GigaDevice Semiconductor Inc.

GD32F1x0
Arm® Cortex®-M3 32-bit MCU

For GD32F130xx,GD32F150xx

User Manual

Revision 4.1

(Jan. 2026)

Table of Contents

Table of Contents 2
List of Figures 14
List of Tables 20
1. System and memory architecture 22
1.1. ARM Cortex-M3 processor 22
1.2. System architecture 23
1.3. Memory map 24
1.3.1. Bit-banding 26
1.3.2. On-chip SRAM memory 27
1.3.3. On-chip Flash memory 28
1.4. Boot configuration 28
1.5. System configuration registers (SYSCFG) 30
1.5.1. System configuration register 0 (SYSCFG_CFG0) 30
1.5.2. EXTI sources selection register 0 (SYSCFG_EXTISS0) 31
1.5.3. EXTI sources selection register 1 (SYSCFG_EXTISS1) 32
1.5.4. EXTI sources selection register 2 (SYSCFG_EXTISS2) 33
1.5.5. EXTI sources selection register 3 (SYSCFG_EXTISS3) 35
1.5.6. System configuration register 2 (SYSCFG_CFG2) 36
1.6. Device electronic signature 37
1.6.1. Memory density information 37
1.6.2. Unique device ID (96 bits) 37
2. Flash memory controller (FMC) 39
2.1. Overview 39
2.2. Characteristics 39
2.3. Function overview 39
2.3.1. Flash memory architecture 39
2.3.2. Read operations 40
2.3.3. Unlock the FMC_CTL register 40
2.3.4. Page erase 40
2.3.5. Mass erase 41
2.3.6. Main flash programming 42
2.3.7. Option bytes erase 44
2.3.8. Option bytes programming 45
2.3.9. Option bytes description 45
2.3.10. Page erase/Program protection 46
2.3.11. Security protection 47

2.4. Register definition 48
2.4.1. Wait state register (FMC_WS) 48
2.4.2. Unlock key register (FMC_KEY) 48
2.4.3. Option bytes unlock key register (FMC_OBKEY) 49
2.4.4. Status register (FMC_STAT) 49
2.4.5. Control register (FMC_CTL) 50
2.4.6. Address register (FMC_ADDR) 51
2.4.7. Option bytes status register (FMC_OBSTAT) 52
2.4.8. Write protection register (FMC_WP) 52
2.4.9. Wait state enable register (FMC_WSEN) 53
2.4.10. Product ID register (FMC_PID) 53
3. Power management unit (PMU) 55
3.1. Overview 55
3.2. Characteristics 55
3.3. Function overview 55
3.3.1. Backup domain 57
3.3.2. V_{DD} / V_{DDA} power domain 57
V_{DD} domain 58
3.3.3. V_{CORE} power domain 59
3.3.4. Power saving modes 60
3.4. Register definition 62
3.4.1. Control register (PMU_CTL) 62
3.4.2. Power control/status register (PMU_CS) 63
4. Reset and clock unit (RCU) 65
4.1. Reset control unit (RCTL) 65
4.1.1. Overview 65
4.1.2. Function overview 65
4.2. Clock control unit (CCTL) 66
4.2.1. Overview 66
4.2.2. Characteristics 68
4.2.3. Function overview 68
4.3. Register definition 72
4.3.1. Control register 0 (RCU_CTL0) 72
4.3.2. Configuration register 0 (RCU_CFG0) 73
4.3.3. Interrupt register (RCU_INT) 77
4.3.4. APB2 reset register (RCU_APB2RST) 80
4.3.5. APB1 reset register (RCU_APB1RST) 81
4.3.6. AHB enable register (RCU_AHBEN) 83
4.3.7. APB2 enable register (RCU_APB2EN) 85
4.3.8. APB1 enable register (RCU_APB1EN) 86
4.3.9. Backup domain control register (RCU_BDCTL) 88

4.3.10. Reset source /clock register (RCU_RSTSCK) 89
4.3.11. AHB reset register (RCU_AHBRST) 91
4.3.12. Configuration register 1 (RCU_CFG1) 92
4.3.13. Configuration register 2 (RCU_CFG2) 93
4.3.14. Control register 1 (RCU_CTL1) 94
4.3.15. APB1 additional enable register (RCU_ADDAPB1EN) 95
4.3.16. APB1 additional reset register (RCU_ADDAPB1RST) 95
4.3.17. Voltage key register (RCU_VKEY) 95
4.3.18. Deep-sleep mode voltage register (RCU_DSV) 96
4.3.19. Power down voltage select register (RCU_PDVSEL) 96

5. Interrupt / event controller (EXTI) 98

5.1. Overview 98

5.2. Characteristics 98

5.3. Interrupts function overview 98

5.4. External interrupt and event block diagram 101

5.5. External interrupt and event function overview 101

5.6. Register definition 104

5.6.1. Interrupt Enable register (EXTI_INTEN) 104
5.6.2. Event enable register (EXTI_EVEN) 104
5.6.3. Rising edge trigger enable register (EXTI_RTEN) 105
5.6.4. Falling edge trigger enable register (EXTI_FTEN) 105
5.6.5. Software interrupt event register (EXTI_SWIEV) 106
5.6.6. Pending register (EXTI_PD) 107

6. General-purpose I/Os (GPIO) 108

6.1. Overview 108

6.2. Characteristics 108

6.3. Function overview 108

6.3.1. GPIO pin configuration 109
6.3.2. Alternate functions (AF) 110
6.3.3. Additional functions 110
6.3.4. Input configuration 110
6.3.5. Output configuration 111
6.3.6. Analog configuration 111
6.3.7. Alternate function (AF) configuration 112
6.3.8. GPIO locking function 113
6.3.9. GPIO single cycle toggle function 113

6.4. Register definition 114

6.4.1. Port control register (GPIOx_CTL, x=A..D,F) 114
6.4.2. Port output mode register (GPIOx_OMODE, x=A..D,F) 115
6.4.3. Port output speed register (GPIOx_OSPD, x=A..D,F) 117

6.4.4. Port pull-up/down register (GPIOx_PUD, x=A..D,F) 119
6.4.5. Port input status register (GPIOx_ISTAT, x=A..D,F) 120
6.4.6. Port output control register (GPIOx_OCTL, x=A..D,F) 121
6.4.7. Port bit operate register (GPIOx_BOP, x=A..D,F) 121
6.4.8. Port configuration lock register (GPIOx_LOCK, x=A, B) 122
6.4.9. Alternate function selected register0 (GPIOx_AFSEL0, x=A, B, C) 123
6.4.10. Alternate function selected register1 (GPIOx_AFSEL1, x=A,B,C) 124
6.4.11. Bit clear register (GPIOx_BC, x=A..D,F) 125

7. Cyclic redundancy checks management unit (CRC) 126

7.1. Overview 126

7.2. Characteristics 126

7.3. Function overview 127

7.4. Register definition 129

7.4.1. Data Register (CRC_DATA) 129
7.4.2. Free Data Register (CRC_FDATA) 129
7.4.3. Control Register (CRC_CTL) 130
7.4.4. Initialization Data Register (CRC_IDATA) 130

8. Direct memory access controller (DMA) 132

8.1. Overview 132

8.2. Characteristics 132

8.3. Block diagram 133

8.4. Function overview 133

8.4.1. DMA operation 133
8.4.2. Peripheral handshake 135
8.4.3. Arbitration 135
8.4.4. Address generation 136
8.4.5. Circular mode 136
8.4.6. Memory to memory mode 136
8.4.7. Channel configuration 136
8.4.8. Interrupt 137
8.4.9. DMA request mapping 137

8.5. Register definition 141

8.5.1. Interrupt flag register (DMA_INTF) 141
8.5.2. Interrupt flag clear register (DMA_INTC) 141
8.5.3. Channel x control register (DMA_CHxCTL) 142
8.5.4. Channel x counter register (DMA_CHxCNT) 144
8.5.5. Channel x peripheral base address register (DMA_CHxPADDR) 145
8.5.6. Channel x memory base address register (DMA_CHxMADDR) 145

9. Debug (DBG) 147

9.1. Overview 147
9.2. Serial Wire Debug port overview 147
9.2.1. Pin assignment 147
9.2.2. JEDEC-106 ID code 147
9.3. Debug hold function overview 148
9.3.1. Debug support for power saving mode 148
9.3.2. Debug support for TIMER, I2C, RTC, WWDGT and FWDGT 148
9.4. DBG registers 149
9.4.1. ID code register (DBG_ID) 149
9.4.2. Control register 0(DBG_CTL0) 149
9.4.3. Control register 1 (DBG_CTL1) 151
10. Analog to digital converter (ADC) 153
10.1. Overview 153
10.2. Characteristics 153
10.3. Pins and internal signals 154
10.4. Function overview 155
10.4.1. Foreground calibration function 155
10.4.2. Dual clock domain architecture 156
10.4.3. ADCON enable 156
10.4.4. Routine sequence 156
10.4.5. Operation modes 156
10.4.6. Conversion result threshold monitor function 159
10.4.7. Data storage mode 159
10.4.8. Sample time configuration 160
10.4.9. External trigger configuration 161
10.4.10. DMA request 161
10.4.11. ADC internal channels 161
10.4.12. Battery voltage monitoring 162
10.4.13. ADC interrupts 162
10.5. Register definition 163
10.5.1. Status register (ADC_STAT) 163
10.5.2. Control register 0 (ADC_CTL0) 163
10.5.3. Control register 1 (ADC_CTL1) 165
10.5.4. Sampling time register 0 (ADC_SAMPT0) 166
10.5.5. Sampling time register 1 (ADC_SAMPT1) 167
10.5.6. Watchdog high threshold register (ADC_WDHT) 168
10.5.7. Watchdog low threshold register (ADC_WDLT) 169
10.5.8. Routine sequence register 0 (ADC_RSQ0) 169
10.5.9. Routine sequence register 1 (ADC_RSQ1) 170
10.5.10. Routine sequence register 2 (ADC_RSQ2) 170
10.5.11. Routine data register (ADC_RDATA) 171

11. Digital-to-analog converter (DAC) 171
11.1. Overview 171
11.2. Characteristics 172
11.3. Function description 173
11.3.1. DAC enable 173
11.3.2. DAC output buffer 173
11.3.3. DAC data configuration 173
11.3.4. DAC trigger 173
11.3.5. DAC conversion 174
11.3.6. DAC output voltage 174
11.3.7. DMA request 174
11.4. DAC register 175
11.4.1. DACx control register 0 (DAC_CTL0) 175
11.4.2. DACx software trigger register (DAC_SWT) 176
11.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH) 176
11.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH) 177
11.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH) 177
11.4.6. DACx_OUT0 data output register (DAC_OUT0_DO) 178
11.4.7. DACx status register 0 (DAC_STAT0) 178
12. Comparator (CMP) 179
12.1. Overview 179
12.2. Characteristics 179
12.3. Function overview 179
12.3.1. CMP clock 180
12.3.2. CMP I / O configuration 180
12.3.3. CMP operating mode 181
12.3.4. CMP windows mode 181
12.3.5. CMP hysteresis 182
12.3.6. CMP register write protection 182
12.3.7. CMP interrupt 182
12.4. Register definition 183
12.4.1. CMP Control/status register (CMP_CS) 183
13. Watchdog timer (WDGT) 187
13.1. Free watchdog timer (FWDGT) 187
13.1.1. Overview 187
13.1.2. Characteristics 187
13.1.3. Function overview 187
13.1.4. Register definition 190
13.2. Window watchdog timer (WWDGT) 194
13.2.1. Overview 194

13.2.2. Characteristics 194
13.2.3. Function overview 194
13.2.4. Register definition 197
14. Real-time clock(RTC) 199
14.1. Overview 199
14.2. Characteristics 199
14.3. Function overview 200
14.3.1. Block diagram 200
14.3.2. Clock source and prescalers 200
14.3.3. Shadow registers introduction 201
14.3.4. Configurable and field maskable alarm 201
14.3.5. RTC initialization and configuration 202
14.3.6. Calendar reading 203
14.3.7. Resetting the RTC 204
14.3.8. RTC shift function 204
14.3.9. RTC reference clock detection 205
14.3.10. RTC smooth digital calibration 206
14.3.11. Time-stamp function 207
14.3.12. Tamper detection 208
14.3.13. Calibration clock output 209
14.3.14. Alarm output 209
14.3.15. RTC power saving mode management 209
14.3.16. RTC interrupts 210
14.4. Register definition 211
14.4.1. Time register (RTC_TIME) 211
14.4.2. Date register (RTC_DATE) 211
14.4.3. Control register (RTC_CTL) 212
14.4.4. Status register (RTC_STAT) 214
14.4.5. Prescaler register (RTC_PSC) 216
14.4.6. Alarm 0 time and date register (RTC_ALRM0TD) 216
14.4.7. Write protection key register (RTC_WPK) 218
14.4.8. Sub second register (RTC_SS) 218
14.4.9. Shift function control register (RTC_SHIFTCTL) 218
14.4.10. Time of time stamp register (RTC_TTS) 219
14.4.11. Date of time stamp register (RTC_DTS) 220
14.4.12. Sub second of time stamp register (RTC_SSTS) 221
14.4.13. High resolution frequency compensation register (RTC_HRFC) 221
14.4.14. Tamper register (RTC_TAMP) 222
14.4.15. Alarm 0 sub second register (RTC_ALRM0SS) 224
14.4.16. Backup registers (RTC_BKPx) (x=0..4) 225
15. Timer (TIMERx) 227
15.1. Advanced timer (TIMERx,x=0) 228

15.1.1. Overview 228
15.1.2. Characteristics 228
15.1.3. Block diagram 228
15.1.4. Function overview 229
15.1.5. TIMERx registers(x=0) 256
15.2. General level0 timer (TIMERx, x=1, 2) 283
15.2.1. Overview 283
15.2.2. Characteristics 283
15.2.3. Block diagram 283
15.2.4. Function overview 284
15.2.5. TIMERx registers(x=1, 2) 297
15.3. General level2 timer (TIMERx, x=13) 322
15.3.1. Overview 322
15.3.2. Characteristics 322
15.3.3. Block diagram 322
15.3.4. Function overview 323
15.3.5. TIMERx registers(x=13) 331
15.4. General level3 timer (TIMERx, x=14) 341
15.4.1. Overview 341
15.4.2. Characteristics 341
15.4.3. Block diagram 341
15.4.4. Function overview 342
15.4.5. TIMERx registers(x=14) 358
15.5. General level4 timer (TIMERx, x=15,16) 377
15.5.1. Overview 377
15.5.2. Characteristics 377
15.5.3. Block diagram 377
15.5.4. Function overview 378
15.5.5. TIMERx registers(x=15,16) 392
15.6. Basic timer (TIMERx, x=5) 407
15.6.1. Overview 407
15.6.2. Characteristics 407
15.6.3. Block diagram 407
15.6.4. Function overview 407
15.6.5. TIMERx registers(x=5) 412
16. Infrared ray port (IFRP) 417
16.1. Overview 417
16.2. Characteristics 417
16.3. Function overview 417
17. Universal synchronous asynchronous receiver transmitter (USART) 419

17.1. Overview 419
17.2. Characteristics 419
17.3. Function overview 420
17.3.1. USART frame format 421
17.3.2. Baud rate generation 422
17.3.3. USART transmitter 423
17.3.4. USART receiver 424
17.3.5. Use DMA for data buffer access 425
17.3.6. Hardware flow control 427
17.3.7. Multi-processor communication 428
17.3.8. LIN mode 429
17.3.9. Synchronous mode 430
17.3.10. IrDA SIR ENDEC mode 431
17.3.11. Half-duplex communication mode 432
17.3.12. Smartcard (ISO7816-3) mode 432
17.3.13. ModBus communication 434
17.3.14. Wakeup from deep-sleep mode 435
17.3.15. USART interrupts 435
17.4. Register definition 437
17.4.1. Control register 0 (USART_CTL0) 437
17.4.2. Control register 1 (USART_CTL1) 439
17.4.3. Control register 2 (USART_CTL2) 441
17.4.4. Baud rate generator register (USART_BAUD) 444
17.4.5. Prescaler and guard time configuration register (USART_GP) 445
17.4.6. Receiver timeout register (USART_RT) 446
17.4.7. Command register (USART_CMD) 447
17.4.8. Status register (USART_STAT) 447
17.4.9. Interrupt status clear register (USART_INTC) 451
17.4.10. Receive data register (USART_RDATA) 452
17.4.11. Transmit data register (USART_TDATA) 453
18. Inter-integrated circuit interface (I2C) 454
18.1. Overview 454
18.2. Characteristics 454
18.3. Function overview 454
18.3.1. SDA and SCL lines 455
18.3.2. Data validation 456
18.3.3. START and STOP signal 456
18.3.4. Clock synchronization 456
18.3.5. Arbitration 457
18.3.6. I2C communication flow 457
18.3.7. Programming model 458
18.3.8. SCL line stretching 467

18.3.9. Use DMA for data transfer 468
18.3.10. Packet error checking 468
18.3.11. SMBus support 468
18.3.12. Status, errors and interrupts 470
18.4. Register definition 471
18.4.1. Control register 0 (I2C_CTL0) 471
18.4.2. Control register 1 (I2C_CTL1) 473
18.4.3. Slave address register 0 (I2C_SADDR0) 474
18.4.4. Slave address register 1 (I2C_SADDR1) 474
18.4.5. Transfer buffer register (I2C_DATA) 475
18.4.6. Transfer status register 0 (I2C_STAT0) 475
18.4.7. Transfer status register 1 (I2C_STAT1) 478
18.4.8. Clock configure register (I2C_CKCFG) 479
18.4.9. Rise time register (I2C_RT) 480
19. Serial peripheral interface/Inter-IC sound (SPI/I2S) 480
19.1. Overview 480
19.2. Characteristics 480
19.2.1. SPI characteristics 480
19.2.2. I2S characteristics 481
19.3. SPI function overview 481
19.3.1. SPI block diagram 481
19.3.2. SPI signal description 481
19.3.3. SPI clock timing and data format 482
19.3.4. NSS function 483
19.3.5. SPI operation modes 484
19.3.6. DMA function 488
19.3.7. CRC function 488
19.3.8. SPI interrupts 489
19.4. I2S function overview 491
19.4.1. **I2S block diagram** 491
19.4.2. **I2S signal description** 491
19.4.3. I2S audio standards 491
19.4.4. I2S clock 500
19.4.5. Operation 501
19.4.6. DMA function 505
19.4.7. **I2S interrupts** 505
19.5. Register definition 507
19.5.1. Control register 0 (SPI_CTL0) 507
19.5.2. Control register 1 (SPI_CTL1) 509
19.5.3. Status register (SPI_STAT) 510
19.5.4. Data register (SPI_DATA) 511
19.5.5. CRC polynomial register (SPI_CRCPOLY) 512

19.5.6. RX CRC register (SPI_RCRC) 512
19.5.7. TX CRC register (SPI_TCRC) 513
19.5.8. I2S control register (SPI_I2SCTL) 513
19.5.9. I2S clock prescaler register (SPI_I2SPSC) 515
20. HDMI-CEC controller(HDMI-CEC) 516
20.1. Overview 516
20.2. Characteristics 516
20.3. Function overview 516
20.3.1. CEC bus pin 516
20.3.2. Message description 517
20.3.3. Bit timing description 518
20.3.4. Arbitration 519
20.3.5. SFT option bit description 520
20.3.6. Error definition 520
20.3.7. HDMI-CEC interrupt 523
20.4. Register definition 525
20.4.1. Control register (CEC_CTL) 525
20.4.2. Configuration register (CEC_CFG) 526
20.4.3. Transmit data register (CEC_TDATA) 527
20.4.4. Receive data register (CEC_RDATA) 528
20.4.5. Interrupt Flag Register (CEC_INTF) 528
20.4.6. Interrupt enable register (CEC_INTEN) 530
21. Touch sensing interface (TSI) 533
21.1. Overview 533
21.2. Characteristics 533
21.3. Function Overview 533
21.3.1. TSI block diagram 533
21.3.2. Touch sensing technique overview 533
21.3.3. Charge transfer sequence 534
21.3.4. Charge transfer sequence FSM 536
21.3.5. Clock and duration time of states 538
21.3.6. PIN mode control of TSI 539
21.3.7. Analog switch (ASW) and I/O hysteresis mode 539
21.3.8. TSI operation flow 539
21.3.9. TSI flags and interrupts 540
21.3.10. TSI GPIOs 540
21.4. Registers definition 542
21.4.1. Control register (TSI_CTL) 542
21.4.2. Interrupt enable register (TSI_INTEN) 544
21.4.3. Interrupt flag clear register (TSI_INTC) 544
21.4.4. Interrupt flag register (TSI_INTF) 545

21.4.5. Pin hysteresis mode register (TSI_PHM) 546
21.4.6. Analog switch register (TSI_ASW) 546
21.4.7. Sample configuration register (TSI_SAMPCFG) 547
21.4.8. Channel configuration register (TSI_CHCFG) 547
21.4.9. Group control register (TSI_GCTL) 548
21.4.10. Group x cycle number registers (TSI_GxCYCN) (x = 0…5) 548
22. Universal Serial Bus full-speed device interface (USBD) 550
22.1. Overview 550
22.2. Main features 550
22.3. Block diagram 550
22.4. Signal description 551
22.5. Clock configuration 551
22.6. Function overview 551
22.6.1. USB endpoints 551
22.6.2. Operation procedure 554
22.6.3. USB events and interrupts 557
22.6.4. Operation guide 558
22.7. Register definition 561
22.7.1. USBD control register (USBD_CTL) 561
22.7.2. USBD interrupt flag register (USBD_INTF) 562
22.7.3. USBD status register (USBD_STAT) 563
22.7.4. USBD device address register (USBD_DADDR) 564
22.7.5. USBD buffer address register (USBD_BADDR) 565
22.7.6. USBD endpoint x control and status register (USBD_EPxCS), x=[0..7] 565
22.7.7. USBD endpoint x transmission buffer address register (USBD_EPxTBADDR), x=[0..7] ... 567
22.7.8. USBD endpoint x transmission buffer byte count register (USBD_EPxTBCNT), x=[0..7] .. 567
22.7.9. USBD endpoint x reception buffer address register (USBD_EPxRBADDR), x=[0..7] 568
22.7.10. USBD endpoint x reception buffer byte count register (USBD_EPxRBCNT), x=[0..7] 569
23. Appendix 570
23.1. List of abbreviations used in register 570
23.2. List of terms 570
23.3. Available peripherals 571
24. Revision history 572

List of Figures

Figure 1-1. The structure of the Cortex®-M3 processor 23
Figure 1-2. Series system architecture of GD32F130xx and GD32F150xx devices 24
Figure 2-1. Process of page erase operation 41
Figure 2-2. Process of the mass erase operation 42
Figure 2-3. Process of the word programming operation 44
Figure 3-1. Power supply overview 56
Figure 3-2. Waveform of the POR/PDR 58
Figure 3-3. Waveform of LVD threshold 59
Figure 4-1. The system reset circuit 66
Figure 4-2. Clock tree 67
Figure 4-3. HXTAL clock source 68
Figure 5-1. Block diagram of EXTI. 101
Figure 6-1. Basic structure of a general-pupose I/O 109
Figure 6-2. Basic structure of Input configuration 111
Figure 6-3. Basic structure of Output configuration 111
Figure 6-4. Basic structure of Analog configuration 112
Figure 6-5. Basic structure of Alternate function configuration 113
Figure 7-1. Block Diagram of CRC management unit 127
Figure 8-1. Block diagram of DMA 133
Figure 8-2. Handshake mechanism 135
Figure 8-3. DMA interrupt logic 137
Figure 8-4. DMA request mapping 139
Figure 10-1. ADC module block diagram 155
Figure 10-2. Single operation mode 156
Figure 10-3. Continuous operation mode 157
Figure 10-4. Scan operation mode, continuous disable 158
Figure 10-5. Scan operation mode, continuous enable 158
Figure 10-6. Discontinuous operation mode 159
Figure 10-7. Data storage mode of 12-bit resolution 160
Figure 10-8. Data storage mode of 10-bit resolution 160
Figure 10-9. Data storage mode of 8-bit resolution 160
Figure 10-10. Data storage mode of 6-bit resolution 160
Figure 11-1. DAC block diagram 172
Figure 12-1. CMP block diagram 180
Figure 12-2. CMP hysteresis 182
Figure 13-1. Free watchdog timer block diagram 188
Figure 13-2. Window watchdog timer block diagram 194
Figure 13-3. Window watchdog timer timing diagram 195
Figure 14-1. Block diagram of RTC 200
Figure 15-1. Advanced timer block diagram 229
Figure 15-2. Timing chart of internal clock divided by 1 230

Figure 15-3. Timing chart of PSC value change from 0 to 2 231
Figure 15-4. Timing chart of up counting mode, PSC=0/2 232
Figure 15-5. Timing chart of up counting mode, change TIMERx_CAR on the go 232
Figure 15-6. Timing chart of down counting mode, PSC=0/2 233
Figure 15-7. Timing chart of down counting mode, change TIMERx_CAR on the go 234
Figure 15-8. Center-aligned counter timechart 235
Figure 15-9. Repetition counter timing chart of center-aligned counting mode 236
Figure 15-10. Repetition counter timing chart of up counting mode 236
Figure 15-11. Repetition counter timing chart of down counting mode 237
Figure 15-12. Channel input capture principle 238
Figure 15-13. Channel output compare principle (with complementary output, x=0, 1, 2) 239
Figure 15-14. Channel output compare principle (CH3_O) 239
Figure 15-15. Output-compare under three modes 241
Figure 15-16. EAPWM timechart 242
Figure 15-17. CAPWM timechart 242
Figure 15-18. Channel output complementary PWM with dead-time insertion 245
Figure 15-19. Output behavior in response to a break(The break high active) 246
Figure 15-20. Example of counter operation in quadrature decoder interface mode 247
Figure 15-21. Example of quadrature decoder interface mode with CI0FE0 polarity inverted 247
Figure 15-22. Hall sensor is used to BLDC motor 248
Figure 15-23. Hall sensor timing between two timers 249
Figure 15-24. Restart mode 250
Figure 15-25. Pause mode 250
Figure 15-26. Event mode 251
Figure 15-27. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 252
Figure 15-28. TIMER0 Master/Slave mode timer example 252
Figure 15-29. Triggering TIMER0 with Enable of TIMER2 254
Figure 15-30. Triggering TIMER0 and TIMER2 with TIMER2’s CI0 input 255
Figure 15-31. General Level 0 timer block diagram 284
Figure 15-32. Timing chart of internal clock divided by 1 285
Figure 15-33. Timing chart of PSC value change from 0 to 2 286
Figure 15-34. Timing chart of up counting mode, PSC=0/2 287
Figure 15-35. Timing chart of up counting mode, change TIMERx_CAR on the go. 287
Figure 15-36. Timing chart of down counting mode, PSC=0/2 288
Figure 15-37. Timing chart of down counting mode, change TIMERx_CAR on the go. 289
Figure 15-38. Center-aligned counter timechart 290
Figure 15-39. Channel input capture principle 291
Figure 15-40. Channel output compare principle (x=0,1,2,3) 292
Figure 15-41. Output-compare under three modes 293
Figure 15-42. Timing chart of EAPWM 294
Figure 15-43. Timing chart of CAPWM 294
Figure 15-44. General level2 timer block diagram 322
Figure 15-45. Timing chart of internal clock divided by 1 323

