***1.4.Boot configuration***. 3. Add note "Before entering deep sleep mode, set the SLEEP_SLP bit in the FMC_WS register" in ***3.3.4. Power saving modes***. 4. Modify the description of ***2.3.2. Error Checking and Correcting (ECC)***. 5. Modify the description of ***11.4.1. DMA operation***. 6. Modify the description of BOOT0_PF0_RMP bit field in ***1.6.1. System configuration register 0 (SYSCFG_CFG0)*** .	Jun.27, 2023
1.3	1. Add note “The LXTAL is not supported to used as RTC clock source in 48-pin and 32-pin package." in ***17.3. Function overview***. 2. Modify the the relevant formulas and explanations of ***23.3.8. CAN FD operation Transmitter delay compensation*** section. 3. Add the description “When the MFCOM analog SPI works in LSB mode, the 16-bit data received when reading from MFCOM_SBUFBYSx needs to swap the high and low 8-bits of the received data to get the actual data.” in ***9.4.7. Typical configuration of application***. 4. Modify the article 6 of ***12.2. Characteristics***. 5. Modify the ADC\DAC\CMP chapters.	Jan.19, 2024
1.4	1. Modify the description of the TIMERx_IRMP register CI0_RMP field in ***18.2.5. Registers definition (TIMERx, x=1)*** with the value 11 from CKOUT0SEL to CK_OUT.	Aug.16, 2024

Revision No.	Description	Date
	2. Delete the specific parameter of tRSTTEMPO in ***Figure 3 2. Waveform of the POR / PDR***, which should be subject to the datasheet. 3. Add the description of SRAM0 (0KB~16KB) data retention in ***3.3.4. Power saving modes***. 4. Add a note that the same sram block is mapped to different logical addresses in the memory mapping table of ***1.3. Memory map***. 5. Modify the description from "1K byte cache organized as 2X64 bit 64 cache lines" to "1K byte cache organized as 4X64 bit 32 cache lines" in ***2. Flash memory controller (FMC)***. 6. Modify the description of the CAN matching process received in ***23.3.6. Data reception***. 7. Modify " the CMP output must be connected to the corresponding I/O port via the alternate function of the GPIO." to "CMP output can be connected to the corresponding I/O port via the alternate function of the GPIO." in ***22. Comparator (CMP)***. 8. Modify comparator hysteresis to one-sided hysteresis in ***Figure 22 2. CMP hysteresis***. 9. Add a limited relationship between the number of CANCLK clocks and the number of CAN mailboxes for CAN bit time in ***23.3.10. Communication parameters***. 10. In formula (14-1), "V30-Vtemperature" is changed to "Vtemperature-V30". 11. Modify the MFCOM chapter. 12. Add GD32A503xxxx0E, GD32A503xxxx1E boot mode note in ***1.4. Boot configuration***. 13. Modify the number of DMA1 channels in system architecture.	
1.5	1. In ***2. Flash memory controller (FMC)*** and ***Table 1 2. Memory map of GD32A50x devices***, remove the EEPROM related content. 2. Modify TIMERx_CTL0 CAM=2'b10 (downcount only) to TIMERx_CTL0 CAM=2'b01 (downcount only) in ***Figure 18 8. Timing chart of center-aligned counting mode***. 3. Modify the description of bit 19 LVDRSTF in the ***5.3.10. Reset source /clock register (RCU_RSTSCK)*** and add a note. 4. Update ***Figure 3 1. Power supply overview***, modify the power domain and monitoring domain of POR/PDR and BOR from VDD domain to VDDA domain. 5. Update ***3.3.2. VDD / VDDA power domain***, move the	Nov.26, 2024