Figure 15-46. Timing chart of PSC value change from 0 to 2 324
Figure 15-47. Timing chart of up counting mode, PSC=0/2 325
Figure 15-48. Timing chart of up counting mode, change TIMERx_CAR on the go 325
Figure 15-49. Channel input capture principle 326
Figure 15-50. Channel output compare principle 327
Figure 15-51. Output-compare under three modes 328
Figure 15-52. PWM mode timechart 329
Figure 15-53. General level3 timer block diagram 342
Figure 15-54. Timing chart of internal clock divided by 1 343
Figure 15-55. Timing chart of PSC value change from 0 to 2 344
Figure 15-56. Timing chart of up counting mode, PSC=0/2 345
Figure 15-57. Timing chart of up counting mode, change TIMERx_CAR on the go 345
Figure 15-58. Repetition counter timing chart of up counting mode 346
Figure 15-59. Channel input capture principle 347
Figure 15-60. Channel output compare principle (with complementary output, x=0) 348
Figure 15-61. Channel output compare principle (CH1_O) 348
Figure 15-62. Output-compare under three modes 350
Figure 15-63. PWM mode timechart 351
Figure 15-64. Complementary output with dead-time insertion. 353
Figure 15-65. Output behavior in response to a break(The break high active) 354
Figure 15-66. Restart mode 355
Figure 15-67. Pause mode 355
Figure 15-68. Event mode 356
Figure 15-69. Single pulse mode TIMERx_CHxCV = 4 TIMERx_CAR=99 357
Figure 15-70. General level4 timer block diagram 378
Figure 15-71. Timing chart of internal clock divided by 1 379
Figure 15-72. Timing chart of PSC value change from 0 to 2 379
Figure 15-73. Timing chart of up counting mode, PSC=0/2 380
Figure 15-74. Timing chart of up counting mode, change TIMERx_CAR on the go 381
Figure 15-75. Repetition counter timing chart of up counting mode 382
Figure 15-76. Channel input capture principle 383
Figure 15-77. Output-compare under three modes 385
Figure 15-78. PWM mode timechart 386
Figure 15-79. Channel output complementary PWM with dead-time insertion 388
Figure 15-80. Output behavior in response to a break(The break high active) 389
Figure 15-81. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 390
Figure 15-82. Basic timer block diagram 407
Figure 15-83. Timing chart of internal clock divided by 1 408
Figure 15-84. Timing chart of PSC value change from 0 to 2 409
Figure 15-85. Timing chart of up counting mode, PSC=0/2 410
Figure 15-86. Timing chart of up counting mode, change TIMERx_CAR on the go 410
Figure 16-1. IFRP output timechart 1 417
Figure 16-2. IFRP output timechart 2 418
Figure 16-3. IFRP output timechart 3 418

Figure 17-1. USART module block diagram 421
Figure 17-2. USART character frame (8 bits data and 1 stop bit) 422
Figure 17-3. USART transmit procedure 424
Figure 17-4. Oversampling method of a receive frame bit (OSB=0) 425
Figure 17-5. Configuration step when using DMA for USART transmission 426
Figure 17-6. Configuration step when using DMA for USART reception 427
Figure 17-7. Hardware flow control between two USARTs 427
Figure 17-8. Hardware flow control 428
Figure 17-9. Break frame occurs during idle state 429
Figure 17-10. Break frame occurs during a frame 430
Figure 17-11. Example of USART in synchronous mode 430
Figure 17-12. 8-bit format USART synchronous waveform (CLEN=1) 431
Figure 17-13. IrDA SIR ENDEC module 431
Figure 17-14. IrDA data modulation 432
Figure 17-15. ISO7816-3 frame format 433
Figure 17-16. USART interrupt mapping diagram 436
Figure 18-1. I2C module block diagram 455
Figure 18-2. Data validation 456
Figure 18-3. START and STOP signal 456
Figure 18-4. Clock synchronization 457
Figure 18-5. SDA line arbitration 457
Figure 18-6. I2C communication flow with 7-bit address 458
Figure 18-7. I2C communication flow with 10-bit address (Master Transmit) 458
Figure 18-8. I2C communication flow with 10-bit address (Master Receive) 458
Figure 18-9. Programming model for slave transmitting (10-bit address mode) 460
Figure 18-10. Programming model for slave receiving (10-bit address mode) 461
Figure 18-11. Programming model for master transmitting (10-bit address mode) 463
Figure 18-12. Programming model for master receiving using Solution A (10-bit address mode) 465
Figure 18-13. Programming model for master receiving mode using solution B (10-bit address mode) 466
Figure 19-1. Block diagram of SPI 481
Figure 19-2. SPI timing diagram in normal mode 482
Figure 19-3. A typical full-duplex connection 485
Figure 19-4. A typical simplex connection (Master: receive, Slave: transmit) 485
Figure 19-5. A typical simplex connection (Master: transmit only, Slave: receive) 486
Figure 19-6. A typical bidirectional connection 486
Figure 19-7. Block diagram of I2S 491
Figure 19-8. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 492
Figure 19-9. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 492
Figure 19-10. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 492
Figure 19-11. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 493
Figure 19-12. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 493
Figure 19-13. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 493

Figure 19-14. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 493
Figure 19-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 494
Figure 19-16. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) ... 494
Figure 19-17. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) ... 494
Figure 19-18. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) ... 494
Figure 19-19. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) ... 494
Figure 19-20. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) ... 495
Figure 19-21. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) ... 495
Figure 19-22. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) ... 495
Figure 19-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) ... 495
Figure 19-24. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 495
Figure 19-25. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 496
Figure 19-26. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 496
Figure 19-27. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 496
Figure 19-28. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 496
Figure 19-29. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 497
Figure 19-30. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 497
Figure 19-31. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 497
Figure 19-32. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 497
Figure19-33. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 497
Figure 19-34. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 498
Figure 19-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 498
Figure 19-36. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 498
Figure19-37. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 498
Figure 19-38. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 498
Figure 19-39. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 498
Figure 19-40. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 499
Figure 19-41. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 499
Figure 19-42. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 499

Figure 19-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 499
Figure 19-44. Block diagram of I2S clock generator 500
Figure 19-45. I2S initialization sequence 502
Figure 19-46. I2S master reception disabling sequence 504
Figure 20-1. HDMI-CEC Controller block diagram 517
Figure 20-2. Message Structure 517
Figure 20-3. Start Bit Timing 518
Figure 20-4. Data Bit Timing 518
Figure 20-5. The process of CEC line arbitration 519
Figure 20-6. The free time of signal 519
Figure 20-7. Erro bit period 520
Figure 20-8. The timing of Bit Period Long Error 522
Figure 20-9. Transmission error detection 523
Figure 21-1. Block diagram of TSI module 533
Figure 21-2. Block diagram of sample pin and channel pin 534
Figure 21-3. Voltage of a sample pin during charge-transfer sequence 536
Figure 21-4. FSM flow of a charge-transfer sequence 537
Figure 22-1. USBD block diagram 550
Figure 22-2. An example with buffer descriptor table usage (USBD_BADDR = 0) 553

List of Tables

Table 1-1. Memory map of GD32F130xx and GD32F150xx devices 25
Table 1-2. Flash module organization 28
Table 1-3. Boot modes 28
Table 2-1. Base address and size for flash memory 39
Table 2-2. Option bytes 45
Table 2-3. OB_WP bit for pages protected 47
Table 3-1. Power saving mode summary 61
Table 4-1. Clock source select 70
Table 4-2. Core domain voltage selected in Deep-sleep mode 71
Table 5-1. NVIC exception types in Cotrex®-M3 98
Table 5-2. Interrupt vector table 99
Table 5-3. EXTI source 102
Table 6-1. GPIO configuration table 108
Table 8-1. DMA transfer operation 134
Table 8-2. Interrupt events 137
Table 8-3. DMA requests for each channel 140
Table 10-1. ADC internal input signals 154
Table 10-2. ADC input pins definition 154
Table 10-3. External trigger source for ADC routine sequence 161
Table 11-1. DAC I/O description 172
Table 11-2. DAC triggers and outputs summary 173
Table 11-3. Triggers of DAC 174
Table 12-1. CMP inputs and outputs summary 181
Table 13-1. Min / max FWDGT timeout period at 40 kHz (IRC40K) 189
Table 13-2. Min-max timeout value at 36 MHz (fPCLK1) 195
Table 14-1. RTC power saving mode management 209
Table 14-2. RTC interrupts control 210
Table 15-1. Timers (TIMERx) are devided into six sorts 227
Table 15-2. Complementary outputs controlled by parameters 244
Table 15-3. Counting direction versus quadrature decode signals 247
Table 15-4. Slave mode example table 249
Table 15-5. Input trigger of Timer0 253
Table 15-6. TIMERx(x=1,2) interconnection 295
Table 15-7. Complementary outputs controlled by parameters 352
Table 15-8. Slave mode example table 354
Table 15-9. TIMERx(x=14) interconnection 357
Table 15-10. Complementary outputs controlled by parameters 387
Table 17-1. USART important pins description 421
Table 17-2. Stop bits configuration 422
Table 17-3. USART interrupt requests 435
Table 18-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips

semiconductors) 455
Table 18-2. Event status flags 470
Table 18-3. Error flags 470
Table 19-1. SPI signal description 481
Table 19-2. NSS function in slave mode 483
Table 19-3. NSS function in master mode 483
Table 19-4. SPI operation modes 484
Table 19-5. SPI interrupt requests 490
Table 19-6. I2S bitrate calculation formulas 500
Table 19-7. Audio sampling frequency calculation formulas 500
Table 19-8. Direction of I2S interface signals for each operation mode 501
Table 19-9. I2S interrupt 506
Table 20-1. Frame Structure 517
Table 20-2. Data Bit Timing parameter table 518
Table 20-3. The relationship betwwen Signal Free Time and precondition 519
Table 20-4. Error Handling Timing parameter table 522
Table 20-5. TERR Timing parameter table 523
Table 20-6. HDMI-CEC Interrupt 523
Table 21-1. Pin and analog switch state in a charge-transfer sequence 535
Table 21-2. Duration time of extend charge state in each cycle 538
Table 21-3. TSI errors and flags 540
Table 21-4. TSI pins 540
Table 22-1. USBD signal description 551
Table 22-2. Double-buffering buffer flag definition 554
Table 22-3. Double buffer usage 554
Table 22-4. Reception status encoding 566
Table 22-5. Endpoint type encoding 566
Table 22-6. Endpoint kind meaning 567
Table 22-7. Transmission status encoding 567
Table 23-1. List of abbreviations used in register 570
Table 23-2. List of terms 570
Table 24-1. Revision history 572

1. System and memory architecture

The GD32F1x0 series are 32-bit general-purpose microcontrollers based on the ARM® Cortex®-M3 processor. The Cortex®-M3 processor includes three AHB buses known as I-Code, D-Code and System buses. All memory accesses of the Cortex®-M3 processor are executed on the three 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.

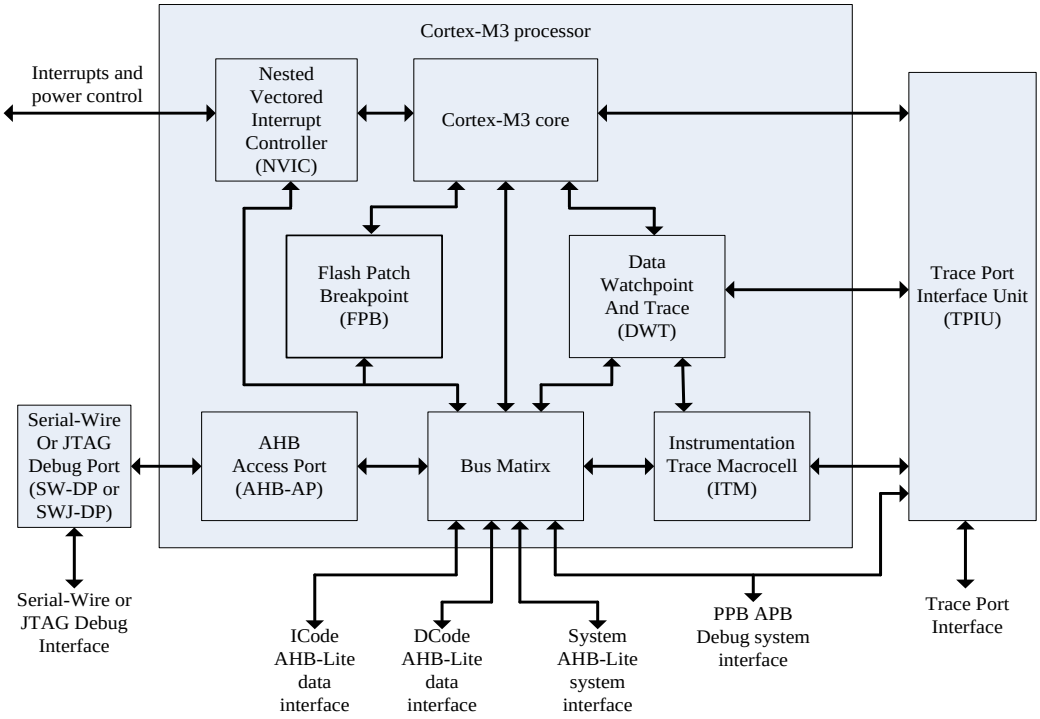
1.1. ARM Cortex-M3 processor

The Cortex®-M3 processor is a 32-bit processor including the features of low interrupt latency and low-cost debug. Integrated and advanced features make the Cortex®-M3 processor suitable for market products that require microcontrollers with high performance and low power consumption. The Cortex®-M3 processor is based on the ARMv7 architecture and supports a powerful and scalable instruction set including general data processing I/O control tasks and advanced data processing bit field manipulations. Some system peripherals listed below are also provided by Cortex®-M3.

- Internal Bus Matrix connected with I-Code bus, D-Code bus, System bus, Private Peripheral Bus (PPB) and debug accesses (AHB-AP).
- Nested Vectored Interrupt Controller (NVIC).
- Flash Patch and Breakpoint (FPB).
- Data Watchpoint and Trace (DWT).
- Instrumentation Trace Macrocell (ITM).
- Serial Wire JTAG Debug Port (SWJ-DP).
- Trace Port Interface Unit (TPIU).

The following figure shows the Cortex®-M3 processor block diagram. For more information, refer to the ARM® Cortex®-M3 Technical Reference Manual.

Figure 1-1. The structure of the Cortex®-M3 processor

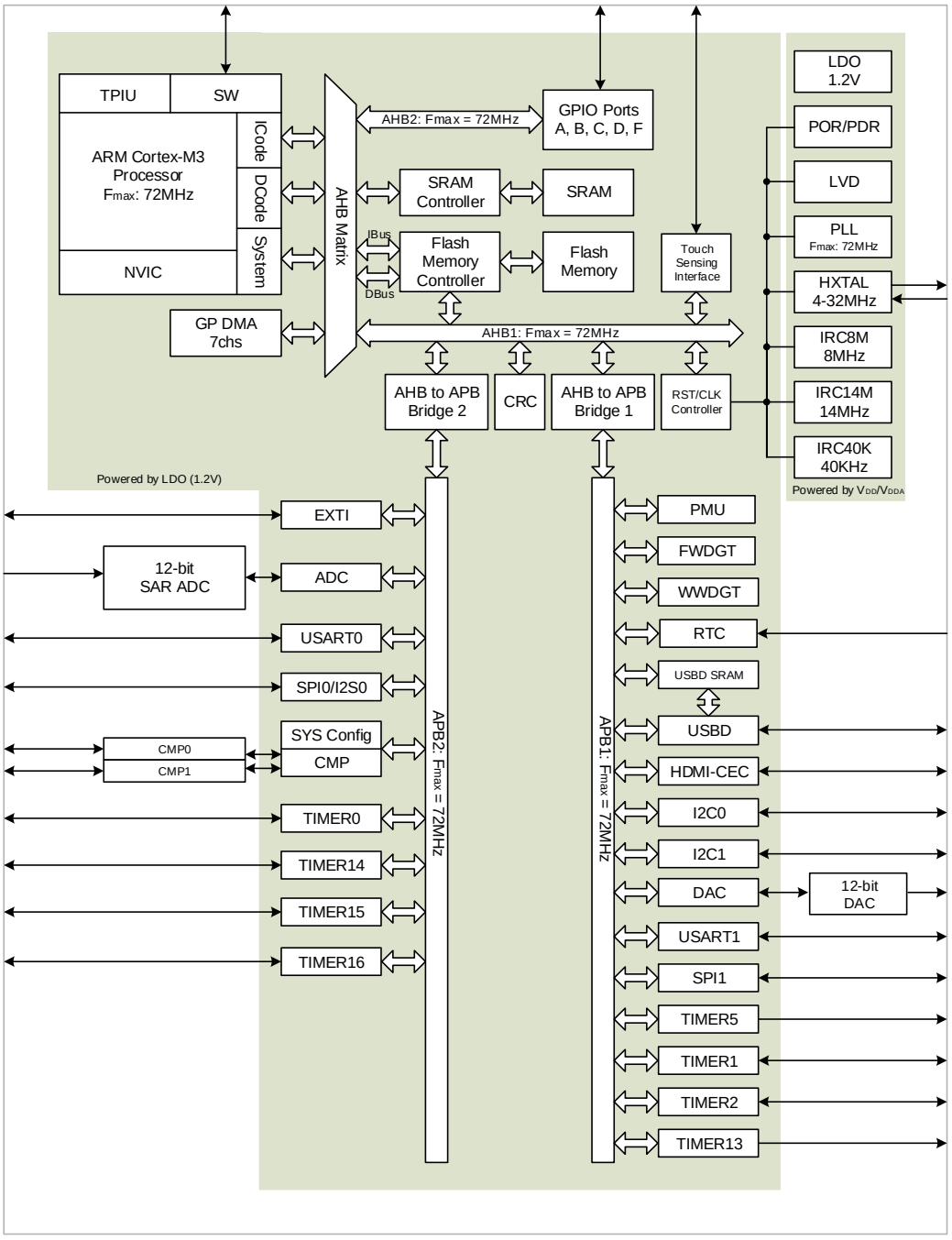


1.2. System architecture

The system architecture of GD32F1x0 series is shown in the following figure. The AHB matrix based on AMBA 3.0 AHB-LITE is a multi-layer AHB, which enables parallel access paths between multiple masters and slaves in the system. There are four masters on the AHB matrix, including I-Code, D-Code, system bus of the Cortex®-M3 core and DMA. The I-Code bus is the instruction bus and also used for vector fetches from the Code region (0x0000 0000 ~ 0x1FFF FFFF) to the Cortex®-M3 core. The D-Code bus is used for loading/storing data and also for debugging access of the Code region. Similarly, the System bus 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. The AHB matrix consists of five slaves, including I-Code and D-Code interfaces of the flash memory controller, internal SRAM, AHB1 and AHB2.

The AHB2 connects with the GPIO ports. The AHB1 connects with the AHB peripherals including two AHB-to-APB bridges which provide full synchronous connections between the AHB1 and the two APB buses. The two APB buses connect with all the APB peripherals.

Figure 1-2. Series system architecture of GD32F130xx and GD32F150xx devices



1.3. Memory map

The ARM® Cortex®-M3 processor is structured in Harvard architecture which can use separate buses to fetch instructions and load / store data. The instruction code and data are both located in the same memory address space but in different address ranges. 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®-M3 since it has a 32-bit bus address width. Additionally, a pre-defined memory map is provided by the Cortex®-M3 processor to reduce the software complexity of repeated implementation of different device

vendors. However, some regions are used by the ARM® Cortex®-M3 system peripherals. The following figure shows the memory map of GD32F1x0 series, including Code, SRAM, peripheral, and other pre-defined regions. Each peripheral of either type is allocated 1KB of space. This allows simplifying the address decoding for each peripheral.

Table 1-1. Memory map of GD32F130xx and GD32F150xx devices

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0xE000 0000 - 0xE00F FFFF	Cortex-M3 internal peripherals
External Device		0xA000 0000 - 0xDFFF FFFF	Reserved
External RAM		0x6000 0000 - 0x9FFF FFFF	Reserved
Peripherals	AHB1	0x5000 0000 - 0x5FFF FFFF	Reserved
	AHB2	0x4800 1800 - 0x4FFF FFFF	Reserved
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	Reserved
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4002 4400 - 0x47FF FFFF	Reserved
		0x4002 4000 - 0x4002 43FF	TSI
		0x4002 3400 - 0x4002 3FFF	Reserved
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	Reserved
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	Reserved
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0400 - 0x4002 0FFF	Reserved
		0x4002 0000 - 0x4002 03FF	DMA
	APB2	0x4001 4C00 - 0x4001 FFFF	Reserved
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 4000 - 0x4001 43FF	TIMER14
		0x4001 3C00 - 0x4001 3FFF	Reserved
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	Reserved
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	Reserved
		0x4001 2400 - 0x4001 27FF	ADC
		0x4001 0800 - 0x4001 23FF	Reserved
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG+CMP
	APB1	0x4000 C400 - 0x4000 FFFF	Reserved

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x4000 C000 - 0x4000 C3FF	Reserved
		0x4000 8000 - 0x4000 BFFF	Reserved
		0x4000 7C00 - 0x4000 7FFF	Reserved
		0x4000 7800 - 0x4000 7BFF	CEC
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6400 - 0x4000 6FFF	Reserved
		0x4000 6000 - 0x4000 63FF	USB SRAM
		0x4000 5C00 - 0x4000 5FFF	USB registers
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4800 - 0x4000 53FF	Reserved
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	Reserved
		0x4000 3C00 - 0x4000 3FFF	Reserved
		0x4000 3800 - 0x4000 3BFF	SPI1
		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	TIMER13
		0x4000 1400 - 0x4000 1FFF	Reserved
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0800 - 0x4000 0FFF	Reserved
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2000 2000 - 0x3FFF FFFF	Reserved
		0x2000 0000 - 0x2000 1FFF	SRAM
Code		0x1FFF F810 - 0x1FFF FFFF	Reserved
		0x1FFF F800 - 0x1FFF F80F	Option bytes
		0x1FFF EC00 - 0x1FFF F7FF	System memory
		0x0801 0000 - 0x1FFF EBFF	Reserved
		0x0800 0000 - 0x0800 FFFF	Main Flash memory
		0x0000 0000 - 0x07FF FFFF	Aliased to Flash or system memory

1.3.1. Bit-banding

In order to reduce the time of read-modify-write operations, the Cortex®-M3 processor provides a bit-banding function to perform a single atomic bit operation. The memory map

includes two bit-band regions. These occupy the SRAM and Peripherals respectively. These bit-band regions map each word in an alias region of memory to a bit in a bit-band region of memory.

A mapping formula shows how to reference each word in the alias region to a corresponding bit, or target bit, in the bit-band region. The mapping formula is:

bit_word_addr =bit_band_base +(byte_offset×32)+(bit_number×4) (1-1)

where:

- bit_word_addr is the address of the word in the alias memory region that maps to the targeted bit.
- bit_band_base is the starting address of the alias region.
- byte_offset is the number of the byte in the bit-band region that contains the targeted bit.
- bit_number is the bit position (0-7) of the targeted bit.

For example, to access bit 7 of address 0x2000 0200, the bit-band alias is:

bit_word_addr = 0x2200 0000 + (0x200 * 32)+ (7 * 4)= 0x2200 401C (1-2)

Writing to address 0x2200 401C will cause bit 7 of address 0x2000 0200 change while a read to address 0x2200 401C will return 0x01 or 0x00 according to the value of bit 7 at the SRAM address 0x2000 0200.

1.3.2. On-chip SRAM memory

The GD32F1x0 series contain up to 8KB of on-chip SRAM which starts at the address 0x2000 0000. It supports byte, half-word (16 bits), and word (32 bits) accesses. In order to increase memory robustness, parity check is supported. The user can enable the parity check function using the bit SRAM_PARITY_CHECK in the user option bytes (refer to Chapter 2.3.9 ***Option bytes description***). When enabled, an NMI is generated if the parity check fails. The SRAM parity check error flag is implemented in the system configuration register 2 (SYSCFG_CFG2). The error flag can be connected to the break input of TIMER 0 / TIMER 14 / TIMER 15 / TIMER 16, if the SRAM_PARITY_ERROR_LOCK control bit in the system configuration register 2 (SYSCFG_CFG2) is set to 1.

The real data width of the SRAM is 36 bits, including 32 bits for data and 4 bits for parity (1 bit per byte). When writing, the parity bits are computed and stored into the SRAM. When reading, the parity bits are also computed using the stored data in SRAM. The computed parity bits are compared with the stored parity bits which are computed during the writing access. If they are different, the parity check fails.

Note: When enabling the SRAM parity check, it is recommended to initialize the whole SRAM memory by software at the beginning of the code, in order to avoid getting parity check errors when reading non-initialized locations.

1.3.3. On-chip Flash memory

The devices provide up to 64 KB of on-chip flash memory. The flash memory consists of up to 64 KB main flash organized into 64 pages with 1 KB capacity per page and a 3 KB information block for the boot loader. The following table shows details.

Table 1-2. Flash module organization

Block	Name	Address	Size
Main Flash Block	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbytes
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbytes
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbytes
	.	.	.
	.	.	.
	Page 63	0x0800 FC00 - 0x0800 FFFF	1 Kbytes
Information Block	System memory	0x1FFF EC00 - 0x1FFF F7FF	3 Kbytes
Option Bytes Block	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16 bytes

Read accesses to the preceding 32 pages can be performed 32 bits per cycle without any wait state. All of byte, half-word (16 bits) and word (32 bits) read accesses are supported. The flash memory can be programmed half-word (16 bits) or word (32 bits) at a time. Each page of the flash memory can be erased individually. The whole flash memory space except information blocks can be erased at a time.

1.4. Boot configuration

The GD32F1x0 series provides three kinds of boot sources which can be selected using the bit BOOT1_n in the user option bytes (refer to Chapter 2.3.9 ***Option bytes description***) and the BOOT0 pins. The value on the BOOT0 pin is latched on the 4th rising edge of SYSCLK after a reset. It is up to the user to set the BOOT1_n and BOOT0 after a power-on reset or a system reset to select the required boot source. The details are shown in the following table.

Table 1-3. Boot modes

Selected boot source	Boot mode selection pins	
	BOOT1	BOOT0
Main Flash Memory	x	0
System Memory	0	1
On-chip SRAM	1	1

Note: The BOOT1 value is the opposite of the BOOT1_n value.

After power-on sequence or a system reset, the ARM® Cortex®-M3 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 EC00) is aliased in the boot memory space which begins at the address 0x0000 0000. When the on-chip SRAM whose memory space is beginning at 0x2000 0000 is selected as the boot source, in the application initialization code, you have to relocate the vector table in SRAM using the NVIC exception table and offset register.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory. The boot loader can be activated through one of the following serial interfaces: USART0 or USART1.

1.5. System configuration registers (SYSCFG)

SYSCFG base address: 0x4001 0000

1.5.1. System configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00

Reset value: 0x0000 000X (X indicates BOOT_MODE[1:0] may be any value according to the BOOT0 pin and the BOOT1_n option bit after reset)

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												PB9_HCCE	Reserved		
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TIMER16_DMA_RMP	TIMER15_DMA_RMP	USART0_RX_DMA_RMP	USART0_TX_DMA_RMP	ADC_DMA_RMP	Reserved						BOOT_MODE[1:0]	
			rw	rw	rw	rw	rw							r	

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	PB9_HCCE	PB9 pin high current capability enable When it is set, the PB9 pin can be used to control an infrared LED directly. 0: High current capability on the PB9 pin is disabled. 1: High current capability on the PB9 pin is enabled, and the speed control of the pin is bypassed.
18:13	Reserved	Must be kept at reset value.
12	TIMER16_DMA_RMP	TIMER 16 DMA request remapping enable 0: Not remap (TIMER16_CH0 and TIMER16_UP DMA requests are mapped on DMA channel 0) 1: Remap (TIMER16_CH0 and TIMER16_UP DMA requests are mapped on DMA channel 1)
11	TIMER15_DMA_RMP	TIMER 15 DMA request remapping enable 0: Not remap (TIMER15_CH0 and TIMER15_UP DMA requests are mapped on DMA channel 2) 1: Remap (TIMER15_CH0 and TIMER15_UP DMA requests are mapped on DMA channel 3)
10	USART0_RX_DMA_RMP	USART0_RX DMA request remapping enable 0: Not remap (USART0_RX DMA requests are mapped on DMA channel 2) 1: Remap (USART0_RX DMA requests are mapped on DMA channel 4)
9	USART0_TX_DMA_	USART0_TX DMA request remapping enable

	RMP	0: Not remap (USART0_TX DMA requests are mapped on DMA channel 1) 1: Remap (USART0_TX DMA requests are mapped on DMA channel 3)
8	ADC_DMA_RMP	ADC DMA request remapping enable 0: Not remap (ADC DMA requests are mapped on DMA channel 0) 1: Remap (ADC DMA requests are mapped on DMA channel 1)
7:2	Reserved	Must be kept at reset value
1:0	BOOT_MODE[1:0]	Boot mode (Refer to Chapter 1.4 ***Boot configuration*** for details) Bit0 is mapping to the BOOT0 pin; the value of bit1 is the opposite of the BOOT1_n option bit value. x0: Boot from the Main Flash 01: Boot from the system memory 11: Boot from the embedded SRAM

1.5.2. EXTI sources selection register 0 (SYSCFG_EXTISS0)

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
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: Reserved X100: Reserved X101: Reserved X110: Reserved X111: Reserved
11:8	EXTI2_SS[3:0]	EXTI 2 sources selection X000: PA2 pin X001: PB2 pin X010: PC2 pin

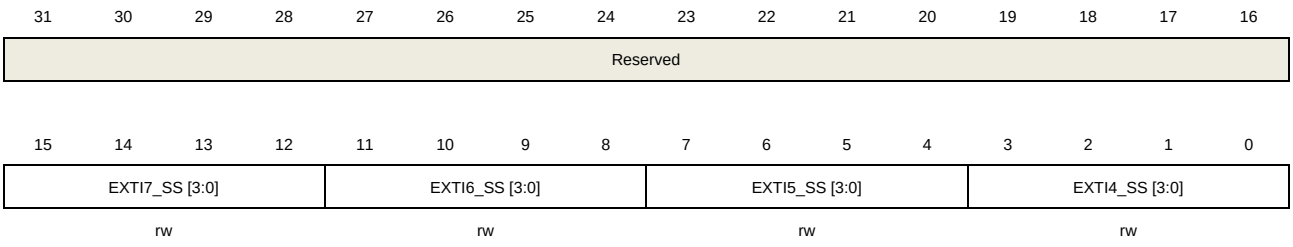
		X011: PD2 pin X100: Reserved X101: Reserved X110: Reserved X111: Reserved
7:4	EXTI1_SS[3:0]	EXTI 1 sources selection X000: PA1 pin X001: PB1 pin X010: PC1 pin X011: Reserved X100: Reserved X101: PF1 pin X110: Reserved X111: Reserved
3:0	EXTI0_SS[3:0]	EXTI 0 sources selection X000: PA0 pin X001: PB0 pin X010: PC0 pin X011: Reserved X100: Reserved X101: PF0 pin X110: Reserved X111: Reserved

1.5.3. EXTI sources selection register 1 (SYSCFG_EXTISS1)

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.
15:12	EXTI7_SS[3:0]	EXTI 7 sources selection X000: PA7 pin X001: PB7 pin X010: PC7 pin

		X011: Reserved X100: Reserved 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: Reserved X100: Reserved 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: Reserved X100: Reserved 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: Reserved X100: Reserved X101: PF4 pin X110: Reserved X111: Reserved

1.5.4. EXTI sources selection register 2 (SYSCFG_EXTISS2)

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

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: Reserved X100: Reserved 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: Reserved X100: Reserved 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: Reserved X100: Reserved 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: Reserved X100: Reserved X101: Reserved X110: Reserved X111: Reserved

1.5.5. EXTI sources selection register 3 (SYSCFG_EXTISS3)

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
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			
rw				rw				rw				rw			

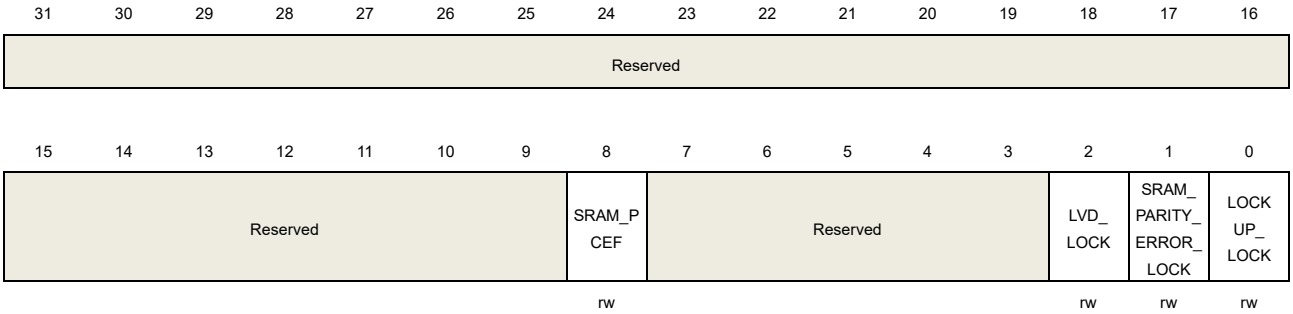
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI15_SS[3:0]	EXTI 15 sources selection X000: PA15 pin X001: PB15 pin X010: PC15 pin X011: Reserved X100: Reserved X101: Reserved X110: Reserved X111: Reserved
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection X000: PA14 pin X001: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

1.5.6. System configuration register 2 (SYSCFG_CFG2)

Address offset: 0x18
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	SRAM_PCEF	SRAM parity check error flag This bit is set by hardware when an SRAM parity check error occurs. It is cleared by software by writing 1. 0: No SRAM parity check error detected 1: SRAM parity check error detected
7:3	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 TIMER0/14/15/16. LVDEN and LVDT[2:0] in the PMU_CTL register can be programmed. 1: The LVD interrupt is connected from the break input of TIMER0/14/15/16. LVDEN and LVDT[2:0] in the PMU_CTL register are read only.
1	SRAM_PARITY_ERROR_LOCK	SRAM parity check error lock This bit is set by software and cleared by a system reset. 0: The SRAM parity check error is disconnected from the break input of TIMER0/14/15/16

		1: The SRAM parity check error is connected from the break input of TIMER0/14/15/16
0	LOCKUP_LOCK	Cortex®-M3 LOCKUP output lock This bit is set by software and cleared by a system reset. 0: The Cortex®-M3 LOCKUP output is disconnected from the break input of TIMER0 / 14 / 15 / 16 1: The Cortex®-M3 LOCKUP output is connected from the break input of TIMER0 / 14 / 15 / 16

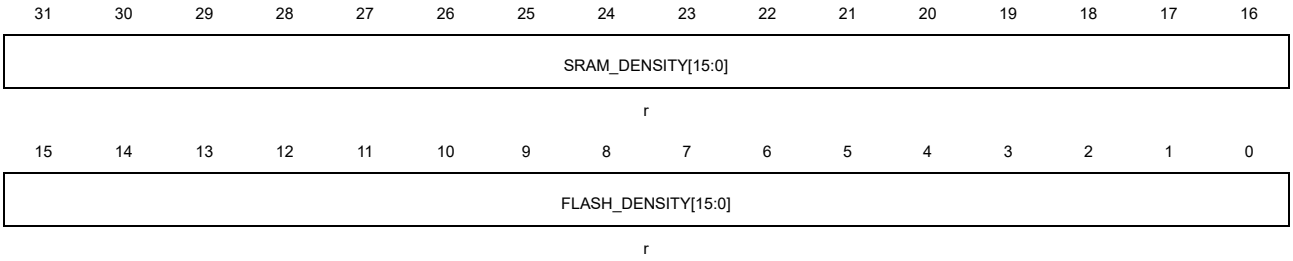
1.6. Device electronic signature

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.

1.6.1. 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).



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

1.6.2. Unique device ID (96 bits)

Base address: 0x1FFF F7AC
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 F7B0

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 F7B4

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[95:80]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[79:64]															
r															

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. There is no waiting time within 32K bytes while CPU executes instruction. It also provides page erase, mass erase, and word/half word program for flash memory.

2.2. Characteristics

- Up to 64KB of on-chip flash memory for storing instruction/data.
- No waiting time within 32K bytes when CPU executes instruction.
- A long delay when fetch 32K ~ 64K bytes date from flash.
- 3K bytes information block for boot loader.
- 16 bytes option bytes block for user application requirements.
- 1K bytes page size.
- Word or half word programming, page erase and mass erase capability.
- Flash read protection to prevent illegal code/data access.
- Page erase/program protection to prevent unexpected operation.

2.3. Function overview

2.3.1. Flash memory architecture

The flash memory consists of up to 64 KB main flash organized into 64 pages with 1 KB capacity per page and a 3 KB Information Block for the Boot Loader. The main flash memory contains a total of up to 64 pages which can be erased individually. The following table shows the base address and size.

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

Block	Name	Address	size(bytes)
Main Flash Block	Page 0	0x0800 0000 - 0x0800 03FF	1KB
	Page 1	0x0800 0400 - 0x0800 07FF	1KB
	Page 2	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	.	.	.
	Page 63	0x0800 FC00 - 0x0800 FFFF	1KB
Information Block	Boot loader	0x1FFF EC00 - 0x1FFF F7FF	3KB
Option Bytes Block	Option bytes	0x1FFF F800 - 0x1FFF F80F	16B

Note: The Information Block stores the bootloader - this block cannot be programmed or

erased by user.

2.3.2. 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 IBUS or DBUS from the CPU.

2.3.3. Unlock the FMC_CTL register

After reset, the FMC_CTL register is not accessible in write mode, except for the OBRLD bit, which is used for reloading the option bytes, and the LK bit in FMC_CTL register is 1. An unlocking sequence consists of two write operations to the FMC_KEY register can open the access to the FMC_CTL register. The two write operations are writing 0x45670123 and 0xCDEF89AB to the FMC_KEY register. After the two write operations, the LK bit in FMC_CTL register is set to 0 by hardware. The software can lock the FMC_CTL again by setting the LK bit in FMC_CTL register to 1. If there is any wrong operations on the FMC_KEY register, the LK bit in FMC_CTL register will be set, and the FMC_CTL register will be locked, then it will generate a bus error.

The OBPG bit and OBER bit in FMC_CTL are also protected by FMC_OBKEY register. The unlocking sequence includes two write operations, which are writing 0x45670123 and 0xCDEF89AB to FMC_OBKEY register. And then set the OBWEN bit in FMC_CTL register to 1. The software can set OBWEN bit to 0 to protect the OBPG bit and OBER bit in FMC_CTL register again.

2.3.4. Page erase

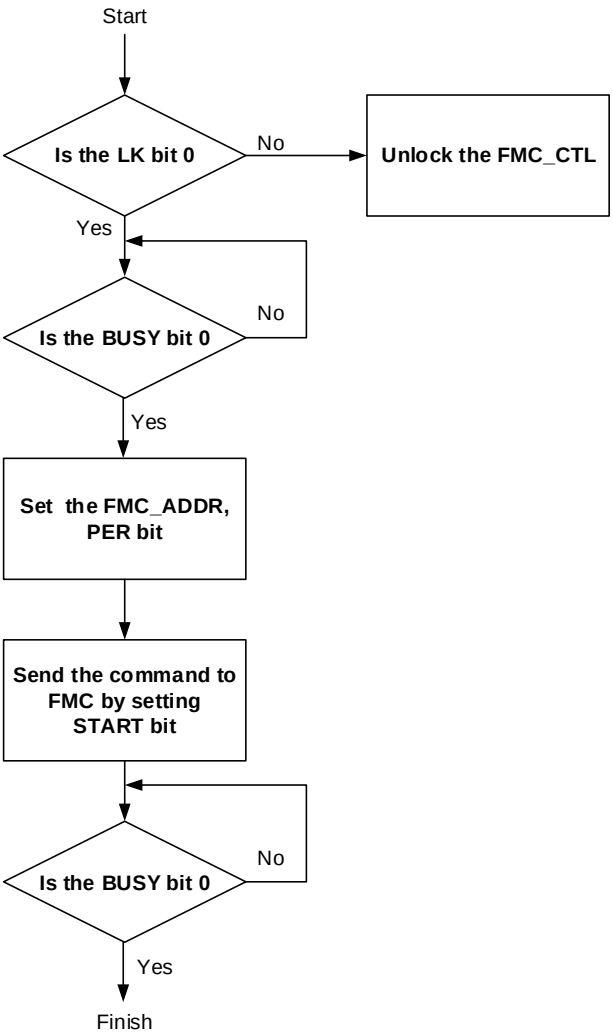
The FMC provides a page erase function which is used for initializing the contents of a main flash memory page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the register for a page erase operation.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT 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_ADDR register.
- Write the page erase command by setting PER bit in FMC_CTL register.
- Send the page erase command to the FMC by setting the START bit in FMC_CTL register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the flash memory if required by using a DBUS access.

When the operation is executed successfully, an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set, and the ENDF in FMC_STAT register is set. If the target erasing page is being used for fetching codes or accessing data, the software may run

out of control. The FMC will not provide any notification when this occurs. Additionally, the page erase operation will be ignored on protected pages. A Flash Operation Error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the PGERR bit in the FMC_STAT register to detect this condition in the interrupt handler. The end of this operation is indicated by the ENDF bit in the FMC_STAT register. The ***Figure 2-1. Process of page erase operation*** shows the page erase operation flow.

Figure 2-1. Process of page erase operation



2.3.5. 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.

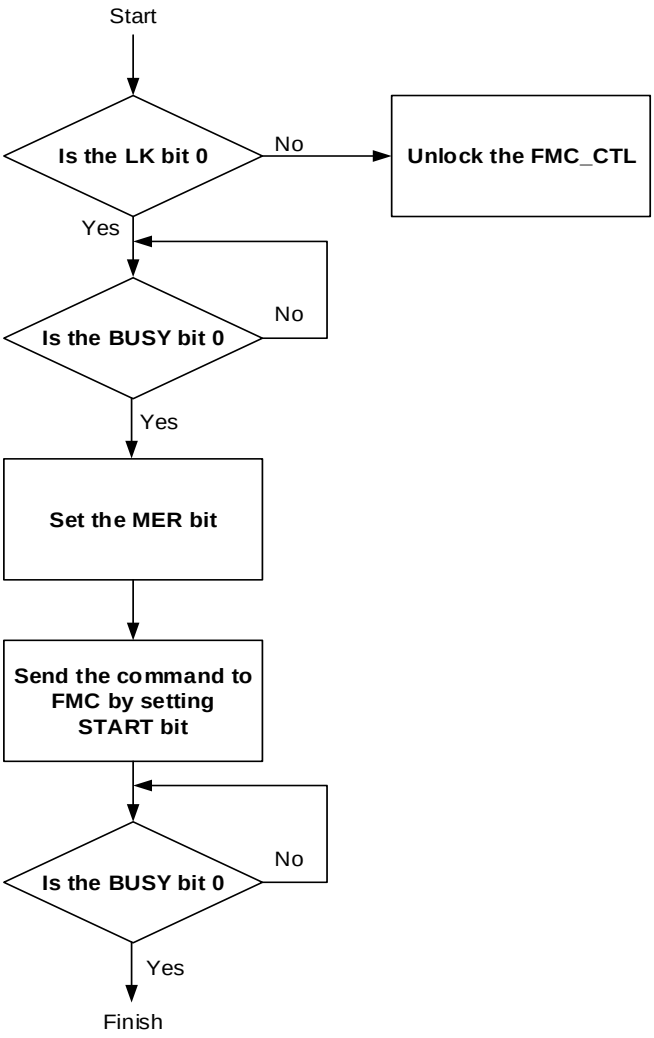
- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Write the mass erase command by setting MER bit in FMC_CTL register.
- Send the mass erase command to the FMC by setting the START bit in FMC_CTL

register.

- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT 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_CTL register is set, and the ENDF in FMC_STAT 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. The end of this operation is indicated by the ENDF bit in the FMC_STAT register. (The starting address of programming operation should be 0x0800 0000) The ***Figure 2-2. Process of the mass erase operation*** indicates the mass erase operation flow.

Figure 2-2. Process of the mass erase operation



2.3.6. Main flash programming

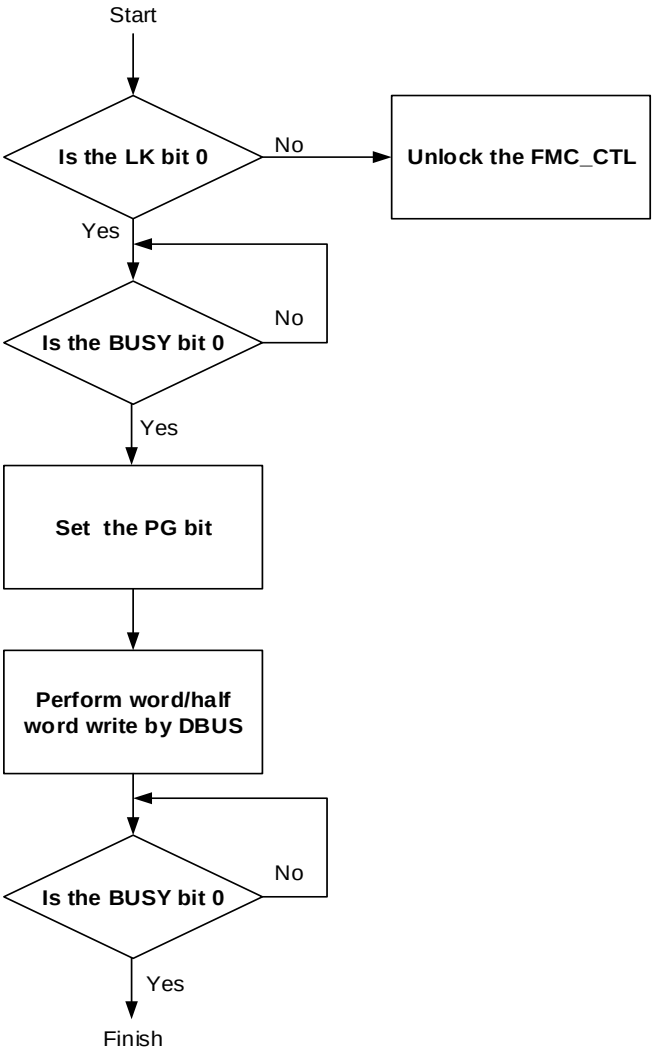
The FMC provides a 32-bit word/16-bit half word programming function which is used to

modify the main flash memory contents. The following steps show the word programming operation register access sequence.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT 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_CTL register.
- A 32-bit word/16-bit half word write at desired address by DBUS.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT 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_CTL register is set, and the ENDF in FMC_STAT register is set. Note that before the word/half 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 except programming 0x0. Additionally, the program operation will be ignored on protected pages. A flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the PGERR bit in the FMC_STAT register to detect this condition in the interrupt handler. The end of this operation is indicated by the ENDF bit in the FMC_STAT register. The following figure displays the word programming operation flow.

Figure 2-3. Process of the word programming operation



2.3.7. Option bytes erase

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

- Unlock the FMC_CTL register if necessary.
- Unlock the OBWEN bit in FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT 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 option bytes erase command into OBER bit in FMC_CTL register.
- Send the option bytes erase command to the FMC by setting the START bit in FMC_CTL register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT 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_CTL register is set, and the ENDF in FMC_STAT register is set. The end of this operation is indicated by the ENDF bit in the FMC_STAT register.

2.3.8. Option bytes programming

The FMC provides a 32-bit word/16-bit half word programming function which is used for modifying the option bytes block contents. The following steps show the programming operation sequence.

- Unlock the FMC_CTL register if necessary.
- Unlock the OBWEN bit in FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT 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 program command into the OBPG bit in FMC_CTL register.
- A 32-bit word/16-bit half word write at desired address by DBUS.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT 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_CTL register is set, and the ENDF in FMC_STAT register is set. Note that before the word/half 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 except programming 0x0. The end of this operation is indicated by the ENDF bit in the FMC_STAT register.

2.3.9. Option bytes description

The option bytes block of flash memory reloaded to FMC_OBSTAT and FMC_WP registers after each system reset or OBRLD bit set in FMC_CTL register, and the option bytes work. The option complement bytes are the opposite of option bytes. When option bytes reload, if the option complement bytes and option bytes does not match, the OBERR bit in FMC_OBSTAT register is set, and the option bytes is set to 0xFF. The following table is the detail of option bytes.

Table 2-2. Option bytes

Address	Name	Description
0x1fff f800	OB_SPC	option bytes Security Protection Code 0xA5: No 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	option bytes which user defined [7]: Reserved [6]: SRAM_PARITY_CHECK 0: Enable sram parity check

Address	Name	Description
		1: Disable sram parity check [5]: VDDA_VISOR 0: Disable V_{DDA} monitor 1: Enable V_{DDA} monitor [4]: BOOT1_n 0: BOOT1 bit is 1 1: BOOT1 bit is 0 [3]: Reserved [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_SW 0: Hardware free watchdog timer 1: Software free watchdog timer
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_WP[7:0]	Page Erase/Program Protection bit 7 to 0
0x1fff f809	OB_WP_N[7:0]	OB_WP complement value bit 7 to 0
0x1fff f80a	OB_WP[15:8]	Page Erase/Program Protection bit 15 to 8
0x1fff f80b	OB_WP_N[15:8]	OB_WP complement value bit 15 to 8

2.3.10. Page 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_STAT register will then be set by the FMC. If the WPERR bit is set and the ERRIE bit is also set to 1 to enable the corresponding interrupt, then the flash operation error interrupt will be triggered by the FMC to get the attention of the CPU. The page protection function can be individually enabled by configuring the OB_WP [15:0] bit field to 0 in the option bytes. If a page erase operation is executed on the Option Bytes region, all the flash memory page protection functions will be disabled. When setting or resetting OB_WP in the option bytes, the software need to set OBRLD in FMC_CTL register or a system reset to reload the OB_WP bits. The following table shows which pages are protected by set OB_WP [15:0].

Table 2-3. OB_WP bit for pages protected

OB_WP bit	pages protected
OB_WP[0]	page 0 ~ page 3
OB_WP[1]	page 4 ~ page 7
OB_WP[2]	page 8 ~ page 11
. . .	. . .
OB_WP[14]	page 56 ~ page 59
OB_WP[15]	page 60 ~ page 63

2.3.11. 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 protecting:

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

Protection level low: when setting OB_SPC byte and its complement value to any value except 0x0x5AA5 or 0x33CC, protection level low performed. The main flash can only be accessed by user code. In debug mode, boot from SRAM or boot from boot loader mode, all operations to main flash is forbidden. If a read operation is executed to main flash in debug mode, boot from SRAM or boot from boot loader mode, a bus error will be generated. If a program/erase operation is executed to main flash in debug mode, boot from SRAM or boot from boot loader mode, the PGERR bit in FMC_STAT register will be set. At protection level low, option bytes block are accessible by all operations. If program back to no protection level by setting OB_SPC byte and its complement value to 0x0x5AA5, a mass erase for main flash will be performed.

Protection level high: when set OB_SPC byte and its complement value to 0x33CC, protection level high performed. When this level is programmed in debug mode, boot from SRAM or boot from boot loader mode is disabled. The main flash block is accessible by all operations from user code. The option bytes cannot be erased, and the OB_SPC byte and its complement value cannot be reprogrammed. So, if protection level high is programmed, it cannot move back to protection level low or no protection level.

2.4. Register definition

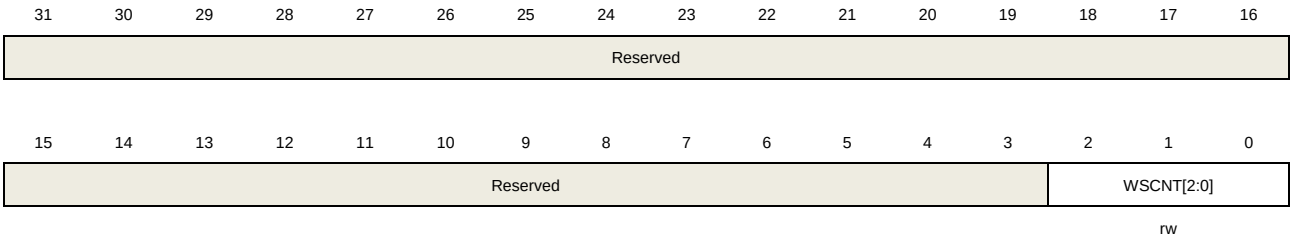
FMC base address: 0x4002 2000

2.4.1. Wait state register (FMC_WS)

Address offset: 0x00

Reset value: 0x0000 0000

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



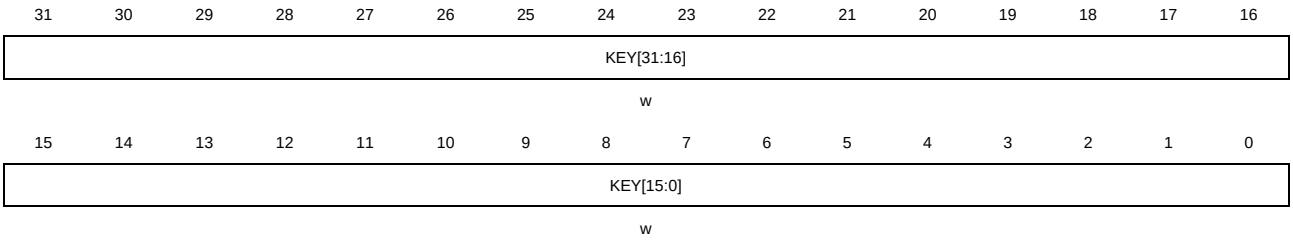
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	WSCNT[2:0]	Wait state counter register These bits set and reset by software. The WSCNT valid when WSEN bit is set 000: 0 wait state added 001: 1 wait state added 010: 2 wait state added 011 ~ 111: Reserved

2.4.2. Unlock key register (FMC_KEY)

Address offset: 0x04

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:0	KEY[31:0]	FMC_CTL unlock registers These bits are only be written by software

Write KEY[31:0] with key to unlock FMC_CTL register.

2.4.3. Option bytes unlock key register (FMC_OBKEY)

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
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_CTL option bytes operation unlock registers These bits are only be written by software Write OBKEY[31:0] with key to unlock option bytes command in FMC_CTL register.

2.4.4. Status register (FMC_STAT)

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										ENDF	WPERR	Reserved	PGERR	Reserved	BUSY
										rc_w1	rc_w1		rc_w1		r

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
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 erasing/programming on protected pages, this bit is set by hardware. The software can clear it by writing 1.

3	Reserved	Must be kept at reset value.
2	PGERR	Program error flag bit When programming to the flash while it is not 0xFFFF, this bit is set by hardware. The software can clear it by writing 1.
1	Reserved	Must be kept at reset value.
0	BUSY	The flash busy bit When the operation is in progress, this bit is set to 1. When the operation is end or an error generated, this bit is clear to 0.

2.4.5. Control register (FMC_CTL)

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	OBRLD	Option bytes reload bit This bit is set by software. 0: No effect 1: Force option bytes reload, and generate a system reset
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 bytes erase/program enable bit This bit is set by hardware when right sequence written to FMC_OBKEY register.