Revision No.	Description	Date
	description of POR/PDR and BOR from the VDD domain to the VDDA domain. 6. In ***2. Flash memory controller (FMC)***, remove the description related to booting from SRAM. 7. Modify the content of ***1.4. Boot Configuration***, and remove the description related to booting from SRAM. 8. In ***1.6.1. System configuration register 0 (SYSCFG_CFG0)***, modify the description of the BOOT_MODE bit field.	
1.6	1. In section ***23.5.13.Bit timing register (CAN_BT)***, add a description of the BAUDPSC[9:0] baud rate prescaler " Baud rate prescaler = BAUDPSC[9:0] + 1." 2. In section ***23.5.30.FD bit timing register (CAN_FDBT)***, add a description of the DBAUDPSC[9:0] Baud rate prescaler for data bit time " Baud rate prescaler for data bit time = DBAUDPSC[9:0] + 1". 3. In section ***23.3.9. Errors and states*** under the sub-section " Bus integration state " add the note: " Recommend to always enable edge filtering by reset EFDIS bit, to avoid mistakenly detection of bus idle condition." 4. In section ***23.2. Characteristics*** , add “The clock source of Protocol controller (CANCLK) is configurable” and the note about CANCLK: " The CANCLK source selection has restrictions. For details, please refer to the errata manual of the GD32A50x series." 5. Delete descriptions related to the 64K main flash memory: remove ***Table 2 4. Base address and size for 64KB flash memory***, and delete the EFALC[3:0] configuration when the main flash memory is 64k from ***Table 2 7. Option bytes 1***. 6. In section ***9.4.3. Timer***, modify the description from “When the timer start bit is enabled, the timer is compared on the first rising edge of the shift clock and the compare register is reloaded.” to “When the timer start bit is enabled, the timer will reload the compare register on the first rising edge of the shifter clock.”. 7. In ***Table 14 2. ADC input pins definition***, remove "2.7 V ≤ VDDA ≤5V, 2.7 V ≤ VREFP ≤ VDDA, VREFN = VSSA". 8. In ***14.2. Characteristics***, remove “Module supply requirements: 2.7V to 5.5V, and typical power supply voltage is 5V.”. 9. In ***23.3.10. Communication parameters***, modify the description of the internal counter clock source. 10. Modify ***3.Power management unit (PMU)*** description:	Feb.17, 2025

Revision No.	Description	Date
	(1) In ***3.1.Overview*** and ***3.2.Characteristics***: remove content of the backup domain and change the number of power domains to two. (2) In ***Figure 3 1. Power supply overview***: remove SLEEPING, VBAK signals; remove BPOR; add Backup domain, RTC, BREG, LXTAL, and BAP PAD to the VDD domain; move HXTAL to the VDDA domain; change PLLs to PLL. (3) In ***3.3.1.VDD / VDDA power domain***: add backup domain, LXTAL to the VDD domain description and add HXTAL to the VDDA domain and change PLLs to PLL. (4) Modify the description of ***Backup domain***: remove content related to VBAK and BPOR; move the Backup Domain subsection to the VDD Domain subsection. (5) In ***3.3.2. 1.1V power domain***: remove the backup domain. (6) In ***VDDA domain***: change VREF+ / VREF- to VREFP / VREFN.	
1.7	1. In ***20.3.8. I2C master mode***, modify the description of BYTENUM and RELOAD bits from the I2C_CTL0 register to the I2C_CTL1 register. 2. Add the BKP module in ***Figure 1 2. Series system architecture of GD32A50x series***. 3. Update the Important Notice.	Mar.12, 2025
1.8	1.Remove specific value of $t_{RSTTEMPO}$ in ***Figure 3 2. Waveform of the POR / PDR***. 2. Modify the TX pin from "open-drain output mode" to "open-drain mode." in ***19.3.11. Half-duplex communication mode***. 3. Modify the description of the ADDMSK2[2:0] bit field in ***20.4.4. Slave address register 1 (I2C_SADDR1)***. 4. Add the following note: "When the original data programmed is 0xFFFF FFFF FFFF FFFF, the written ECC code is not 0xF. If programming other values again, since the programmed area is not all F, it cannot be written, resulting in the programming error PGERR. The area needs to be erased before performing the programming operation again" in ***2.3.2. Error Checking and Correcting (ECC)***. 5. Add the following note: “The configuration value of the I2S serial clock must be set to less than 1/6 of the PCLK clock (excluding 1/6).” in ***21.4.4.I2S clock***.	Aug.8, 2025

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. the Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, aeronautic or aerospace applications, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); and/or (iii) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

While the Company has implemented advanced security features, the Product may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on Customer's applications and products, and to the maximum extent permitted by applicable law, the Company accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2025 GigaDevice Semiconductor Inc. – All rights reserved