		This bit can be cleared by software.
8	Reserved	Must be kept at reset value.
7	LK	FMC_CTL lock bit This bit is cleared by hardware when right sequent written to FMC_KEY 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	OBER	Option bytes erase command bit This bit is set or cleared by software. 0: No effect 1: Option bytes erase command
4	OBPG	Option bytes program command bit This bit is set or cleared by software. 0: No effect 1: Option bytes program command
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 cleared by software. 0: No effect 1: Main flash page erase command
0	PG	Main flash page program command bit This bit is set or cleared by software. 0: No effect 1: Main flash page program command

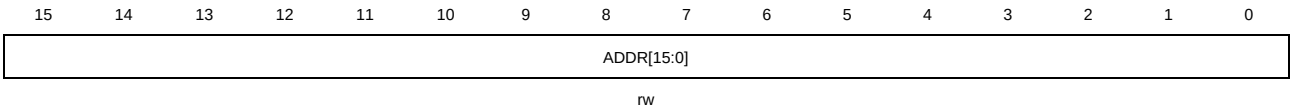
2.4.6. Address register (FMC_ADDR)

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]															
rw															

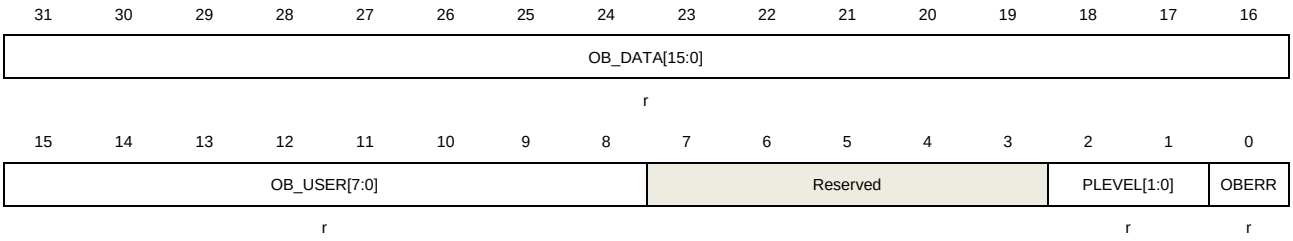


Bits	Fields	Descriptions
31:0	ADDR[31:0]	Flash command address bits These bits are set by software. ADDR bits are the address of flash erase command

2.4.7. Option bytes status register (FMC_OBSTAT)

Address offset: 0x1C
Reset value: 0xXXXX XX0X

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

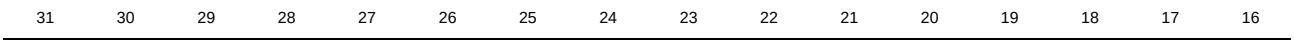


Bits	Fields	Descriptions
31:16	OB_DATA[15:0]	Store OB_DATA[15:0] of option bytes block after system reset
15:8	OB_USER[7:0]	Store OB_USER byte of option bytes block after system reset
7:3	Reserved	Must be kept at reset value.
2:1	PLEVEL[1:0]	Security Protection level 00: No protection level 01: Protect level low 11: Protect level high
0	OBERR	Option bytes read error bit. This bit is set by hardware when the option bytes and its complement bytes do not match, and the option bytes set 0xFF.

2.4.8. Write protection register (FMC_WP)

Address offset: 0x20
Reset value: 0x0000 XXXX

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
OB_WP[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	OB_WP[15:0]	Store OB_WP[15:0] of option bytes block after system reset 0: Protection active 1: Unprotected

2.4.9. Wait state enable register (FMC_WSEN)

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															WSEN
															rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	WSEN	FMC wait state enable register This bit set and reset by software. This bit is also protected by the FMC_KEY register. The software need writing 0x45670123 and 0xCDEF89AB to the FMC_KEY register. 0: No wait state added when fetching flash 1: Wait state added when fetching flash

2.4.10. 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	Fields	Descriptions
31:0	PID[31:0]	Product reserved ID code register These bits are read only by software. These bits are unchanged constantly after power on. These bits are one time programmed when the chip product.

3. Power management unit (PMU)

3.1. Overview

The power consumption is regarded as one of the most important issues for the devices of GD32F1x0 series. The 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 the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. As is shown in ***Figure 3-1. Power supply overview***, there are three power domains, including V_{DD} / V_{DDA} domain, V_{CORE} domain, and Backup domain, as is shown in the following figure. 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 V_{CORE} domain power. A power switch is implemented for the Backup domain. It can be powered from the VBAT voltage when the main VDD supply is shut down.

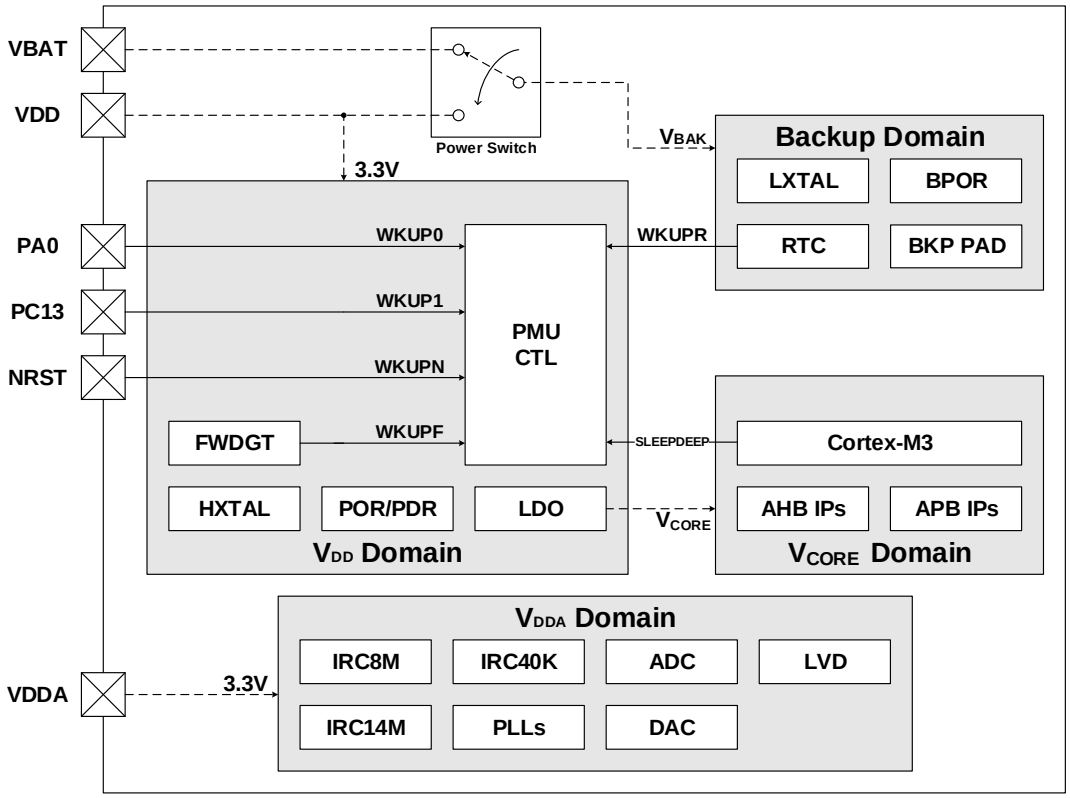
3.2. Characteristics

- Three power domains: V_{BAK}, V_{DD}/V_{DDA} and V_{CORE} power domains
- Three power saving modes: Sleep, Deep-sleep and Standby modes
- Internal Voltage regulator(LDO) supplies V_{CORE} voltage source
- Low Voltage Detector(LVD) can issue an interrupt or event when the power is lower than a programmed threshold
- Battery power (VBAT) for Backup domain when VDD is shut down

3.3. Function overview

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

Figure 3-1. Power supply overview



LVD: Low Voltage Detector
LDO: Voltage Regulator
BPOR: V_{BAK} Power On Reset
POR: Power On Reset
PDR: Power Down Reset

3.3.1. Backup domain

The Backup domain is powered by the VDD or the battery power source (VBAT) selected by the internal power switch, and the V_{BAK} pin which drives Backup Domain, power supply for RTC unit, LXTAL oscillator, BPOR, and three pads, including PC13 to PC15. In order to keeping the content of the Backup domain registers and the RTC supply, when VDD supply is shut down, VBAT pin can be connected to an optional standby voltage supplied by a battery or by another source. The power switch is controlled by the Power Down Reset circuit in the V_{DD} / V_{DDA} domain. If no external battery is used in the application, it is recommended to connect VBAT pin externally to VDD pin with a 100nF external ceramic decoupling capacitor.

The Backup Domain reset sources include the Backup domain power-on-reset (BPOR) and the Backup Domain software reset. The BPOR signal forces the device to stay in the reset mode until V_{BAK} is completely powered up. Also the application software can trigger the Backup domain software reset by setting the BKPRST bit in the RCU_BDCTL register to reset the Backup domain.

The clock source of Real Time Clock (RTC) circuit can be derived from the Internal 40 KHz RC oscillator (IRC40K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL clock divided by 32. When VDD is shut down, only LXTAL is valid for RTC. Before entering the power saving mode by executing the WFI/WFE instruction, the Cortex®-M3 can setup the RTC register with an expected alarm time and enable the alarm functionn and according EXTI lines to achieve the RTC timer wakeup event. After entering the power saving mode for a certain amount of time, the RTC alarm 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 by VDD (V_{BAK} pin is connected to VDD), the following functions are available:

- PC13 can be used as GPIO or RTC function pin described in the ***Real-time clock(RTC)***.
- PC14 and PC15 can be used as either GPIO or LXTAL Crystal oscillator pins.

When the Backup domain is supplied by VBAT (V_{BAK} pin is connected to VBAT), the following functions are available:

- PC13 can be used as RTC function pin described in the ***Real-time clock(RTC)***.
- PC14 and PC15 can be used as LXTAL Crystal oscillator pins only.

Note: Since PC13, PC14 and PC15 are supplied through the Power Switch, which can only be obtained by a small current, the speed of GPIOs PC13 to PC15 should not exceed 2MHz when they are in output mode(maximum load: 30pF).

3.3.2. 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 HXTAL (High Speed Crystal oscillator), LDO (Voltage Regulator), POR / PDR (Power On / Down Reset), 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), IRC14M (Internal 14MHz RC oscillator at 14MHz frequency), IRC40K (Internal 40KHz RC oscillator), PLLs (Phase Locking Loop), LVD (Low Voltage Detector), etc.

V_{DD} domain

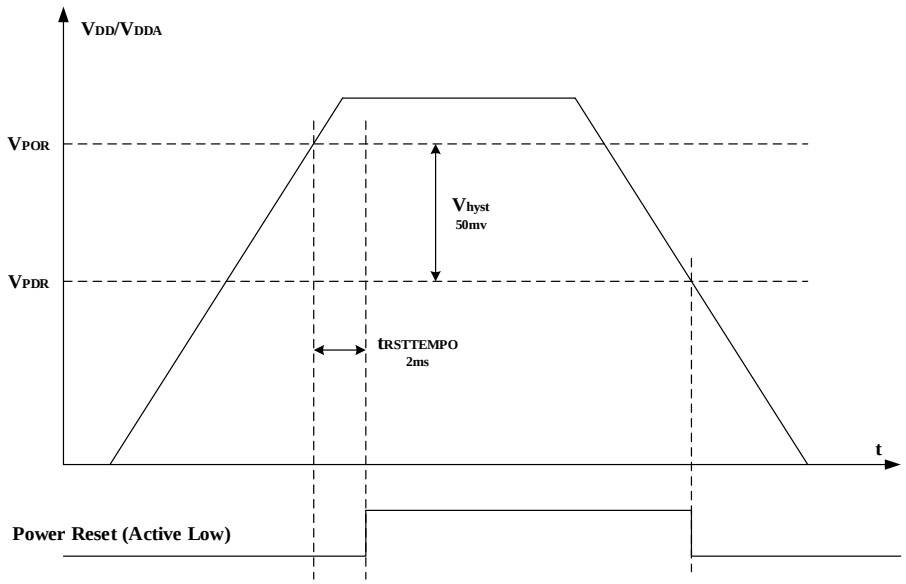
The LDO, which is implemented to supply power for the V_{CORE} domain, is always enabled after reset. It can be configured to operate in three different status, including in the Sleep mode (full power on), in the Deep-sleep mode (on or low power), and in the Standby mode (power off).

The POR/PDR circuit is implemented to detect V_{DD}/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. The POR circuit only detects V_{DD} supply voltage, and the PDR circuit detects both V_{DD} and V_{DDA} supply voltages.

Note: V_{DDA} monitor function detected by the PDR circuit can be disabled by reset VDDA_VISOR bit in option byte register OB_USER, to reduce power consumption when users are sure that the V_{DDA} is higher than or equal to V_{DD}.

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. V_{PDR} is configurable. There are two V_{PDR} values which can be selected by the PDVSEL bit in the RCU_PDVSEL registers (Refer to ***Register definition***). When the lower one is selected, it is strongly recommended that neither programming nor erasing is performed to the flash memory since it may fail when the voltage is low enough nearly the threshold. The hysteresis voltage (V_{hyst}) is around 50mV.

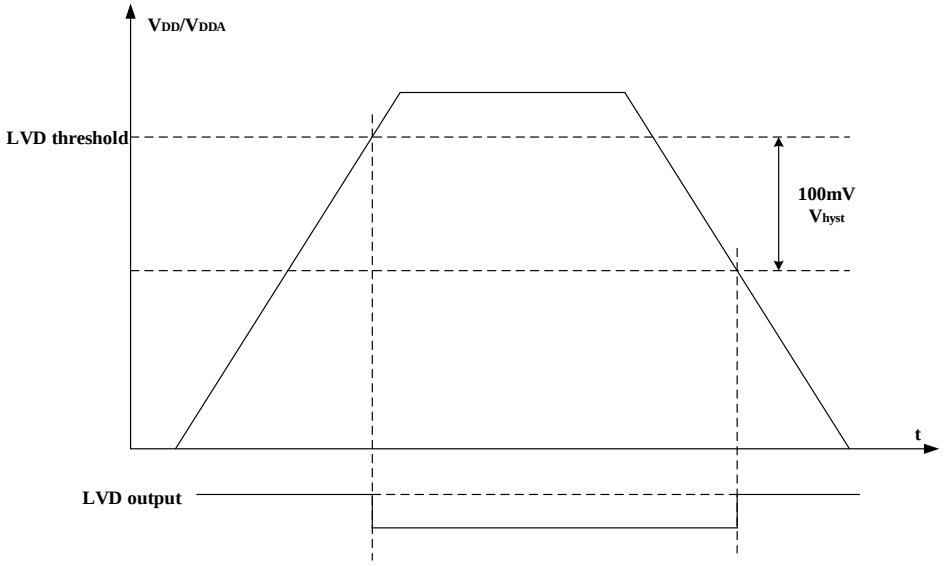
Figure 3-2. Waveform of the POR/PDR



V_{DDA} domain

The LVD is used to detect whether the V_{DD}/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, and LVDF bit, which in the Power status register (PMU_CS), indicates if V_{DD}/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 3-3. Waveform of 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 shows the relationship between the supply voltage and the LVD signal. The hysteresis voltage (V_{hyst}) is 100mV.

Figure 3-3. Waveform of LVD threshold



Generally, digital circuits are powered by V_{DD}, while most of analog circuits are powered by V_{DDA}. To improve the ADC and DAC conversion accuracy, 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 that avoids noise on V_{DDA}, and V_{SSA} should be connected to V_{SS} through the specific circuit independently. Otherwise, If the V_{DD} and V_{DDA} are not provided by the same power supply (the difference is less than 0.3V), the V_{DDA} must be greater than or equal to V_{DD} during the power-on process.

3.3.3. V_{CORE} power domain

The main functions that include Cortex®-M3 logic, AHB / APB peripherals, the APB interfaces for the Backup domain and the V_{DD}/V_{DDA} domain, etc., are located in the V_{CORE} power domain. Once the V_{CORE} is powered up, the POR will generate a reset sequence on the V_{CORE} power domain. To enter the expected 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 an expected power saving mode which will be discussed in the following section. Refer to the datasheet for the V_{CORE} voltage value.

3.3.4. Power saving modes

After a system reset or a power reset, the GD32F1x0 MCU operates at full function and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1, PCLK2) or gating 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®-M3. In Sleep mode, only clock of Cortex®-M3 is off. To enter the Sleep mode, it is only necessary to clear the SLEEPDEEP bit in the Cortex®-M3 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®-M3 Technical Reference Manual). The mode offers the lowest wakeup time as no time is wasted in interrupt entry or exit.

According to the SLEEPONEXIT bit in the Cortex®-M3 System Control Register, there are two options to select the Sleep mode entry mechanism.

- Sleep-now: if the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode as soon as 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 lowest priority ISR.

Deep-sleep Mode

The Deep-sleep mode is based on the SLEEPDEEP mode of the Cortex®-M3. In Deep-sleep mode, all clocks in the V_{CORE} domain for GD32F130/150xx devices are off, and all of IRC8M, IRC14M for GD32F130/150xx devices, HXTAL and PLLs are disabled. The contents of SRAM and registers are preserved. The LDO can operate normally 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®-M3 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®-M3 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.

Note: In order to enter Deep-sleep mode smoothly, all EXTI line pending status (in the EXTI_PD register) and and related peripheral flags must be reset, refer to ***Table 5-3. EXTI***

source If not, the program will skip the entry process of Deep-sleep mode to continue to execute the following procedure.

Standby Mode

The Standby mode is based on the SLEEPDEEP mode of the Cortex®-M3, too. In Standby mode, the whole V_{CORE} domain for GD32F130/150xx devices, the LDO is shut down, and all of IRC8M, IRC14M for GD32F130/150xx devices, HXTAL and PLLs are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M3 System Control Register, and set the STBMOD bit in the PMU_CTL register, and clear the WUF bit in the PMU_CS register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed. There are five wakeup sources for the Standby mode, including the external reset from NRST pin, the RTC alarm/time stamp/tamper events, the FWDGT reset, and the rising edge on WKUP0 or WKUP1 pin. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in V_{CORE} power domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset of V_{CORE} domain occurs and the Cortex®-M3 will execute instruction code from the 0x0000 0000 address.

Table 3-1. Power saving mode summary

Mode	Sleep	Deep-sleep	Standby
Description	Only CPU clock is off	1. All clocks in the V_{CORE} domain for GD32F130/150xx devices are off 2. Disable IRC8M, IRC14M for GD32F130/150xx devices, HXTAL and PLL	1. V_{CORE} domain for GD32F130/150xx devices is power off 2. Disable IRC8M, IRC14M for GD32F130/150xx devices, HXTAL and PLL
LDO Status	On (normal power mode)	On (normal or low power 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. RTC 3. FWDGT reset 4. WKUP0 pin 5. WKUP1 pin
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 0000 (reset by 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							BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
							rw	rw			rw	rc_w1	rc_w1	rw	rw

Bits	Fields	Descriptions
31:9	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: 2.2V 001: 2.3V 010: 2.4V 011: 2.5V 100: 2.6V 101: 2.7V 110: 2.8V 111: 2.9V
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®-M3 enters SLEEPDEEP mode 1: Enter the Standby mode when the Cortex®-M3 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. Power control/status register (PMU_CS)

Address offset: 0x04

Reset value: 0x0000 0000 (not reset by 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					LVDF	STBF	WUF
						rw	rw						r	r	r

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	WUPEN1	WKUP1 Pin (PC13) Enable 0: Disable WKUP1 pin function 1: Enable WKUP1 pin function If WUPEN1 is set before entering the Standby mode, a rising edge on the WKUP1 pin wakes up the system from the Standby mode. As the WKUP1 pin is active high, the WKUP1 pin is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
8	WUPEN0	WKUP0 Pin (PA0) Enable 0: Disable WKUP0 pin function 1: Enable WKUP0 pin function If WUPEN0 is set before entering the Standby mode, a rising edge on the WKUP0 pin

		wakes up the system from the Standby mode. As the WKUP0 pin is active high, the WKUP0 pin is internally configured to input pull down mode. And set this bit will trigger a wakeup event when the input is already high.
7:3	Reserved	Must be kept at reset value.
2	LVDF	Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (V_{DD} is higher than the specified LVD threshold) 1: Low Voltage event occurred (V_{DD} is equal to or lower than the specified LVD threshold) **Note:** The LVD function is stopped in Standby mode.
1	STBF	Standby Flag 0: The device has not entered the Standby mode 1: The device has been in 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 wakeup event including RTC Tamper event, RTC alarm event，RTC Time Stamp event This bit is cleared only by a POR/PDR or by setting the WURST bit in the PMU_CTL register.

4. Reset and clock unit (RCU)

4.1. Reset control unit (RCTL)

4.1.1. Overview

GD32F1x0 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. These 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.

4.1.2. 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 1.2V power. 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:

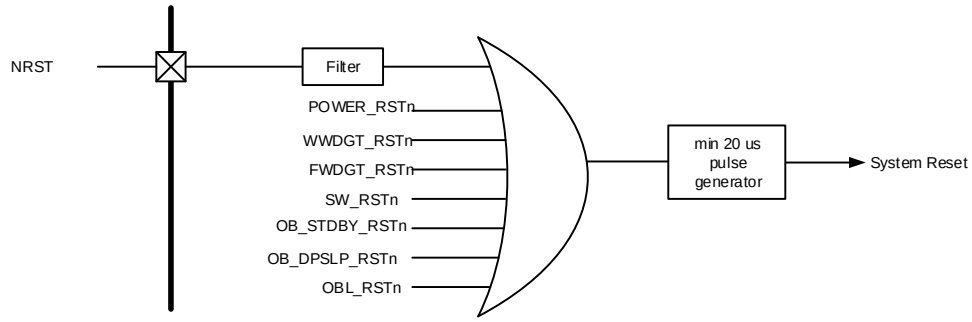
- A power on reset (POWER_RSTn).
- A external pin reset (NRST).
- A window watchdog timer reset (WWDGT_RSTn).
- A free watchdog timer reset (FWDGT_RSTn).
- The SYSRESETREQ bit in Cortex®-M3 application interrupt and reset control register is set (SW_RSTn).
- Option byte loader reset (OBL_RSTn).
- Reset 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_DSLP bit in user option bytes (OB_DPSLP_RSTn).

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 4-1. The system reset circuit



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} or V_{BAT} power on, if both supplies have previously been powered off).

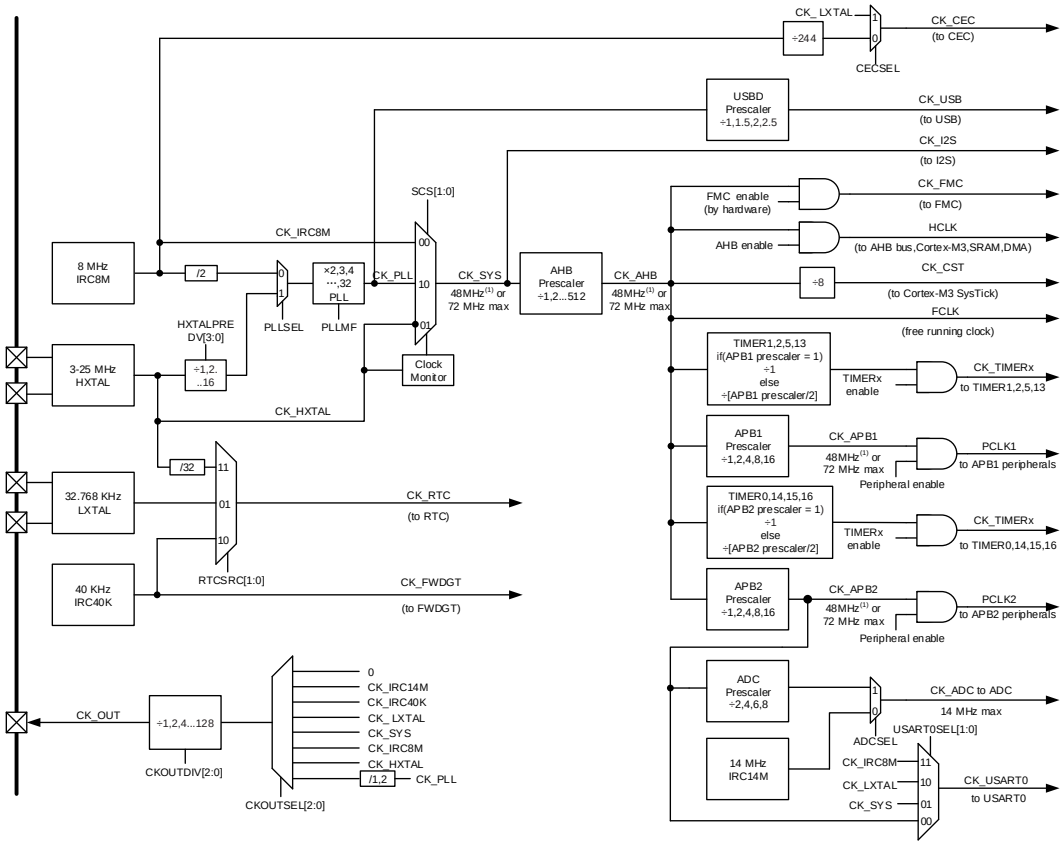
4.2. Clock control unit (CCTL)

4.2.1. Overview

The cllock control unit provides a range of frequencies and clock functions. These include a Internal 8 MHz RC oscillator (IRC8M), a Internal 14 MHz RC oscillator (IRC14M), a high speed crystal oscillator (HXTAL), internal 40KHz RC oscillator (IRC40K), a Low speed crystal oscillator (LXTAL), a Phase Lock Loop (PLL), a HXTAL clock monitor, clock prescalers, clock multiplexers and clock gating circuitry.

The clocks of the AHB, APB and Cortex®-M3 are derived from the system clock (CK_SYS) which can source from the IRC8M, HXTAL or PLL. The maximum operating frequency of GD32F150xx system clock (CK_SYS) can be up to 72 MHz, and the maximum operating frequency of GD32F130xx system clock (CK_SYS) can be up to 48 MHz.

Figure 4-2. Clock tree



Note:(1) The maximum operating frequency of GD32F150xx is 72 MHz, and the maximum operating frequency of GD32F130xx is 48 MHz.

The frequency of AHB, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB and APB2/APB1 domains of the GD32F150xx series is 72 MHz, and the maximum frequency of the GD32F130 is 48 MHz. When using I2Cx peripherals, APB1 clock need to ensure that no more than 36MHz.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 Cortex clock (HCLK), configurable in the systick control and status register.

The ADC are clocked by the clock of APB2 divided by 2, 4, 6, 8 or IRC14M clock selected by ADCSEL bit in configuration register 2 (RCU_CFG2). The USART0 is clocked by IRC8M clock or LXTAL clock or system clock or APB2 clock, which selected by USART0SEL bits in configuration register 2 (RCU_CFG2). The CEC clock is clocked by IRC8M divided 244 or LXTAL clock which selected by CECSEL bit in configuration register 2 (RCU_CFG2).

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.

4.2.2. Characteristics

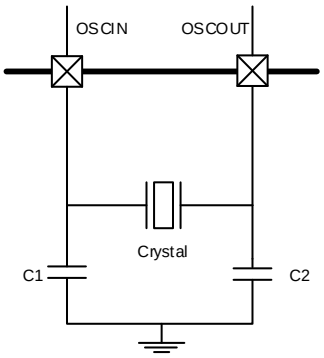
- 3 to 25 MHz high speed crystal oscillator (HXTAL).
- Internal 8 MHz RC oscillator (IRC8M).
- Internal 14 MHz RC oscillator (IRC14M).
- 32,768 Hz Low speed crystal oscillator (LXTAL).
- Internal 40KHz RC oscillator (IRC40K).
- PLL clock source can be HXTAL or IRC8M.
- HXTAL clock monitor.

4.2.3. Function overview

High Speed Crystal Oscillator (HXTAL)

The high speed crystal oscillator (HXTAL), which has a frequency from 3 to 25 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 4-3. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register 0, RCU_CTL0. The HXTALSTB flag in control register 0, RCU_CTL0 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 0, RCU_CTL0. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

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 0, RCU_CTL0. The IRC8MSTB flag in the control register 0, RCU_CTL0 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 16~72 MHz clock output which is 2 ~32 multiples of a fundamental reference frequency of 4 ~ 32 MHz.

The PLL can be switched on or off by using the PLLEN bit in the Control register 0, RCU_CTL0. The PLLSTB flag in the control register 0, RCU_CTL0 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.

Internal 14 MHz RC oscillator (IRC14M)

The internal 14 MHz RC Oscillator, IRC14M, has a fixed frequency of 14 MHz and dedicated as ADC clock. The IRC14M RC oscillator can be switched on or off using the IRC14MEN bit in the control register 1 (RCU_CTL1). The IRC14MSTB flag in the control register 1 (RCU_CTL1) is used to indicate if the internal 14M RC oscillator is stable. An interrupt can be generated if the related interrupt enable bit, IRC14MSTBIE, in the interrupt register, RCU_INT, is set when the IRC14M 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.

Internal 40KHz RC oscillator (IRC40K)

The internal 40KHz 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.

HXTAL clock monitor (CKM)

The HXTAL clock monitor function is enabled by the HXTAL clock monitor enable bit, CKMEN, in the control register 0, RCU_CTL0. This function should be enabled after the HXTAL start-up delay and disabled when the HXTAL is stopped. Once the HXTAL failure is detected, the HXTAL will be automatically disabled. The HXTAL clock stuck flag, CKMIF, in the interrupt register, RCU_INT, will be set and the HXTAL failure event will be generated. This failure interrupt is connected to the non-maskable interrupt, NMI, of the Cortex-M3. 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

Clock output capability

The clock output capability of GD32F130xx is ranging from 32 kHz to 48 MHz, and the clock output capability of GD32F150xx is ranging from 32 kHz to 72 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 4-1. Clock source select

Clock Source Selection bits	Clock Source
000	No Clock
001	CK_IRC14M
010	CK_IRC40K
011	CK_LXTAL

Clock Source Selection bits	Clock Source
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 HDMI CEC or USART0 can wake up the MCU, when their clock is provided by LXTAL clock and LXTAL clock is enable.

If the HDMI CEC or USART0 clock is selected IRC8M clock in deep-sleep mode, they have capable of open IRC8M clock or close IRC8M clock, which used to the HDMI CEC or USART0 to wake up the deep-sleep mode.

Voltage control

The power down reset can be controlled by PDRVS bit in the power down voltage select register (RCU_PDVSEL). If the PDRVS bit is 0, the power down reset assert when the V_{DD} is below 2.6V. If the PDRVS bit is 1, the power down reset assert when the V_{DD} is below 1.8V. When the PDRVS bit is 1 and V_{DD} is below 2.6V, the flash program/erase cannot be used.

The core domain voltage in deep-sleep mode can be controlled by DSLPVS[2:0] bit in the deep-sleep mode voltage register (RCU_DSV).

Table 4-2. Core domain voltage selected in Deep-sleep mode

DSLPVS[2:0]	Deep-sleep mode voltage(V)
000	1.2
001	1.1
010	1.0
011	0.9
100 ~ 111	reserved

The RCU_PDVSEL and RCU_DSV register are protected by Voltage Key register (RCU_VKEY). Only after write 0x1A2B3C4D to the RCU_VKEY register, the RCU_PDVSEL and RCU_DSV register can be write.

4.3. Register definition

RCU base address: 0x4002 1000

4.3.1. Control register 0 (RCU_CTL0)

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				CKMEN	HXTALBPS	HXTALSTB	HXTALEN
						r	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
23:20	Reserved	Must be kept at reset value.
19	CKMEN	HXTAL clock monitor enable 0: Disable the external crystal oscillator (HXTAL) clock monitor 1: Enable the external 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 crystal oscillator disabled 1: External crystal oscillator enabled
15:8	IRC8MCALIB[7:0]	High speed internal oscillator calibration value register These bits are load automatically at power on.
7:3	IRC8MADJ[4:0]	High speed internal 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

4.3.2. Configuration register 0 (RCU_CFG0)

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

PLLDV	CKOUTDIV[2:0]	PLLMF[4]	CKOUTSEL[2:0]	USBDPSC[1:0]	PLLMF[3:0]	PLLPRE DV	PLLSEL
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11	10 9 8	7 6	5 4 3 2	1	0

ADCPSC[1:0]	APB2PSC[2:0]	APB1PSC[2:0]	AHBPSC[3:0]	SCSS[1:0]	SCS[1:0]
rw	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: Internal 14M RC oscillator clock selected 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	USBDPSC[1:0]	USBD clock prescaler selection Set and reset by software to control the USBD clock prescaler value. The USBD clock must be 48MHz. These bits can’t be reset if the USBD clock is enabled. 00: (CK_PLL / 1.5) selected 01: CK_PLL selected 10: (CK_PLL / 2.5) selected 11: (CK_PLL / 2) selected
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 32) **Note:** The PLL output frequency must not exceed 48 MHz for GD32F130xx. and the PLL output frequency must not exceed 72 MHz for GD32F150xx.
17	PLLPREDV	HXTAL divider for PLL source clock selection. This bit is the same bit as bit HXTALPREDV[0] from RCU_CFG1. Refer to RCU_CFG1 HXTALPREDV bits description. Set and cleared by software to divide HXTAL or not which is selected to PLL. 0: HXTAL clock selected 1: (CK_HXTAL / 2) clock selected
16	PLLSEL	PLL clock source selection

		Set and reset by software to control the PLL clock source. 0: (CK_IRC8M / 2) selected as PLL source clock 1: HXTAL selected as PLL source clock
15:14	ADCPSC[1:0]	ADC clock prescaler selection Set and cleared by software. 00: (CK_APB2 / 2) selected 01: (CK_ APB2 / 4) selected 10: (CK_ APB2 / 6) selected 11: (CK_ APB2 / 8) selected
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

4.3.3. 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	Reserved	IRC14MSTBIC	PLLSTBIC	HXTALSTBIC	IRC8MSTBIC	LXTALSTBIC	IRC40KSTBIC
								w		w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		IRC14MSTBIE	PLLSTBIE	HXTALSTBIE	IRC8MSTBIE	LXTALSTBIE	IRC40KSTBIE	CKMIF	Reserved	IRC14MSTBIF	PLLSTBIF	HXTALSTBIF	IRC8MSTBIF	LXTALSTBIF	IRC40KSTBIF
		rw	rw	rw	rw	rw	rw	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	Reserved	Must be kept at reset value.
21	IRC14MSTBIC	IRC14M stabilization interrupt clear Write 1 by software to reset the IRC14MSTBIF flag. 0: Not reset IRC14MSTBIF flag 1: Reset IRC14MSTBIF 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 LXTALRDYF 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
15:14	Reserved	Must be kept at reset value.
13	IRC14MSTBIE	IRC14M stabilization interrupt clear Set and reset by software to enable/disable the IRC14M stabilization interrupt. 0: Disable the IRC14M stabilization interrupt 1: Enable the IRC14M stabilization interrupt
12	PLLSTBIE	PLL stabilization interrupt clear 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 clear 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 clear 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 clear LXTAL stabilization interrupt enable/disable control 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt
8	IRC40KSTBIE	IRC40K stabilization interrupt clear 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. 0: Clock operating normally 1: HXTAL clock stuck
6	Reserved	Must be kept at reset value.
5	IRC14MSTBIF	IRC14M stabilization interrupt flag Set by hardware when the IRC14M is stable and the IRC14MSTBIE bit is set. Reset by software when setting the IRC14MSTBIC bit. 0: No IRC14M stabilization interrupt generated 1: IRC14M stabilization interrupt generated
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 4 ~ 32 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 32kHz 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

4.3.4. 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
Reserved													TIMER16 RST	TIMER15 RST	TIMER14 RST
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART0 RST	Reserved.	SPI0RST	TIMER0 RST	Reserved	ADCRST	Reserved								CFGCMP RST
	rw		rw	rw		rw									rw

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	TIMER16RST	TIMER16 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER16
17	TIMER15RST	TIMER15 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER15
16	TIMER14RST	TIMER14 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER14
14	USART0RST	USART0 Reset This bit is set and reset by software. 0: No reset 1: Reset the USART0
13	Reserved	Must be kept at reset value.
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	Reserved	Must be kept at reset value.
9	ADCRST	ADC reset This bit is set and reset by software. 0: No reset 1: Reset the ADC
8:1	Reserved	Must be kept at reset value.
0	CFGCMPRST	System configuration and comparator reset This bit is set and reset by software. 0: No reset 1: Reset system configuration and comparator

4.3.5. 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	CECRST	DACRST	PMURST	Reserved				USBDRST	I2C1RST	I2C0RST	Reserved			USART1RST	Reserved
	rw	rw	rw					rw	rw	rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	Reserved		WWDGTRST	Reserved		TIMER13RST	Reserved			TIMER5RST	Reserved		TIMER2RST	TIMER1RST
rw	rw			rw			rw				rw			rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CECRST	HDMI CEC reset This bit is set and reset by software. 0: No reset 1: Reset hdmi cec unit
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset DAC unit
28	PMURST	Power control reset This bit is set and reset by software.

		0: No reset 1: Reset power control unit
27:24	Reserved	Must be kept at reset value.
23	USBDRST	USBD reset This bit is set and reset by software. 0: No reset 1: Reset USBD
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
20:18	Reserved	Must be kept at reset value.
17	USART1RST	USART1 reset This bit is set and reset by software. 0: No reset 1: Reset USART1
16	Reserved	Must be kept at reset value.
15	SPI2RST	SPI2 reset This bit is set and reset by software. 0: No reset 1: Reset SPI2
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
10:9	Reserved	Must be kept at reset value.
8	TIMER13RST	TIMER13 timer reset This bit is set and reset by software. 0: No reset

		1: Reset TIMER13 Timer
7:5	Reserved	Must be kept at reset value.
4	TIMER5RST	TIMER5 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER5 Timer
3:2	Reserved	Must be kept at reset value.
1	TIMER2RST	TIMER2 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER2 timer
0	TIMER1RST	TIMER1 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER1 timer

4.3.6. 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							TSIEN	Reserved	PFEN	Reserved	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.									CRCEN	Reserved	FMC SPEN	Reserved	SRAM SPEN	Reserved	DMAEN
									rw		rw		rw		rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	TSIEN	TSI clock enable This bit is set and reset by software. 0: Disabled TSI clock 1: Enabled TSI clock
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	Reserved	Must be kept at reset value.
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
16:7	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 when sleep mode 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	Reserved	Must be kept at reset value.
2	SRAMSPEN	SRAM interface clock enable when sleep mode 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	Reserved	Must be kept at reset value.
0	DMAEN	DMA clock enable This bit is set and reset by software. 0: Disabled DMA clock

1: Enabled DMA clock

4.3.7. 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
	Reserved												TIMER16EN	TIMER15EN	TIMER14EN
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART0EN	Reserved	SPI0EN	TIMER0EN	Reserved	ADCEN		Reserved							CFGCMPEN
	rw		rw	rw		rw									rw

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	TIMER16EN	TIMER16 timer clock enable This bit is set and reset by software 0: Disabled TIMER16 timer clock 1: Enabled TIMER16 timer clock
17	TIMER15EN	TIMER15 timer clock enable This bit is set and reset by software 0: Disabled TIMER15 timer clock 1: Enabled TIMER15 timer clock
16	TIMER14EN	TIMER14 timer clock enable This bit is set and reset by software. 0: Disabled TIMER14 timer clock 1: Enabled TIMER14 timer clock
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	Reserved	Must be kept at reset value.
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	Reserved	Must be kept at reset value.
9	ADCEN	ADC interface clock enable This bit is set and reset by software. 0: Disabled ADC interface clock 1: Enabled ADC interface clock
8:1	Reserved	Must be kept at reset value.
0	CFGCMPEN	System configuration and comparator clock enable This bit is set and reset by software. 0: Disabled System configuration and comparator clock 1: Enabled System configuration and comparator clock

4.3.8. 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	CECEN	DACEN	PMUEN	Reserved				USBDEN	I2C1EN	I2C0EN	Reserved			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
SPI2EN	SPI1EN	Reserved		WWDGTEN	Reserved		TIMER13EN	Reserved			TIMER5EN	Reserved		TIMER2EN	TIMER1EN
rw	rw			rw			rw				rw			rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CECEN	HDMI CEC interface clock enable This bit is set and reset by software. 0: Disabled HDMI CEC interface clock 1: Enabled HDMI CEC interface clock
29	DACEN	DAC interface clock enable This bit is set and reset by software. 0: Disabled DAC interface clock 1: Enabled DAC interface clock

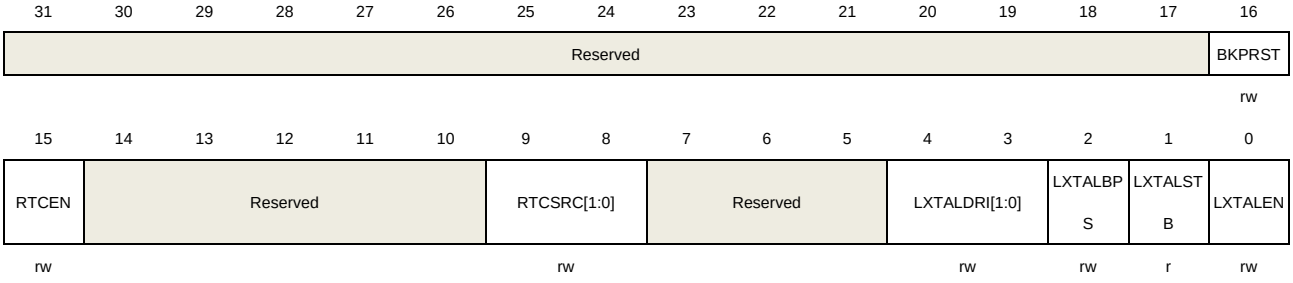
28	PMUEN	Power interface clock enable This bit is set and reset by software. 0: Disabled Power interface clock 1: Enabled Power interface clock
27:24	Reserved	Must be kept at reset value.
23	USBDEN	USBD clock enable This bit is set and reset by software. 0: Disabled USBD clock 1: Enabled USBD clock
22	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
20:18	Reserved	Must be kept at reset value.
17	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
16	Reserved	Must be kept at reset value.
15	SPI2EN	SPI2 clock enable This bit is set and reset by software. 0: Disabled SPI2 clock 1: Enabled SPI2 clock
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
10:9	Reserved	Must be kept at reset value.
8	TIMER13EN	TIMER13 timer clock enable

		This bit is set and reset by software 0: Disabled TIMER13 timer clock 1: Enabled TIMER13 timer clock
7:5	Reserved	Must be kept at reset value.
4	TIMER5EN	TIMER5 timer clock enable This bit is set and reset by software. 0: Disabled TIMER5 timer clock 1: Enabled TIMER5 timer clock
3:2	Reserved	Must be kept at reset value.
1	TIMER2EN	TIMER2 timer clock enable This bit is set and reset by software. 0: Disabled TIMER2 timer clock 1: Enabled TIMER2 timer clock
0	TIMER1EN	TIMER1 timer clock enable This bit is set and reset by software. 0: Disabled TIMER1 timer clock 1: Enabled TIMER1 timer clock

4.3.9. 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.



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 / 32) 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
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

4.3.10. Reset source /clock register (RCU_RSTSCK)

Address offset: 0x24

Reset value: 0x0C00 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

LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	OBLRSTF	RSTFC	V12RSTF	Reserved							
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														IRC40KSTB	IRC40KEN	
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	V12RSTF	1.2V domain power reset flag Set by hardware when a 1.2V domain Power reset generated. Reset by writing 1 to the RSTFC bit. 0: No 1.2V domain Power reset generated 1: 1.2V domain Power reset generated
22:2	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

4.3.11. 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							TSIRST	Reserved	PFRST	Reserved	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.															

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	TSIRST	TSI unit reset This bit is set and reset by software. 0: No reset TSI unit 1: Reset TSI unit

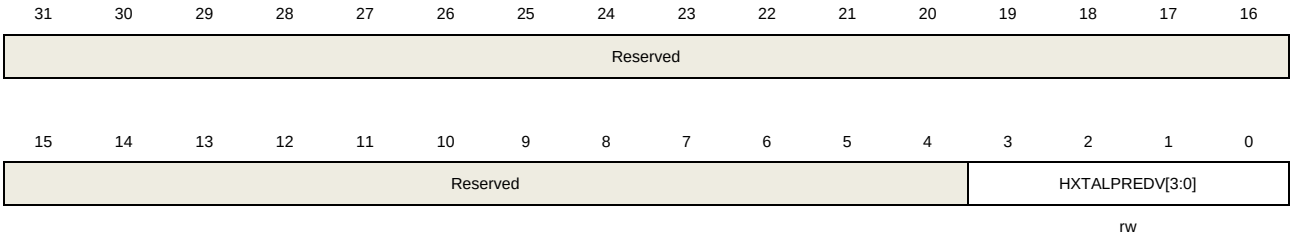
23	Reserved	Must be kept at reset value.
22	PFRST	GPIO port F reset This bit is set and reset by software. 0: No reset GPIO port F 1: Reset GPIO port F
21	Reserved	Must be kept at reset value.
20	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset GPIO port D 1: Reset GPIO port 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
16:0	Reserved	Must be kept at reset value.

4.3.12. 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	HXTALPREDV[3:0]	CK_HXTAL divider previous PLL

This bit is set and reset by software. These bits can be written when PLL is disable

Note: The bit 0 of HXTALPREDV is same as bit 17 of RCU_CFG0, so modifying bit 17 of RCU_CFG0 aslo modifies bit 0 of RCU_CFG2.

The CK_HXTAL is divided by (HXTALPREDV + 1).

0000: HXTAL input to PLL not divided

0001: HXTAL input to PLL divided by 2

0010: HXTAL input to PLL divided by 3

0011: HXTAL input to PLL divided by 4

0100: HXTAL input to PLL divided by 5

0101: HXTAL input to PLL divided by 6

0110: HXTAL input to PLL divided by 7

0111: HXTAL input to PLL divided by 8

1000: HXTAL input to PLL divided by 9

1001: HXTAL input to PLL divided by 10

1010: HXTAL input to PLL divided by 11

1011: HXTAL input to PLL divided by 12

1100: HXTAL input to PLL divided by 13

1101: HXTAL input to PLL divided by 14

1110: HXTAL input to PLL divided by 15

1111: HXTAL input to PLL divided by 16

4.3.13. 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
31:9	Reserved	Must be kept at reset value.
8	ADCSEL	CK_ADC clock source selection This bit is set and reset by software. 0: CK_ADC select CK_IRC14M 1: CK_ADC select CK_APB2 which is divided by 2/4/6/8.
7	Reserved	Must be kept at reset value.
6	CECSEL	CK_CEC clock source selection This bit is set and reset by software.

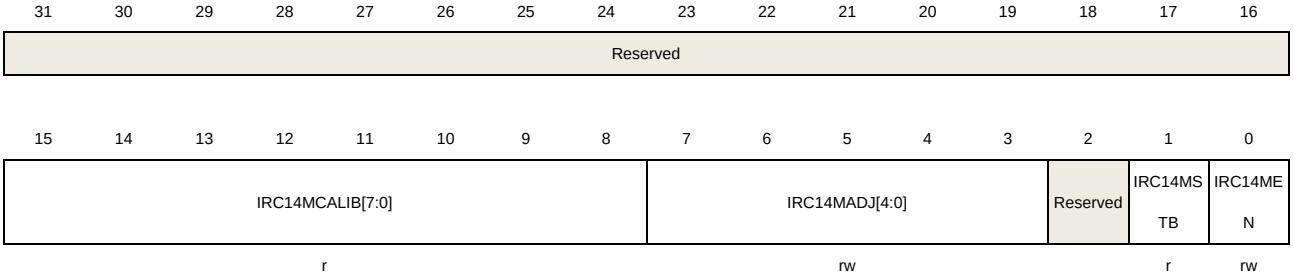
		0: CK_CEC select CK_IRC8M divided by 244 1: CK_CEC select CK_LXTAL
5:2	Reserved	Must be kept at reset value.
1:0	USART0SEL[1:0]	CK_USART0 clock source selection This bit is set and reset by software. 00: CK_USART0 select CK_APB2 01: CK_USART0 select CK_SYS 10: CK_USART0 select CK_LXTAL 11: CK_USART0 select CK_IRC8M

4.3.14. Control register 1 (RCU_CTL1)

Address offset: 0x34

Reset value: 0x0000 XX80 where X is undefined.

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



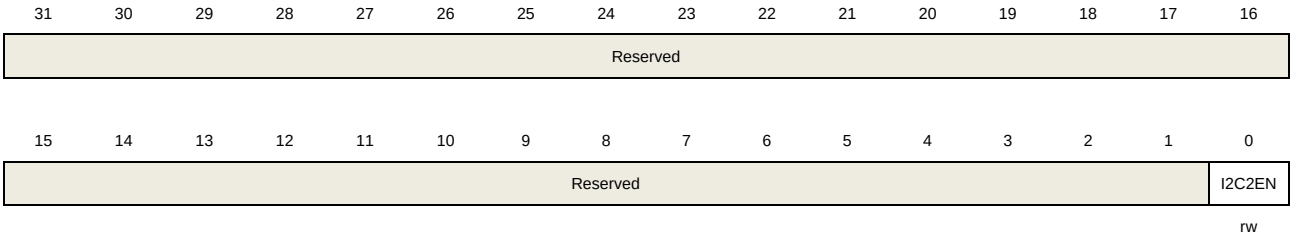
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	IRC14MCALIB[7:0]	Internal 14M RC oscillator calibration value register These bits are load automatically at power on.
7:3	IRC14MADJ[4:0]	Internal 14M RC Oscillator clock trim adjust value These bits are set by software. The trimming value is there bits (IRC14MADJ) added to the IRC14MCALIB[7:0] bits. The trimming value should trim the IRC14M to 14 MHz ± 1%.
2	Reserved	Must be kept at reset value.
1	IRC14MSTB	IRC14M internal 14M RC oscillator stabilization flag Set by hardware to indicate if the IRC14M oscillator is stable and ready for use. 0: IRC14M oscillator is not stable 1: IRC14M oscillator is stable
0	IRC14MEN	IRC14M Internal 14M RC oscillator enable Set and reset by software. 0: Internal 14 MHz RC oscillator disabled 1: Internal 14 MHz RC oscillator enabled

4.3.15. APB1 additional enable register (RCU_ADDAPB1EN)

Address offset: 0xF8

Reset value: 0x0000 0000

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



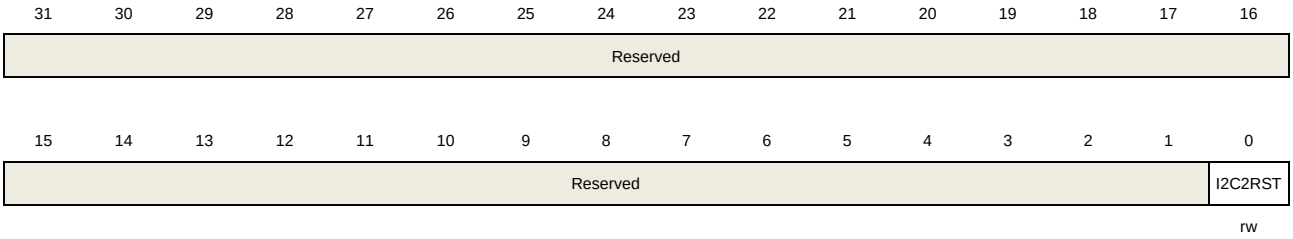
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	I2C2EN	I2C2 unit clock enable This bit is set and reset by software 0: Disable I2C2 unit clock 1: Enable I2C2 unit clock

4.3.16. APB1 additional reset register (RCU_ADDAPB1RST)

Address offset: 0xFC

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	I2C2RST	I2C2 unit reset This bit is set and reset by software 0: Not reset I2C2 unit 1: Reset I2C2 unit

4.3.17. 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_PDVSEL and RCU_DSV register These bits are written only by software and read as 0. Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_PDVSEL and RCU_DSV register can be written.

4.3.18. Deep-sleep mode voltage register (RCU_DSV)

Offset: 0x134

Reset value: 0x0000 0000

Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_DSV register can be written.

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
Reserved													DSLPVS[2:0]		
													rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	DSLPVS[2:0]	Deep-sleep mode voltage select These bits is set and reset by software 000 : The core voltage is 1.2V in Deep-sleep mode 001 : The core voltage is 1.1V in Deep-sleep mode 010 : The core voltage is 1.0V in Deep-sleep mode 011 : The core voltage is 0.9V in Deep-sleep mode 100~111 : Reserved

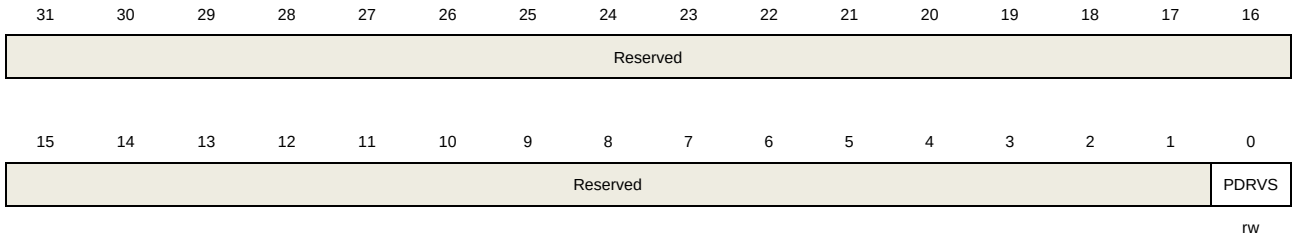
4.3.19. Power down voltage select register (RCU_PDVSEL)

Address offset: 0x138

Reset value: 0x0000 0000.

Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_PDVSEL register can be written.

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



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	PDRVS	Power down voltage select This bit is set and reset by software 0: The Power down voltage is 2.6V 1: The Power down voltage is 1.8V

5. Interrupt / event controller (EXTI)

5.1. Overview

Cortex®-M3 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 processor core. You can read the Technical Reference Manual of Cortex®-M3 for more details about NVIC.

EXTI (interrupt / event controller) contains up to 23 independent edge detectors and generates interrupt requests or events to the processor. 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.

5.2. Characteristics

- Cortex®-M3 system exception.
- Up to 52 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 23 independent edge detectors in EXTI.
- 3 trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

5.3. Interrupts function overview

The Arm Cortex®-M3 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 5-1. NVIC exception types in Cotrex®-M3* and *Table 5-2. Interrupt vector table* list all exception types.

Table 5-1. NVIC exception types in Cotrex®-M3

Exception type	Vector number	Priority (a)	Vector address	Description
-	0	-	0x0000_0000	Reserved

Exception type	Vector number	Priority (a)	Vector address	Description
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

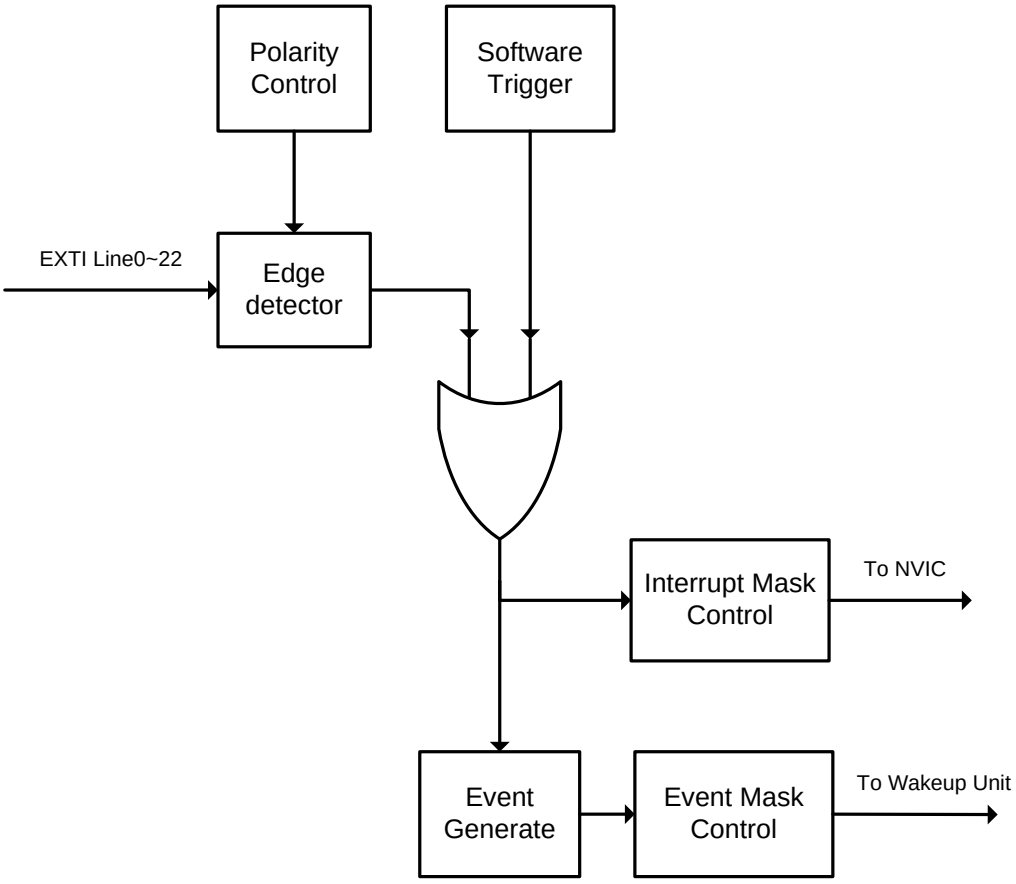
Table 5-2. Interrupt vector table

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	LVD from EXTI interrupt	0x0000_0044
IRQ 2	18	RTC global interrupt	0x0000_0048
IRQ 3	19	FMC global interrupt	0x0000_004C
IRQ 4	20	RCU global interrupt	0x0000_0050
IRQ 5	21	EXTI line0-1 interrupts	0x0000_0054
IRQ 6	22	EXTI line2-3 interrupts	0x0000_0058
IRQ 7	23	EXTI line4-15 interrupts	0x0000_005C
IRQ 8	24	TSI global interrupt	0x0000_0060
IRQ 9	25	DMA channel0 global interrupt	0x0000_0064
IRQ 10	26	DMA channel1-2 global interrupts	0x0000_0068
IRQ 11	27	DMA channel3-4 global interrupts	0x0000_006C
IRQ 12	28	ADC and CMP0-1 interrupts	0x0000_0070
IRQ 13	29	TIMER0 break, update, trigger and commutation interrupts	0x0000_0074
IRQ 14	30	TIMER0 capture compare interrupt	0x0000_0078
IRQ 15	31	TIMER1 global interrupt	0x0000_007C
IRQ 16	32	TIMER2 global interrupt	0x0000_0080
IRQ 17	33	TIMER5 and DAC global interrupts	0x0000_0084
IRQ 18	34	Reserved	0x0000_0088

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 19	35	TIMER13 global interrupt	0x0000_008C
IRQ 20	36	TIMER14 global interrupt	0x0000_0090
IRQ 21	37	TIMER15 global interrupt	0x0000_0094
IRQ 22	38	TIMER16 global interrupt	0x0000_0098
IRQ 23	39	I2C0 event interrupt	0x0000_009C
IRQ 24	40	I2C1 event interrupt	0x0000_00A0
IRQ 25	41	SPI0 global interrupt	0x0000_00A4
IRQ 26	42	SPI1 global interrupt	0x0000_00A8
IRQ 27	43	USART0 global interrupt	0x0000_00AC
IRQ 28	44	USART1 global interrupt	0x0000_00B0
IRQ 29	45	Reserved	0x0000_00B4
IRQ 30	46	CEC global interrupt	0x0000_00B8
IRQ 31	47	Reserved	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	Reserved	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	I2C2 event interrupt	0x0000_00CC
IRQ 36	52	I2C2 error interrupt	0x0000_00D0
IRQ 37	53	USBD low priority interrupts	0x0000_00D4
IRQ 38	54	USBD high priority interrupts	0x0000_00D8
IRQ 39-41	55-57	Reserved	0x0000_00DC-0x0000_00E4
IRQ 42	58	USBD wake up from EXTI interrupt	0x0000_00E8
IRQ 43-47	59-63	Reserved	0x0000_00EC-0x0000_00FC
IRQ 48	64	DMA channel5-6 global interrupts	0x0000_0100
IRQ 49-50	65-66	Reserved	0x0000_0104-0x0000_0108
IRQ 51	67	SPI2 global interrupt	0x0000_010C

5.4. External interrupt and event block diagram

Figure 5-1. Block diagram of EXTI.



5.5. External interrupt and event function overview

The EXTI contains up to 23 independent edge detectors and generates interrupt requests or events to the processor. 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 8 lines from internal modules which refers to ***Table 5-3. EXTI source*** for detail. All GPIO pins can be selected as an EXTI trigger source by configuring SYSCFG_EXTISSx registers in SYSCFG module (please refer to ***System configuration registers (SYSCFG)*** for detail).

EXTI can provide not only interrupts but also event signals to the processor. The Cortex®-M3 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 events. EXTI can be used to wake up the processor and the whole system when some expected events occur, such as a special I/O 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 5-3. EXTI source

EXTI line number	Source
0	PA0 / PB0 / PC0 / PF0
1	PA1 / PB1 / PC1 / PF1
2	PA2 / PB2 / PC2 / PD2
3	PA3 / PB3 / PC3
4	PA4 / PB4 / PC4 / PF4
5	PA5 / PB5 / PC5 / PF5
6	PA6 / PB6 / PC6 / PF6
7	PA7 / PB7 / PC7 / PF7
8	PA8 / PB8 / PC8
9	PA9 / PB9 / PC9
10	PA10 / PB10 / PC10
11	PA11 / PB11 / PC11
12	PA12 / PB12/ PC12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC alarm

EXTI line number	Source
18	USBD wakeup
19	RTC tamper and timestamp
20	Reserved
21	CMP0 output
22	CMP1 output
23	Reserved
24	Reserved
25	USART0 wakeup
26	Reserved
27	CEC wakeup

5.6. Register definition

EXTI base address: 0x4001 0400

5.6.1. Interrupt Enable register (EXTI_INTEN)

Address offset: 0x00

Reset value: 0x0F90 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				INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
				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
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
31:28	Reserved	Must be kept at reset value.
27: 0	INTENx	Interrupt enable bit x (x = 0…27) 0: Interrupt from linex is disabled 1: Interrupt from linex is enabled

5.6.2. 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				EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
				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
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
31:28	Reserved	Must be kept at reset value.
27: 0	EVENx	Event enable bit x (x = 0…27) 0: Event from linex is disabled 1: Event from linex is enabled

5.6.3. 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									RTEN22	RTEN21	Reserved	RTEN19	RTEN18	RTEN17	RTEN16
									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
31:23	Reserved	Must be kept at reset value.
22:21	RTENx	Rising edge trigger enable (x = 21, 22) 0: Rising edge of linex is not valid 1: Rising edge of linex is valid as an interrupt / event request
20	Reserved	Must be kept at reset value.
19	RTENx	Rising edge trigger enable (x = 19) 0: Rising edge of linex is not valid 1: Rising edge of linex is valid as an interrupt / event request
18	RTENx	Rising edge trigger enable (x = 18) This bit is valid only for GD32F150xx device. 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt / event request
17: 0	RTENx	Rising edge trigger enable (x = 0…17) 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt / event request

5.6.4. 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									FTEN22	FTEN21	Reserved	FTEN19	FTEN18	FTEN17	FTEN16
									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: 23	Reserved	Must be kept at reset value.
22: 21	FTENx	Falling edge trigger enable (x = 21, 22) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request
20	Reserved	Must be kept at reset value.
19	FTENx	Falling edge trigger enable (x = 19) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request
18	FTENx	Falling edge trigger enable (x = 18) This bit is valid only for GD32F150xx device. 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request
17: 0	FTENx	Falling edge trigger enable (x = 0…17) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request

5.6.5. 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									SWIEV22	SWIEV21	Reserved	SWIEV19	SWIEV18	SWIEV17	SWIEV16
									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
31:23	Reserved	Must be kept at reset value.
22: 21	SWIEVx	Interrupt / Event software trigger (x = 21, 22) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request
20	Reserved	Must be kept at reset value.
19: 0	SWIEVx	Interrupt / Event software trigger (x = 0…19) 0: Deactivate the EXTIx software interrupt / event request

1: Activate the EXTIx software interrupt / event request

5.6.6. Pending register (EXTI_PD)

Address offset: 0x14

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									PD22	PD21	Reserved	PD19	PD18	PD17	PD16
									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: 23	Reserved	Must be kept at reset value.
22: 21	PDx	Interrupt pending status (x = 21, 22) 0: EXIT linex is not triggered 1: EXIT linex is triggered This bit is cleared to 0 by writing 1 to it.
20	Reserved	Must be kept at reset value.
19: 0	PDx	Interrupt pending status (x=0…19) 0: EXIT linex is not triggered 1: EXIT linex is triggered This bit is cleared to 0 by writing 1 to it.

6. General-purpose I/Os (GPIO)

6.1. Overview

There are up to 55 general purpose I/O pins, (GPIO), named PA0 ~ PA15 and PB0 ~ PB15, PC0 ~ PC15, PD2, PF0, PF1, PF4 ~ 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 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 mode.

Each of the GPIO pins can be configured by software as output (push-pull or open-drain), as input, as peripheral alternate function or as 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.

6.2. Characiteristics

- 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.
- Output drive speed selection.
- Analog input/output configurations.
- Alternate function input/output configurations.
- Port configuration lock.

6.3. 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 control register (GPIOx_CTL). AFIO input or output direction is decided by AFIO function after AFIO 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).

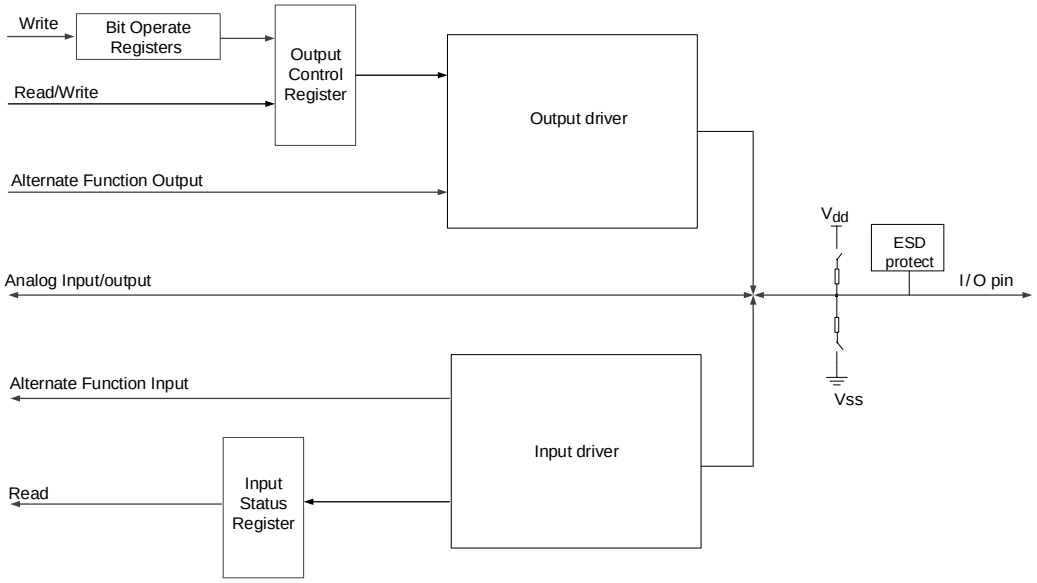
Table 6-1. GPIO configuration table

PAD TYPE			CTLn	OMn	PUDn
GPIO	X	Floating	00	X	00

INPUT		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 6-1. Basic structure of a general-pupose I/O shows the basic structure of an I/O port bit.

Figure 6-1. Basic structure of a general-pupose I/O



6.3.1. 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 Serial-Wired Debug pins are in AF PU/PD mode after reset:

PA14: SWCLK in AF pull-down mode

PA13: SWDIO in AF pull-up mode

The GPIO pins can be configured as inputs or outputs. And all GPIO pins have an internal weak pull-up and weak pull-down which can be chosen. When the GPIO pins are configured as input pins, the data on the external pads 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.

When programming the GPIOx_OCTL at bit level is not need to disable interrupts, 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.

6.3.2. Alternate functions (AF)

When the port is configured as AFIO (set CTLy to "10" bits, 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 select registers (GPIOx_AFSELy(y=0,1)). The detail alternate function assignments for each port are in the device datasheet.

6.3.3. Additional functions

Some pins have additional functions, which have priority over the configuration in the standard GPIO registers. When for ADC or DAC 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.

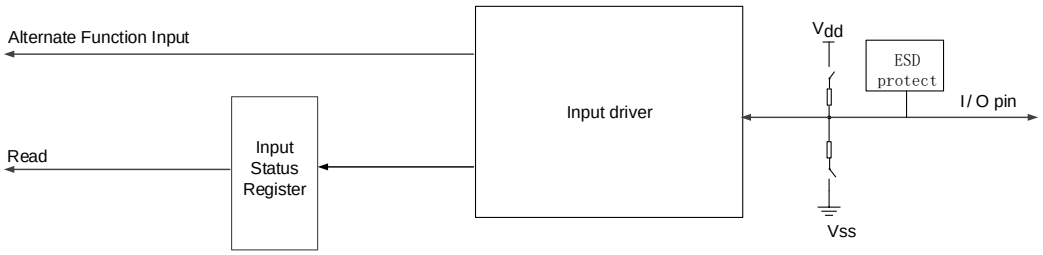
6.3.4. Input configuration

When GPIO pin is configured as input:

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

Figure 6-2. Basic structure of Input configuration shows the input configuration of the I/O Port bit.

Figure 6-2. Basic structure of Input configuration



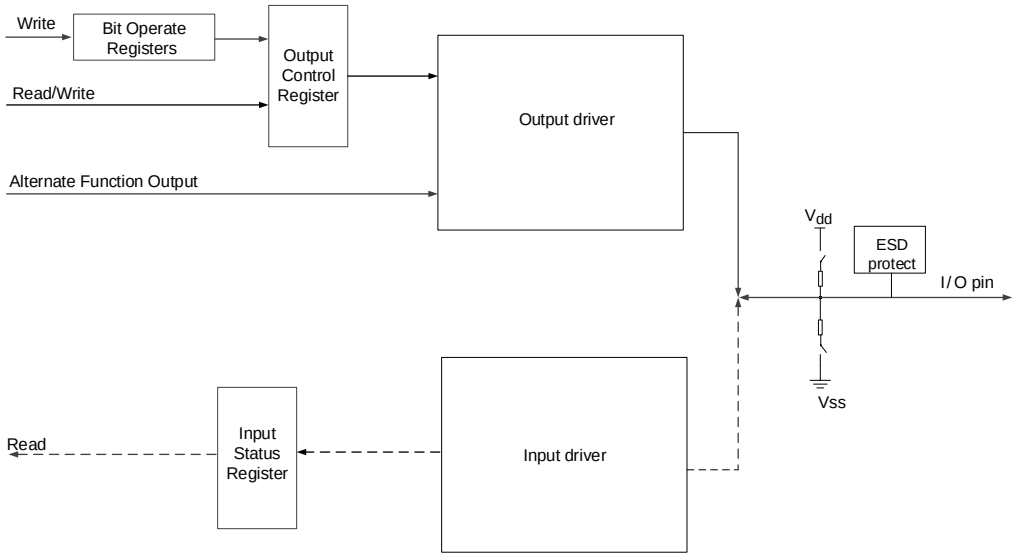
6.3.5. Output configuration

When GPIO pin is configured as output:

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

Figure 6-3. Basic structure of Output configuration shows the output configuration of the I/O port bit.

Figure 6-3. Basic structure of Output configuration



6.3.6. 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 de-activated.
- Read access to the port input status register gets the value “0”.

Figure 6-4. Basic structure of Analog configuration shows the high impedance-analog configuration.

Figure 6-4. Basic structure of Analog configuration



6.3.7. Alternate function (AF) configuration

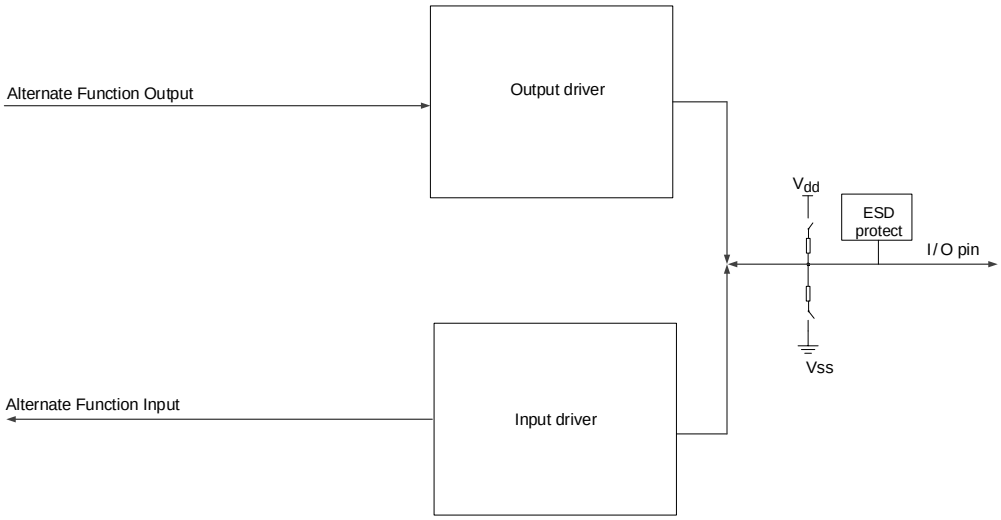
To suit for different device packages, the GPIO supports some alternate functions to some other pins by software.

When be configured as Alternate Function:

- The output buffer is turned on in Open-Drain or Push-Pull configuration.
- The output buffer is driven by the peripheral.
- The schmitt trigger input is activated.
- The weak pull-up and pull-down resistors could be chosen.
- The data present on the I/O pin is sampled into the port input status register every AHB clock cycle.
- A read access to the port input status register gets the I/O state in Open_Drain mode.
- A read access to the port output control register gets the last written value in Push-Pull mode.

Figure 6-5. Basic structure of Alternate function configuration shows the alternate function configuration of the I/O Port bit.

Figure 6-5. Basic structure of Alternate function configuration



6.3.8. 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, GPIOx_AFSELy(y=0,1). It allows the I/O configuration to be frozen by the 32-bit locking register (GPIOx_LOCK). When the LOCK sequence has been applied on a port bit, it is no longer able to modify the value of the port bit until the next reset. It should be recommended to be used in the configuration of driving a power module.

6.3.9. 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.

6.4. Register definition

GPIOA base address: 0x4800 0000
GPIOB base address: 0x4800 0400
GPIOC base address: 0x4800 0800
GPIOD base address: 0x4800 0C00
GPIOF base address: 0x4800 1400

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

Address offset: 0x00
Reset value: 0x2800 0000 for port A; 0x0000 0000 for others.

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).
This register can only be read by 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

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

Address offset: 0x04

Reset value: 0x0000 0000

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).

This register can only be read 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
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

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

Address offset: 0x08

Reset value: 0x0C00 0000 for port A; 0x0000 0000 for others.

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).

This register can only be read by 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

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

Address offset: 0x0C

Reset value: 0x2400 0000 for port A; 0x0000 0000 for others.

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).

This register can only be read by 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]	Pin15 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

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

Address offset: 0x10

eset value: 0x0000 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
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[15:0]	Port input status (y=0..15) These bits are set and cleared by hardware. 0: Input signal low 1: Input signal high

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

Address offset: 0x14

Reset value: 0x0000 0000

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).

This register can only be read 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
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

6.4.7. Port bit operate register (GPIOx_BOP, x=A..D,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

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

6.4.8. Port configuration lock register (GPIOx_LOCK, x=A, B)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).

This register can only be read 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 using the Lock Key Writing Sequence.And can always be read. 0: GPIOx_LOCK register is not locked and the port configuration is 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 hold 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

6.4.9. Alternate function selected register0 (GPIOx_AFSEL0, x=A, B, C)

Address offset: 0x20

Reset value: 0x0000 0000

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).

This register can only be read by 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 (Port A,B only)
0101: AF5 selected (Port A,B only)
0110: AF6 selected (Port A,B only)
0111: AF7 selected (Port A,B only)
1000 ~ 1111: Reserved

6.4.10. Alternate function selected register1 (GPIOx_AFSEL1, x=A,B,C)

Address offset: 0x24
Reset value: 0x0000 0000

This register can be written by byte (8-bit), half word (16-bit) or word (32-bit).
This register can only be read by 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 1 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 (Port A,B only) 0101: AF5 selected (Port A,B only) 0110: AF6 selected (Port A,B only) 0111: AF7 selected (Port A,B only) 1000 ~ 1111: Reserved

6.4.11. Bit clear register (GPIOx_BC, x=A..D,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

7. Cyclic redundancy checks management unit (CRC)

7.1. Overview

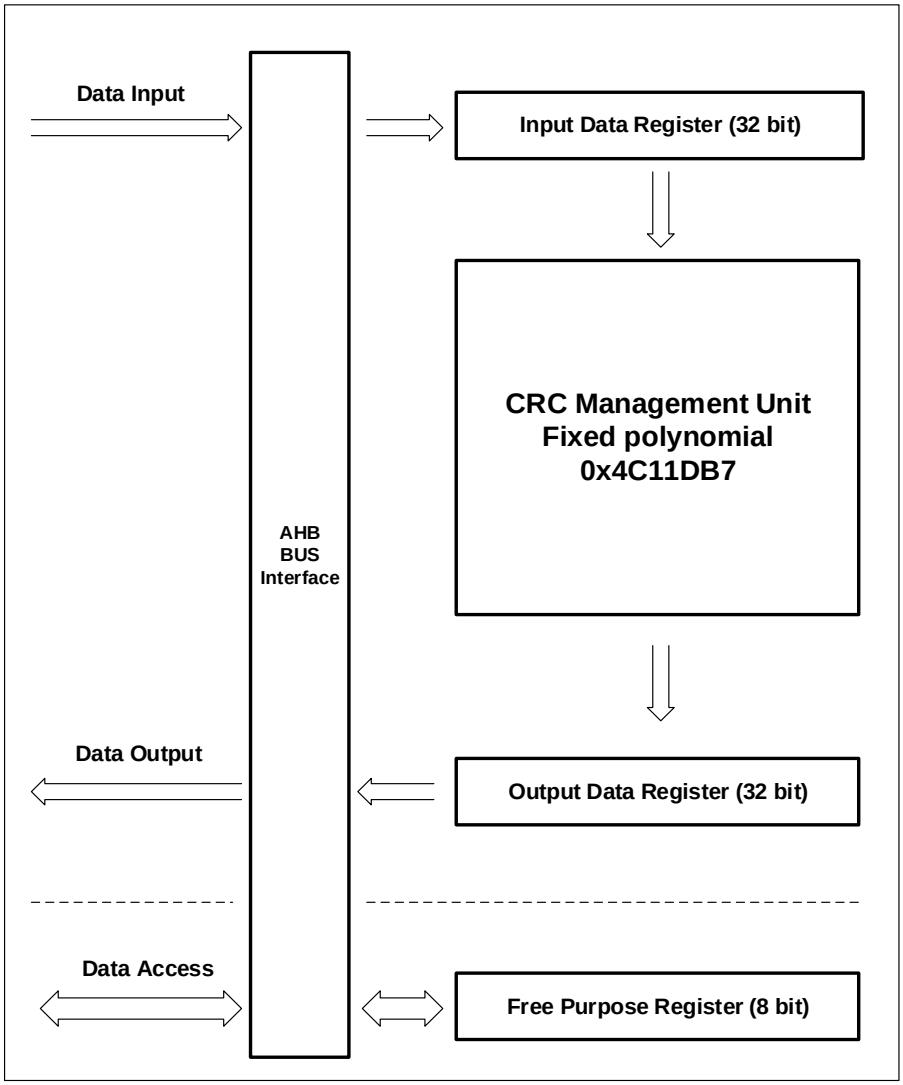
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

This CRC management unit can be used to calculate 32/16/8 bit CRC code within fixed polynomial.

7.2. Characteristics

- Input data supports 8/16/32 size bit.
- Different input size for different calculation time.1/2/4 cycle for 8/16/32 bits.
- Input and output data can be reversed.
- Fixed polynomial: 0x4C11DB7
  X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 +X8 + X7 + X5 + X4 + X2+ X + 1
  This 32-bit CRC polynomial is a common polynomial used in Ethernet.
- User configurable initial value after CRC reset.
- Free 8 bit register is unrelated for calculation and can be used for any other goals by any other peripheral devices.

Figure 7-1. Block Diagram of CRC management unit



7.3. Function overview

- CRC management 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 do not clear the CRC_DATA register by software setting CRC_CTL register, the new input raw data will calculate based on the result of previous value of CRC_DATA.
  CRC calculation will spend 4/2/1 AHB clock cycles for 32/16/8 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, any value you write in will be read out at anytime.
- Reversible function can reverse the input data and output data.
  For input data, 3 reverse types can be selected.
  Original data is 0x3456CDEF:
  1) byte reverse:

32-bit data is divided into 4 groups and reverse implement in group inside. Reversed data: 0x2C6AB3F7

2)half-word reverse:

32-bit data is divided into 2 groups and reverse implement in group inside. Reversed data: 0x6A2CF7B3

3)word reverse:

32-bit data is divided into 1 groups and reverse implement in group inside. Reversed data: 0xF7B36A2C

For output data, reverse type is word reverse.

For example: when REV_O=1, calculation result 0x3344CCDD will be converted to 0xBB3322CC.

- Multiple input data size support function will make users have the flexibility to adjust the combination of the calculation data.

  For example: 6 bytes input data can be combined by 1 word and 1 half-word, also it can be combined by 3 half-word.
- User configurable initial calculation data function will support user calculate CRC data value under any initial value.

7.4. Register definition

CRC base address: 0x4002 3000

7.4.1. Data Register (CRC_DATA)

Address offset: 0x00
Reset value: 0xFFFF FFFF

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[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software write and read. 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 CRC calculation result.

7.4.2. Free Data Register (CRC_FDATA)

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								FDATA[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free Data Register Bits Software write and read. These bits are unrelated with CRC calculation. This byte can be used for any goals

by any other peripheral. The CRC_CTL register will generate no effect to the byte.

7.4.3. Control Register (CRC_CTL)

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								REV_O	REV_I[1:0]		Reserved				RST
								rw	rw						rs

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
4:1	Reserved	Must be kept at reset value.
0	RST	This bit can reset the CRC_DATA register to the value in CRC_IDATA then automatically cleared itself to 0 by hardware. This bit will generate no effect to CRC_FDATA. Software write and read.

7.4.4. Initialization Data Register (CRC_IDATA)

Address offset: 0x10
Reset value: 0xFFFF FFFF

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
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.

8. Direct memory access controller (DMA)

8.1. 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 7 channels in the DMA controller. 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.

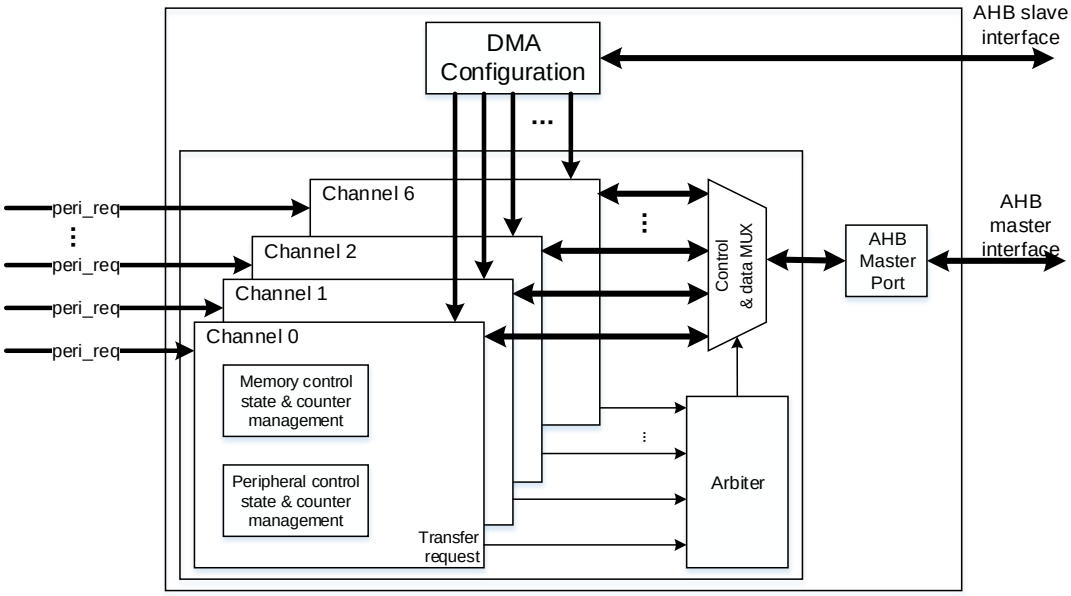
The system bus is shared by the DMA controller and the Cortex®-M3 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system bus bandwidth for the CPU.

8.2. Characteristics

- Programmable length of data to be transferred, max to 65536.
- 7 channels and each channel is 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.

8.3. Block diagram

Figure 8-1. Block diagram of DMA



As shown in ***Figure 8-1. 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.

8.4. Function overview

8.4.1. 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 8-1. DMA transfer operation***.

Table 8-1. 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[7:0] @0x0 2: Write B5B4[7:0] @0x2 3: Write B9B8[7:0] @0x4 4: Write BDBC[7:0] @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 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 will not launch any DMA transfer.

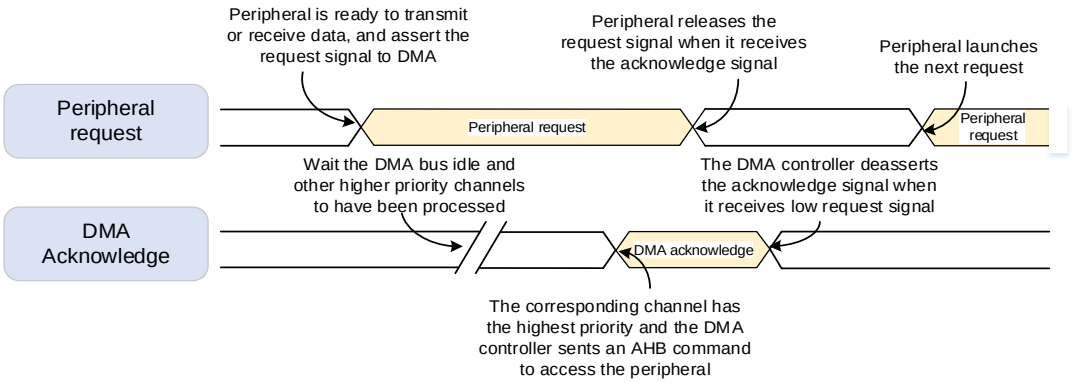
8.4.2. Peripheral handshake

To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and a 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 8-2. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 8-2. Handshake mechanism



8.4.3. 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.

8.4.4. 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.

8.4.5. 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.

8.4.6. 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.

8.4.7. 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.

8.4.8. 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.

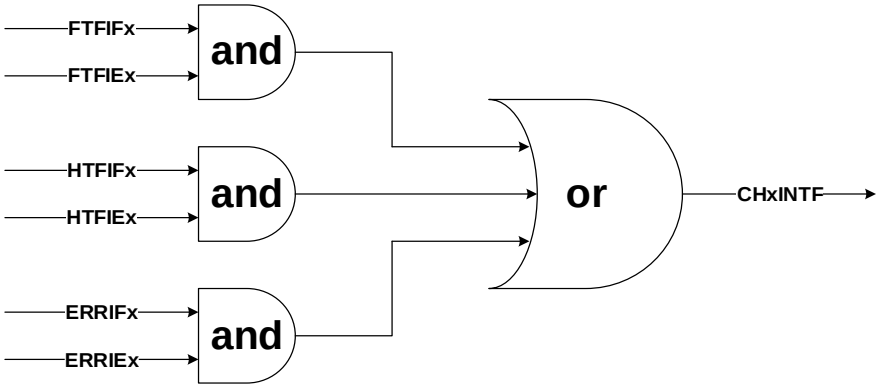
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following ***Table 8-2. Interrupt events***.

Table 8-2. 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 8-3. DMA interrupt logic***, an interrupt can be produced when any type of interrupt event occurs and enabled on the channel.

Figure 8-3. DMA interrupt logic



NOTE: "x" indicates channel number (x=0…6).

8.4.9. DMA request mapping

Several requests from peripherals may be mapped to one DMA channel. They are logically ORed before entering the DMA. For details, see the following ***Figure 8-4. DMA request mapping***. The request of each peripheral can be independently enabled or disabled by

programming the registers of the corresponding peripheral. The user has to ensure that only one request is enabled at a time on one channel. ***Table 8-3. DMA requests for each channel*** lists the support request from peripheral for each channel of DMA.

Figure 8-4. DMA request mapping

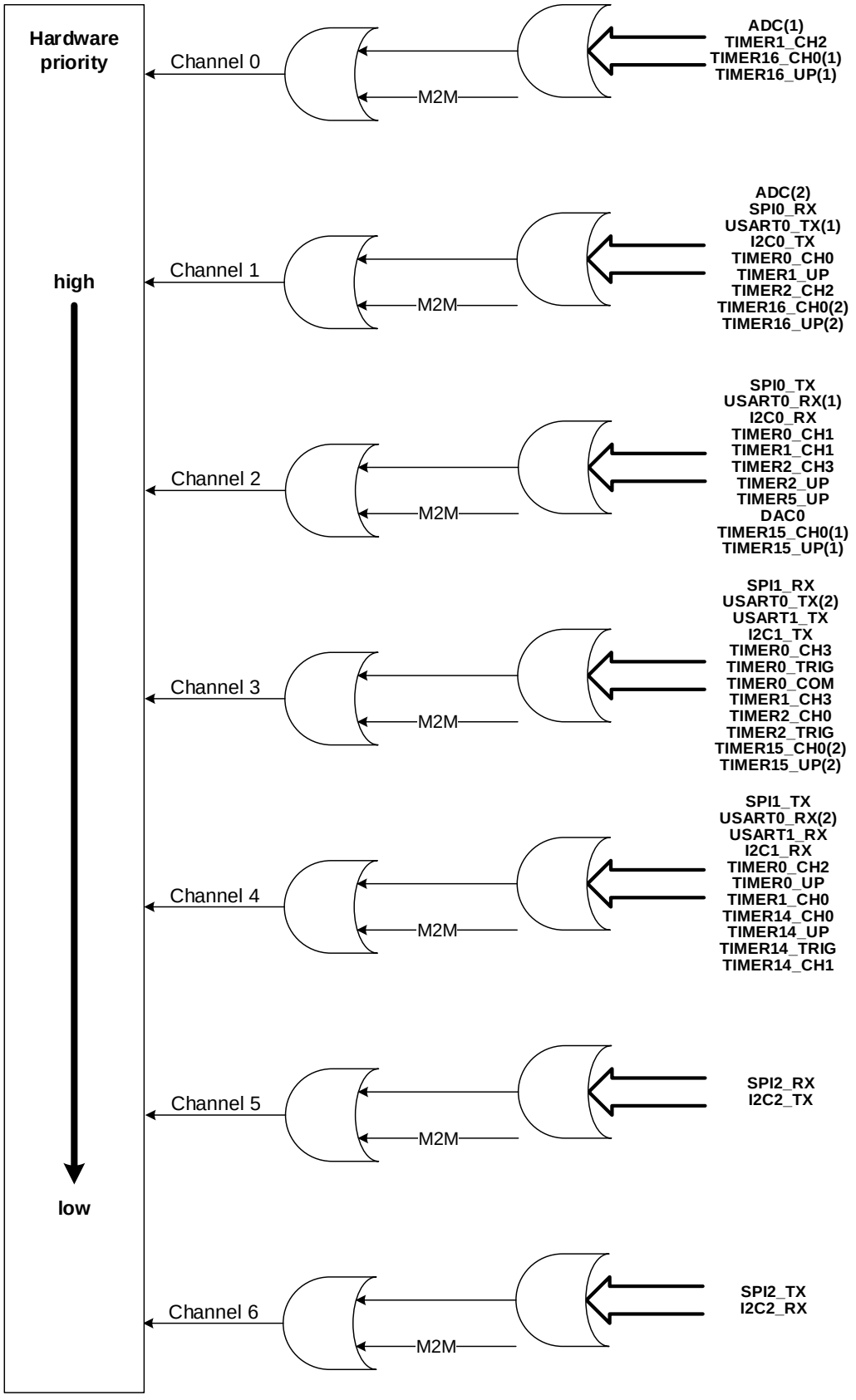


Table 8-3. DMA requests for each channel

Peripheral	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
ADC	ADC(1)	ADC(2)	•	•	•	•	•
SPI/I2S	•	SPI/I2S0_RX	SPI/I2S0_TX	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX
USART	•	USART0_TX(1)	USART0_RX(1)	USART0_TX(2) USART1_TX	USART0_RX(2) USART1_RX	•	•
I²C	•	I2C0_TX	I2C0_RX	I2C1_TX	I2C1_RX	I²C2_TX	I²C2_RX
TINER0	•	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TRIG TIMER0_COM	TIMER0_CH2 TIMER0_UP	•	•
TIMER1	TIMER1_CH2	TIMER1_UP	TIMER1_CH1	TIMER1_CH3	TIMER1_CH0	•	•
TIMER2	•	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	TIMER2_CH0 TIMER2_TRIG	•	•	•
TIMER5/ DAC	•	•	TIMER5_UP DAC	•	•	•	•
TIMER14	•	•	•	•	TIMER14_CH0 TIMER14_UP TIMER14_TRIG TIMER14_CH1	•	•
TIMER15	•	•	TIMER15_CH0(1) TIMER15_UP(1)	TIMER15_CH0(2) TIMER15_UP(2)	•	•	•
TIMER16	TIMER16_CH0(1) TIMER16_UP(1)	TIMER16_CH0(2) TIMER16_UP(2)	•	•	•	•	•

1. When the corresponding remapping bit in the SYSCFG_CFGR0 register is cleared, the request is mapped on the channel.
2. When the corresponding remapping bit in the SYSCFG_CFGR0 register is set, the request is mapped on the channel.

8.5. Register definition

DMA base address: 0x4002 0000

8.5.1. 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

8.5.2. 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
31:28	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

8.5.3. 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

8.5.4. 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).

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]	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.

8.5.5. 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.

8.5.6. 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.

9. Debug (DBG)

9.1. Overview

The GD32F1x0 series provide a large variety of debug, trace and test features. They are implemented with a standard configuration of the ARM CoreSight™ module together with a daisy chained standard TAP controller. Debug and trace functions are integrated into the ARM Cortex-M3. The debug system supports serial wire debug (SWD) and trace functions. The debug and trace functions refer to the following documents:

- Cortex-M3 Technical Reference Manual
- ARM Debug Interface v5 Architecture Specification

The DBG hold unit helps debugger to debug in power saving mode and some peripherals, including TIMER, I2C, RTC, WWDGT and FWDGT. When corresponding bit is set, provide clock when in power saving mode or hold the state for TIMER, I2C, RTC, WWDGT, and FWDGT.

9.2. Serial Wire Debug port overview

Debug capabilities can be accessed by a debug tool via Serial Wire (SW - Debug Port).

9.2.1. Pin assignment

The synchronous serial wire debug (SWD) provides 2-pin SW interface, known as SW data input/output (SWDIO) and SW clock (SWCLK).

The pin assignment is as following:

PA14: SWCLK

PA13: SWDIO

If SWD not used, all 2-pin can be released to other GPIO functions. Please refer to ***GPIO pin configuration***

9.2.2. JEDEC-106 ID code

The Cortex-M3 integrates JEDEC-106 ID code, which is located in ROM table mapped to the address of 0xE00FF000_0xE00FFFFF.

9.3. Debug hold function overview

9.3.1. Debug support for power saving mode

When STB_HOLD bit in DBG control register 0 (DBG_CTL0) 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 0 (DBG_CTL0) 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 0 (DBG_CTL0) 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.

9.3.2. Debug support for TIMER, I2C, RTC, WWDGT and FWDGT

When the core halted and the corresponding bit in DBG control register 0 or 1 (DBG_CTL0 or DBG_CTL1) is set, the following events occur..

For TIMER, the timer counters are stopped and held for debugging.

For I2C, SMBUS timeout is held for debugging.

For RTC, the counter is stopped for debugging.

For WWDGT or FWDGT, the counter clock is stopped for debugging.

9.4. DBG registers

DBG base address: 0xE004 2000

9.4.1. ID code register (DBG_ID)

Address: 0xE004 2000

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 can only be read by software, These bits are unchanged constant.

9.4.2. Control register 0(DBG_CTL0)

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
Reserved				TIMER13_HOLD	Reserved							TIMER5_HOLD	Reserved	I2C2 HOLD	I2C1_HOLD
				rw								rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	Reserved		TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	Reserved					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw			rw	rw	rw	rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27	TIMER13_HOLD	TIMER13 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER13 counter for debug when core is halted
26:20	Reserved	Must be kept at reset value
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 is halted
18	Reserved	Must be kept at reset value
17	I2C2_HOLD	I2C2 hold bit This bit is set and reset by software. 0: no effect 1: hold the I2C2 SMBUS timeout for debug when core is 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 is 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 is halted
14:13	Reserved	Must be kept at reset value
12	TIMER2_HOLD	TIMER2 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER2 counter for debug when core is halted
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 is 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 is 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 is 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 is halted
2	STB_HOLD	Standby mode hold bit This bit is set and reset by software. 0: no effect

		1: in the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, a system reset generated when exiting standby mode.
1	DSLP_HOLD	Deep-sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M.
0	SLP_HOLD	Sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the sleep mode, the clock of AHB is on.

9.4.3. Control register 1 (DBG_CTL1)

Address offset: 0x08

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
Reserved													TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
													rw	rw	rw

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

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	TIMER16_HOLD	TIMER16 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER16 counter for debug when core is halted
17	TIMER15_HOLD	TIMER15 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER15 counter for debug when core is halted
16	TIMER14_HOLD	TIMER14 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER14 counter for debug when core is halted
15:11	Reserved	Must be kept at reset value

10	RTC_HOLD	RTC hold bit This bit is set and reset by software. 0: no effect 1: hold the RTC counter for debug when core is halted
9:0	Reserved	Must be kept at reset value

10. Analog to digital converter (ADC)

10.1. 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, 2 internal channels and the battery voltage (V_{BAT}) channel. The 19 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 alignment(MSB).

10.2. Characteristics

- High performance.
  - ADC sampling rsolution: 12-bit.
  - Foreground calibration function.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit.
  - DMA support.
- Dual clock domain architecture (APB clock and ADC clock).
- Analog input channels.
  - 16 external analog inputs.
  - 1 channel for internal temperature sensor (VSENSE).
  - 1 channel for internal reference voltage (VREFINT).
  - 1 channel for monitoring external VBAT power supply pin.
- Start of the conversion can be initiated.
  - By software.
  - By hardware triggers.
- Operation modes.
  - Convert a single channel or scan a sequence of channels.
  - Single operation mode converts selected inputs once for per trigger.
  - Continuous operation mode converts selected inputs continuously.
  - Discontinuous operation mode.
- Interrupt generation.
  - At the end of routine conversions.
  - Analog watchdog event.
- Conversion result threshold monitor function: analog watchdog.
- Channel input range: $V_{SSA} \leq V_{IN} \leq V_{DDA}$.

10.3. Pins and internal signals

Figure 10-1. ADC module block diagram show the ADC block diagram.*Table 10-1. ADC internal input signals*, *Table 10-2. ADC input pins definition* give the ADC internal signals and pins description.

Table 10-1. ADC internal input signals

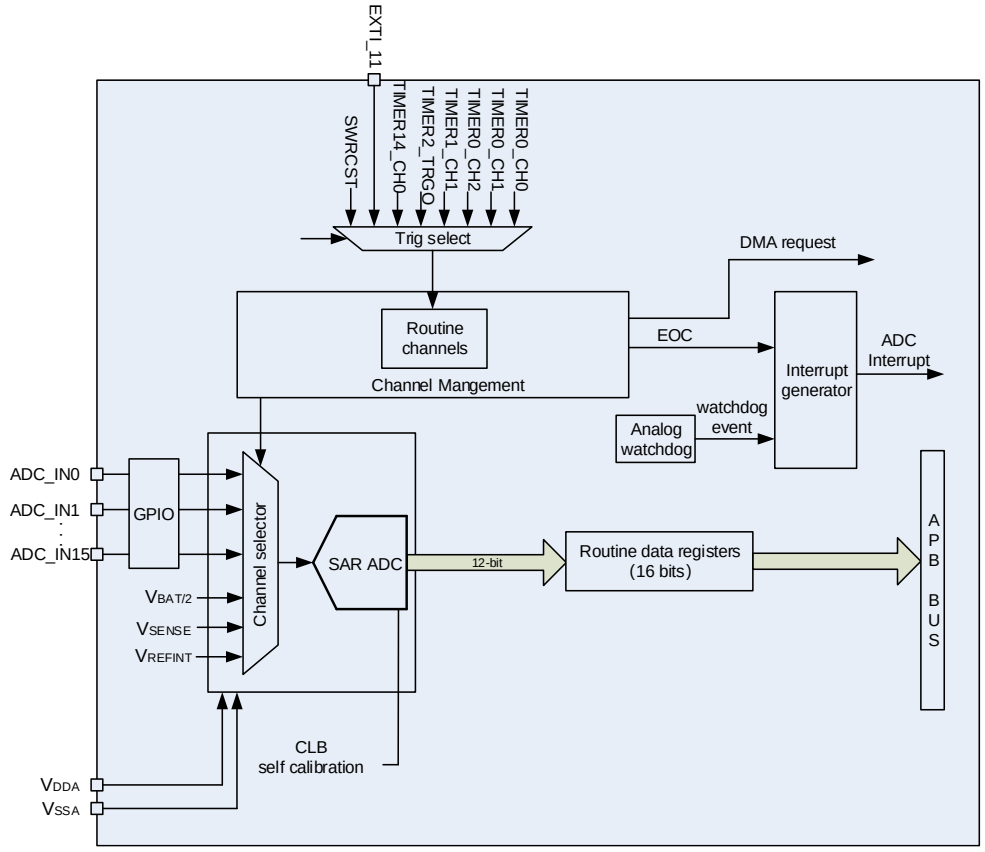
Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{REFINT}	Internal voltage reference output voltage
$V_{BAT}/2$	V_{BAT} pin input voltage divided by 2

Table 10-2. ADC input pins definition

Name	Description
V_{DDA}	Analog power supply equal to V_{DD}
V_{SSA}	Ground for analog power supply equal to V_{SS}
ADCx_IN [15:0]	Up to 16 analog channels

10.4. Function overview

Figure 10-1. ADC module block diagram



10.4.1. 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 can not use the ADC until the calibration is completed. The calibration should be performed before starting A/D conversion. The calibration is initiated by setting the CLB bit to 1. The CLB bit stays 1 during the calibration sequence. Then It is cleared by hardware as soon as the calibration is completed.

When the ADC operating condition changes (supply power voltage V_{DDA} and temperature), 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

10.4.2. Dual clock domain architecture

The ADC sub-module, with exception of the APB interface block, is feed by an ADC clock, which can be asynchronous and independent from the APB clock.

Application can reduce PLCK frequency for low power operation while still keeping optimum ADC performance.

Refer to ***RCU Section 4.2.1*** for more information on generating this clock source.

10.4.3. ADCON enable

The ADC module is enabled or disabled by configuring the ADCON bit in the ADC_CTL1 register. 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 entered power off mode.After ADC is enabled,you need delay t_{us} time for sampling, the value of t_{us} please refer to the chip datasheet.

10.4.4. 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.

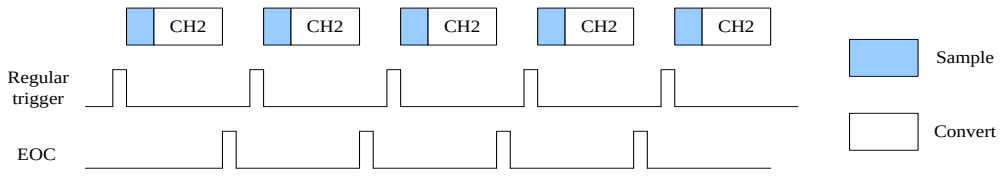
The RL[3:0] bits in the ADC_RSQ0 register specify the total conversion sequence length. The ADC_RSQ0~ADC_RSQ2 registers specify the selected channels of the routine sequence.

10.4.5. Operation modes

Single operation mode

In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits in ADC_RSQ2. When the ADCON is 1, the ADC samples and converts a single channel, once the corresponding software trigger or external trigger is active.

Figure 10-2. 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.

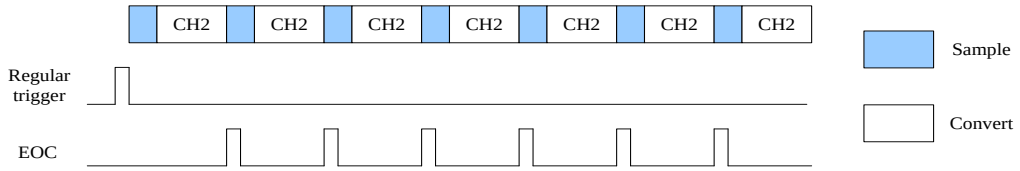
2. Configure RSQ0 with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register if it is needed.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. After a delay of one CK_ADC, read the converted date in the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.

Note: After EOC is set, a delay of one CK_ADC is required before reading the ADC conversion result.

Continuous operation mode

The continuous operation mode will be enabled when the 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 a specified channel, once the corresponding software trigger or external trigger is active. The conversion data will be stored in the ADC_RDATA register.

Figure 10-3. 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 with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register if it is needed.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. After a delay of one CK_ADC, read the converted date 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.

Note: After EOC is set, a delay of one CK_ADC is required before reading the ADC conversion result.

To avoid checking, DMA can be used to transfer the converted data:

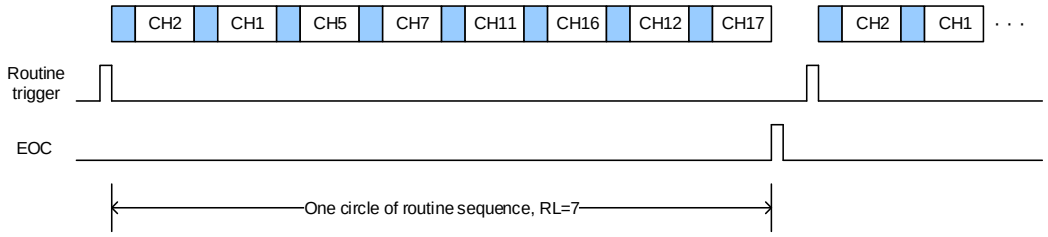
1. Set the CTN and DMA bit in the ADC_CTL1 register.
2. Configure the RSQ0 with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETERC and ETSRC bits in the ADC_CTL1 register in if it is needed.
5. Prepare the DMA module to transfer data from the ADC_RDATA (refer to the spec of the ***Direct memory access controller (DMA)*** module).
6. Set the SWRCST bit, or generate an external 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 all channels with a specific routine 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 routine sequence till the end of sequence, once the corresponding software trigger or external 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.

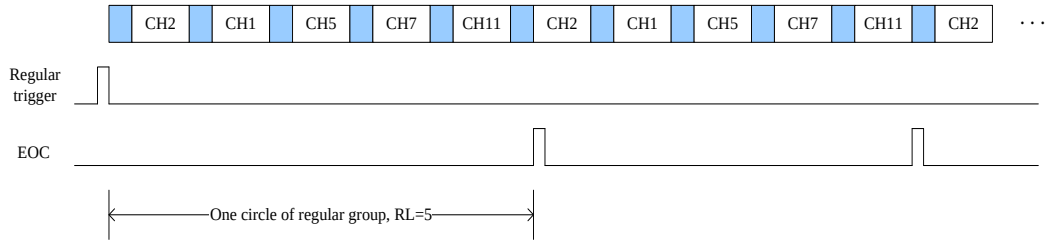
Figure 10-4. Scan operation mode, continuous disable



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 it is needed.
4. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Clear the EOC flag by writing 0.

Figure 10-5. Scan operation mode, continuous enable

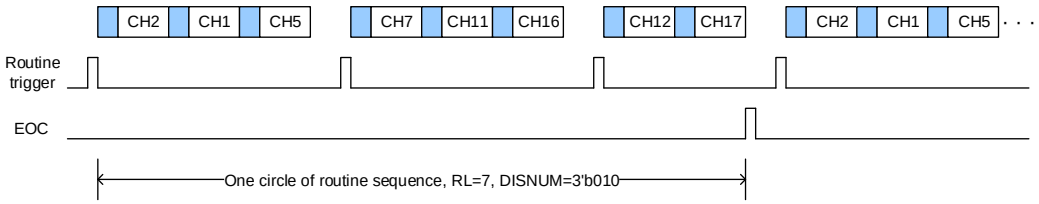


Discontinuous operation mode

The discontinuous operation mode will be enabled when the 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 sequence of 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 external 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 10-6. 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 it is needed.
5. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate an external trigger for the routine sequence.
7. Repeat step6 if it is needed.
8. Wait the EOC flag to be set.
9. Clear the EOC flag by writing 0.

10.4.6. Conversion result threshold monitor function

The analog watchdog is enabled when the RWDEN bit in the ADC_CTL0 register is set for routine sequence. This function is used to monitor whether the conversion result exceeds the set thresholds, and the WDE bit in ADC_STAT register will be set. An interrupt will be generated if the WDEIE bit is set. The ADC_WDHT and ADC_WDLT registers are used to specify the high and low threshold. The comparison is done before the alignment, so the threshold values are independent of the alignment, which is specified by the DAL bit in the ADC_CTL1 register. One or more channels, which are selected by the RWDEN, WDSC and WDCHSEL [4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog.

10.4.7. Data storage mode

The alignment of data stored after conversion can be specified by DAL bit in the ADC_CTL1 register.

When the data is most significant bit alignment, the 12/10/8-bit data are aligned on a half-word, while the 6-bit data is aligned on a byte, which are shown as ***Figure 10-7. Data storage mode of 12-bit resolution***, ***Figure 10-8. Data storage mode of 10-bit resolution***, ***Figure 10-9. Data storage mode of 8-bit resolution*** and ***Figure 10-10. Data storage mode of 6-bit resolution***.

Figure 10-7. Data storage mode of 12-bit resolution

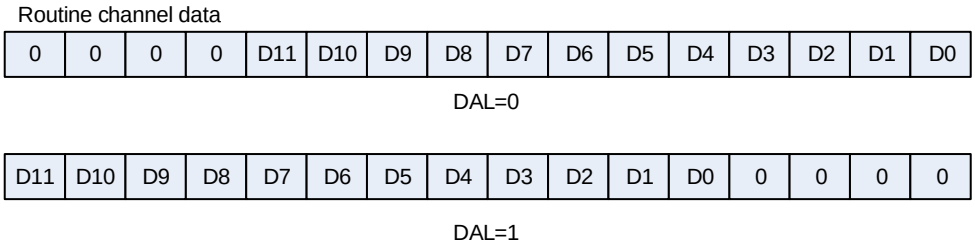


Figure 10-8. Data storage mode of 10-bit resolution

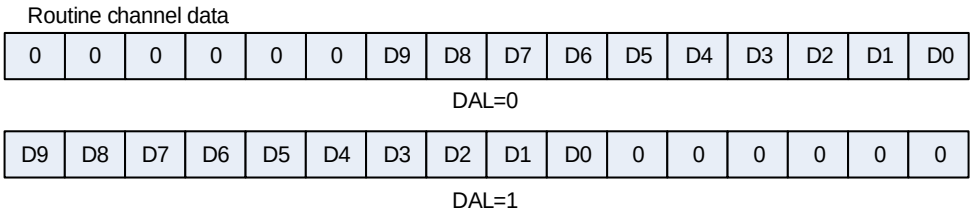


Figure 10-9. Data storage mode of 8-bit resolution

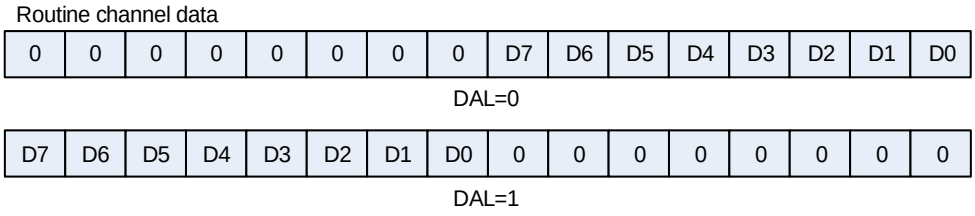
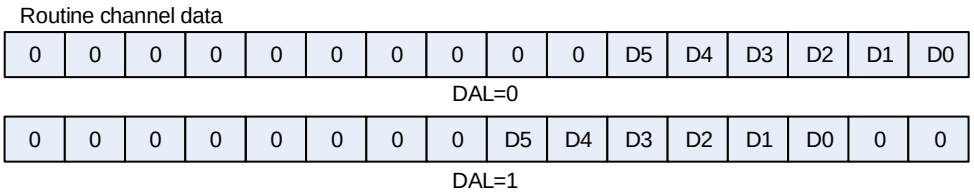


Figure 10-10. Data storage mode of 6-bit resolution



10.4.8. Sample time configuration

The number of ADC_CLK 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. And each channel can specify different sample time. For the 12-bit resolution, the total sampling and conversion time is “sampling time + 12.5” ADC_CLK cycles.

Example:

CK_ADC = 14MHz and sample time is 1.5 cycles, the total conversion time is “1.5+12.5” CK_ADC cycles, that means 1us.

10.4.9. External trigger configuration

The conversion of routine sequence can be triggered by rising edge of external trigger inputs. The external trigger source of routine sequence is controlled by the ETSRC [2:0] bits in the ADC_CTL1 register.

Table 10-3. External trigger source for ADC routine sequence

ETSRC[2:0]	Trigger Source	Trigger Type
000	TIMER0_CH0	Hardware trigger
001	TIMER0_CH1	
010	TIMER0_CH2	
011	TIMER1_CH1	
100	TIMER2_TRGO	
101	TIMER14_CH0	
110	EXTI_11	
111	SWRCST	Software trigger

10.4.10. DMA request

The DMA request, which is enabled by the DMA bit in 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 which is specified by the user.

10.4.11. ADC internal channels

When the TSVREN bit in ADC_CTL1 register is set, the temperature sensor channel (ADC_IN16) and V_{REFINT} channel (ADC_IN17) is enabled. The temperature sensor used to measure the ambient temperature outputs voltage which 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. When this sensor is not in use, it can be put in power down mode by resetting the TSVREN bit.

The output voltage of the temperature sensor changes linearly with temperature. Though there is an offset, which is up to 45°C and varies from chip to chip due to the chip production process variation, the internal temperature sensor is more appropriate to detect temperature variations than absolute temperatures. It is recommended to use an external temperature sensor to dectect accurate temperature.

The internal voltage reference (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and Comparators. V_{REFINT} is internally connected to the ADC_IN17 input channel.

To use the temperature sensor:

1. Configure the conversion sequence (ADC_IN16) and the sampling time (t_{S_temp} μs) for the channel.

2. Enable the temperature sensor by setting the TSVREN bit in the ADC control register 1 (ADC_CTL1).
3. Start the ADC conversion by setting the ADCON bit or by the triggers.
4. Read the internal temperature sensor output voltage($V_{temperature}$), and get the temperature with the following equation:

$$\text{Temperature (°C)} = \{(V_{25} - V_{temperature}) / \text{Avg_Slope}\} + 25.$$

V_{25}: internal temperature sensor output voltage at 25°C, 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.

10.4.12. Battery voltage monitoring

The V_{BAT} channel can be used to measure the backup battery voltage on the V_{BAT} pin. When the VBATEN bit in ADC_CTL1 register is set, V_{BAT} channel (ADC_IN18) is enabled and a bridge divider by 2 integrated on the V_{BAT} pin is also enabled automatically with it. As V_{BAT} may be higher than V_{DDA}, this bridge is used to ensure the ADC correct operation. And it connects $V_{BAT}/2$ to the ADC_IN18 input channel. So, the converted digital value is $V_{BAT}/2$. In order to prevent unnecessary battery energy consumption, it is recommended that the bridge will be enabled only when it is required.

10.4.13. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for routine sequence
- The analog watchdog event (the analog watchdog status bit is set)

Separate interrupt enable bits are available for flexibility.

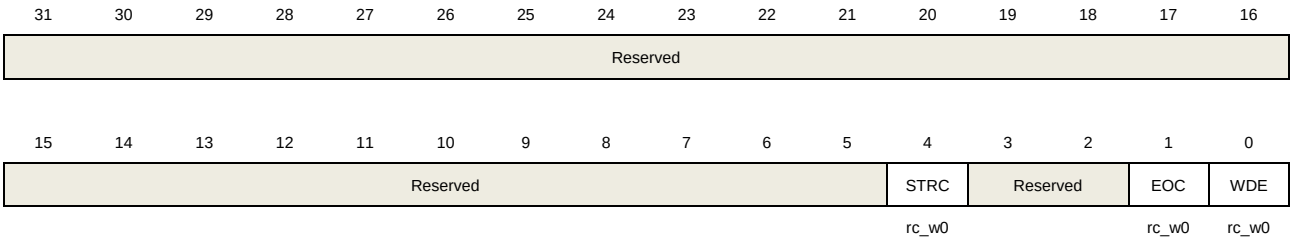
10.5. Register definition

ADC base address: 0x4001 2400

10.5.1. Status register (ADC_STAT)

Address offset: 0x00
Reset value: 0x0000 0000

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

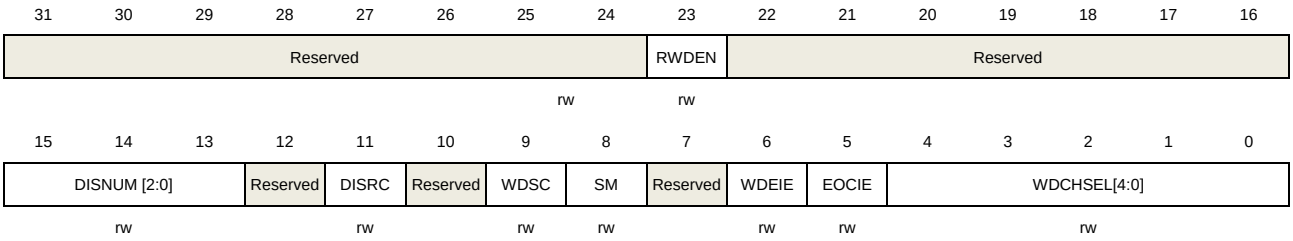


Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	STRC	Start flag of routine sequence conversion 0: Conversion is not started 1: Conversion is started Set by hardware when routine sequence conversion starts. Cleared by software writing 0 to it.
3:2	Reserved	Must be kept at reset value.
1	EOC	End flag of routine sequence conversion 0: No end of routine sequence conversion 1: End of routine sequence conversion Set by hardware at the end of a routine sequence conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.
0	WDE	Analog watchdog event flag 0: No analog watchdog event 1: Analog watchdog event Set by hardware when the converted voltage crosses the values programmed in the ADC_WDLT and ADC_WDHT registers. Cleared by software writing 0 to it.

10.5.2. 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
31:24	Reserved	Must be kept at reset value.
23	RWDEN	Routine channel analog watchdog enable 0: Analog watchdog routine channel disable 1: Analog watchdog routine channel enable
22:16	Reserved	Must be kept at reset value.
15:13	DISNUM [2:0]	Number of conversions in discontinuous mode The number of channels to be converted after a trigger will be DISNUM [2:0] +1 in routine sequence.
12	Reserved	Must be kept at reset value.
11	DISRC	Discontinuous mode on routine sequence 0: Discontinuous operation mode disable 1: Discontinuous operation mode enable
10	Reserved	Must be kept at reset value.
9	WDSC	When in scan mode, analog watchdog is effective on a single channel 0: All channels have analog watchdog function 1: A single channel has analog watchdog function
8	SM	Scan mode 0: Scan operation mode disable 1: Scan operation mode enable
7	Reserved	Must be kept at reset value.
6	WDEIE	Interrupt enable for WDE 0: Interrupt disable 1: Interrupt enable
5	EOCIE	Interrupt enable for EOC 0: Interrupt disable 1: Interrupt enable
4:0	WDCHSEL [4:0]	Analog watchdog channel select 00000: ADC channel0 00001: ADC channel1 00010: ADC channel2

……

01111: ADC channel15

10000: ADC channel16

10001: ADC channel17

10010: ADC channel18

Other values are reserved.

Note: ADC analog inputs Channel16, Channel17 and Channel 18 are internally connected to the temperature sensor, to V_{REFINT} and to V_{BAT} analog inputs.

10.5.3. Control register 1 (ADC_CTL1)

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							VBATEN	TSVREN	SWRCST	Reserved	ETERC	ETSRC[2:0]			Reserved
							rw	rw	rw	rw	rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DAL	Reserved		DMA	Reserved				RSTCLB	CLB	CTN	ADCON
				rw			rw					rw	rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value
24	VBATEN	This bit is set/cleared by software to enable/disable the V_{BAT} channel. 0: V_{BAT} channel disabled 1: V_{BAT} channel enabled
23	TSVREN	Channel 16 and 17 enable of ADC. 0: Channel 16 and 17 of ADC disable 1: Channel 16 and 17 of ADC enable
22	SWRCST	Software start conversion of routine sequence. Set 1 on this bit starts a conversion of of a routine channel if ETSRC is 111. 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
19:17	ETSRC[2:0]	External trigger select for routine sequence 000: TIMER0 CH0

		001: TIMER0 CH1 010: TIMER0 CH2 011: TIMER1 CH1 100: TIMER2 TRGO 101: TIMER14 CH0 110: EXTI line 11 111: 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
7:4	Reserved	Must be kept at reset value
3	RSTCLB	Reset calibration This bit is set by software and cleared by hardware after the calibration registers are initialized. 0: Calibration register initialize done. 1: Initialize calibration register 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. When this bit is high and “1” is written to it with other bits of this register unchanged, the conversion will start. 0: ADC disable and power down 1: ADC enable

10.5.4. Sampling 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 sampling time 000: channel sampling time is 1.5 cycles 001: channel sampling time is 7.5 cycles 010: channel sampling time is 13.5 cycles 011: channel sampling time is 28.5 cycles 100: channel sampling time is 41.5 cycles 101: channel sampling time is 55.5 cycles 110: channel sampling time is 71.5 cycles 111: channel sampling time is 239.5 cycles **Note:** The channel 0 and channel 18 sampling time are specified by the SPT0[2:0] bits in the ADC_SAMPT1.

10.5.5. Sampling time register 1 (ADC_SAMPT1)

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		SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]	SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]		

rw	rw	rw	rw	rw	rw

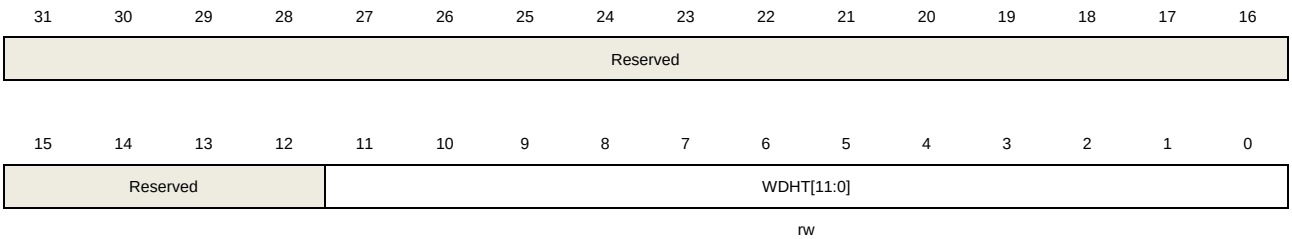
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29:27	SPT9[2:0]	Refer to SPT0[2:0] description
26:24	SPT8[2:0]	Refer to SPT0[2:0] description
23:21	SPT7[2:0]	Refer to SPT0[2:0] description
20:18	SPT6[2:0]	Refer to SPT0[2:0] description
17:15	SPT5[2:0]	Refer to SPT0[2:0] description
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 sampling time 000: channel sampling time is 1.5 cycles 001: channel sampling time is 7.5 cycles 010: channel sampling time is 13.5 cycles 011: channel sampling time is 28.5 cycles 100: channel sampling time is 41.5 cycles 101: channel sampling time is 55.5 cycles 110: channel sampling time is 71.5 cycles 111: channel sampling time is 239.5 cycles **Note:** The channel 0 and channel 18 sampling time are specified by the SPT0[2:0] bits in the ADC_SAMPT1.

10.5.6. Watchdog high threshold register (ADC_WDHT)

Address offset: 0x24

Reset value: 0x0000 0FFF

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

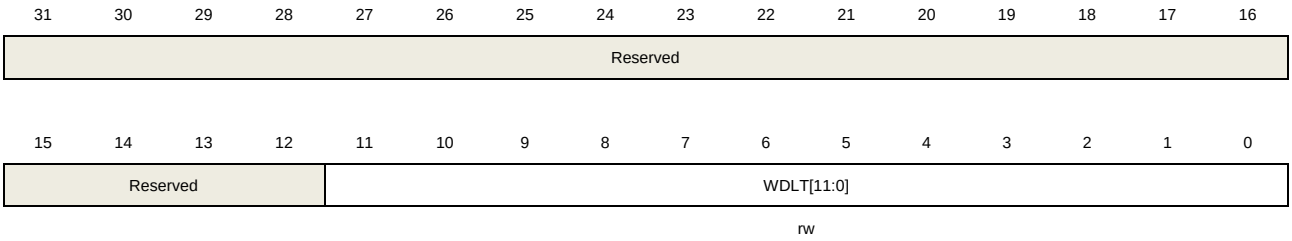


Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	WDHT[11:0]	High threshold for analog watchdog These bits define the high threshold for the analog watchdog.

10.5.7. Watchdog low threshold register (ADC_WDLT)

Address offset: 0x28
Reset value: 0x0000 0000

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

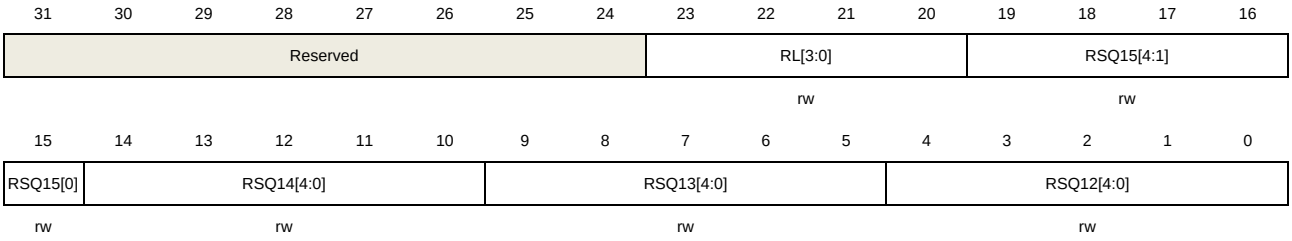


Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value
11:0	WDLT[11:0]	Low threshold for analog watchdog These bits define the low threshold for the analog watchdog.

10.5.8. Routine sequence register 0 (ADC_RSQ0)

Address offset: 0x2C
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
23:20	RL[3:0]	Routine sequence length The total number of conversion in routine sequence equals to RL [3:0] +1.

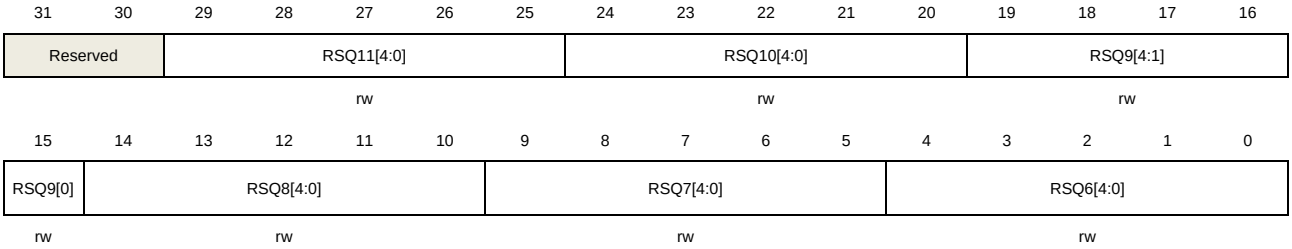
19:15	RSQ15[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ14[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ13[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ12[4:0]	Refer to RSQ0[4:0] description

10.5.9. Routine sequence register 1 (ADC_RSQ1)

Address offset: 0x30

Reset value: 0x0000 0000

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



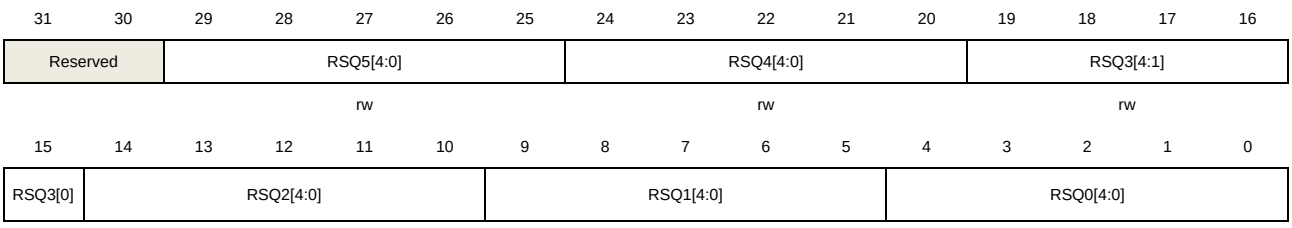
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value
29:25	RSQ11[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ10[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ9[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ8[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ7[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ6[4:0]	Refer to RSQ0[4:0] description

10.5.10. Routine sequence register 2 (ADC_RSQ2)

Address offset: 0x34

Reset value: 0x0000 0000

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



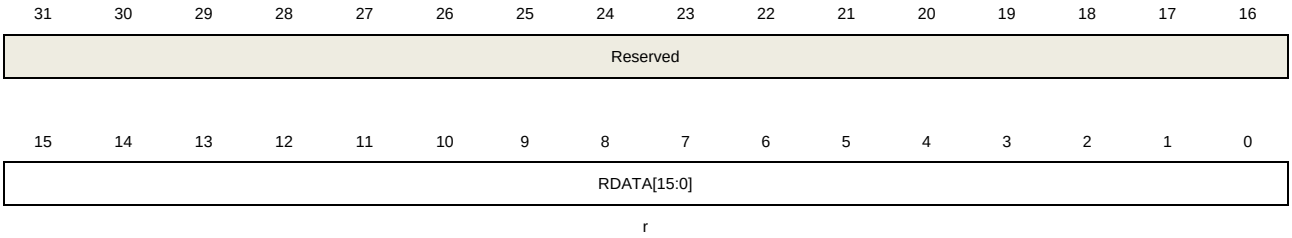
rw rw rw rw

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..18) are written to these bits to select a channel at the nth conversion in the routine sequence.

10.5.11. 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	Reserved	Must be kept at reset value
15:0	RDATA[15:0]	Routine channel data These bits contain the conversion result from routine channel, which is read only.

11. Digital-to-analog converter (DAC)

11.1. 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.

11.2. 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_{DDA}.

Figure 11-1. DAC block diagram and *Table 11-1. DAC I/O description* show the block diagram of DAC and the pin description of DAC, respectively.

Figure 11-1. DAC block diagram

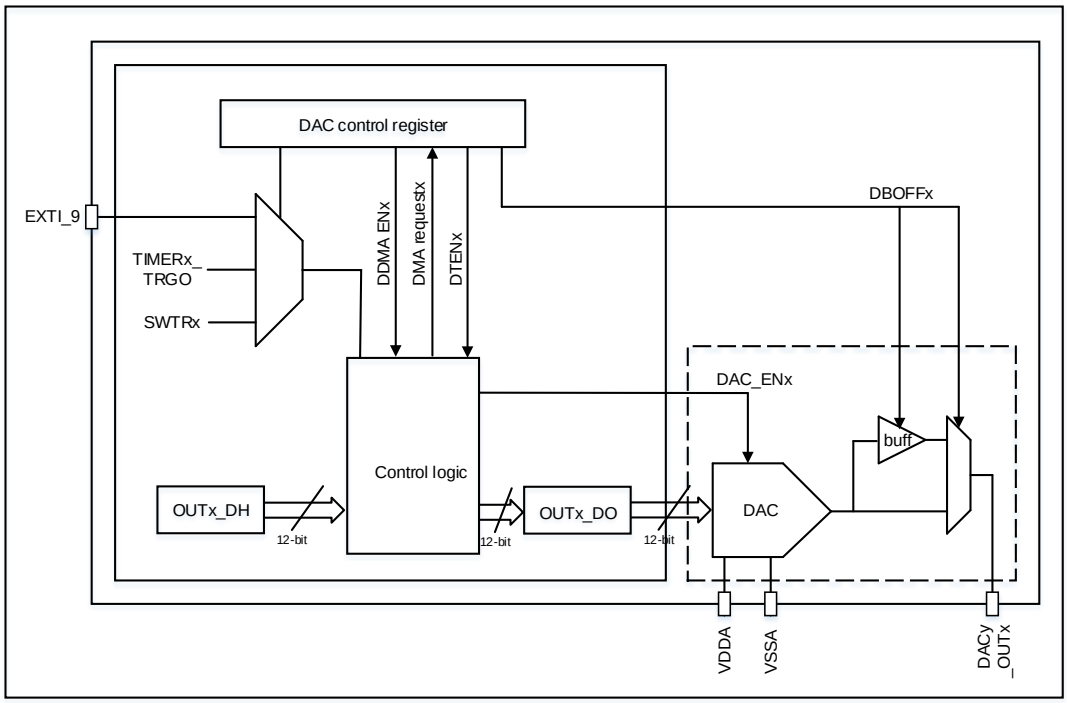


Table 11-1. DAC I/O description

Name	Description	Signal type
V_{DDA}	Analog power supply	Input, analog supply
V_{SSA}	Ground for analog power supply	Input, analog supply ground
DACy_OUTx	DAC analog output	Analog output signal

The below table details the triggers and outputs of the DAC.

Table 11-2. DAC triggers and outputs summary

	DAC0
Channel	**Channel0**
DAC outputs connected to I / Os	PA4
DAC output buffer	•
DAC software trigger	•
DAC trigger signals from EXTI	EXTI_9
DAC trigger signals from TIMER	TIMER1_TRGO TIMER2_TRGO TIMER5_TRGO TIMER14_TRGO

Note: The GPIO pins should be configured to analog mode before enable the DAC module.

11.3. Function description

11.3.1. 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.

11.3.2. 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.

11.3.3. DAC data configuration

The 12-bit DAC holding data (OUTx_DH) can be configured by writing any one of the OUTx_R12DH, OUTx_L12DH and OUTx_R8DH registers. When the data is loaded by OUTx_R8DH register, only the MSB 8 bits are configurable, the LSB 4 bits are forced to 4'b0000.

11.3.4. 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 bits in the DAC_CTL0 register. The DAC external triggers are selected by the DTSELx bits in the DAC_CTL0 register, which is

shown as *Table 11-3. Triggers of DAC*.

Table 11-3. Triggers of DAC

DTSELx[2:0]	Trigger Source	Trigger Type
3b'000	TIMER5_TRGO	Hardware trigger
3b'001	TIMER2_TRGO	
3b'010	Reserved	
3b'011	TIMER14_TRGO	
3b'100	TIMER1_TRGO	
3b'101	Reserved	
3b'110	EXTI_9	
3b'111	SWTR	Software trigger

The TIMERx_TRGO signals are generated from the timers, while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

11.3.5. 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 (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 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.

11.3.6. DAC output voltage

The following equation determines the analog output voltage on the DAC pin.

$$V_{DAC_OUT}=V_{DDA}*OUTx_DO/4096 \quad (1\text{-}1)$$

The digital input is linearly converted to an analog output voltage, its range is 0 to V_{DDA}.

11.3.7. 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.

11.4. DAC register

DAC0 base address: 0x4000 7400

11.4.1. 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		DDUDRIE0	DDMAEN0	Reserved						DTSEL0[2:0]			DTEN0	DBOFF0	DEN0
		rw	rw								rw		rw	rw	rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value
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
11:6	Reserved	Must be kept at reset value
5:3	DTSEL0[2:0]	DACx_OUT0 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 000: TIMER5 TRGO 001: TIMER2 TRGO 010: Reserved 011: TIMER14 TRGO 100: TIMER1 TRGO 101: Reserved 110: EXTI line 9 111: Software trigger
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

11.4.2. DACx software trigger register (DAC_SWT)

Address offset: 0x04

Reset value: 0x0000 0000

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

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															SWTR0
															w

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

11.4.3. 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).

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				OUT0_DH[11:0]											
				rw											

Bits	Fields	Descriptions

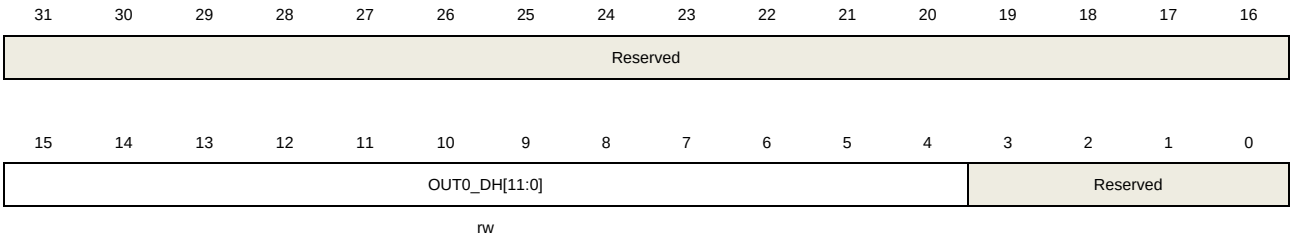
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT0.

11.4.4. 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).



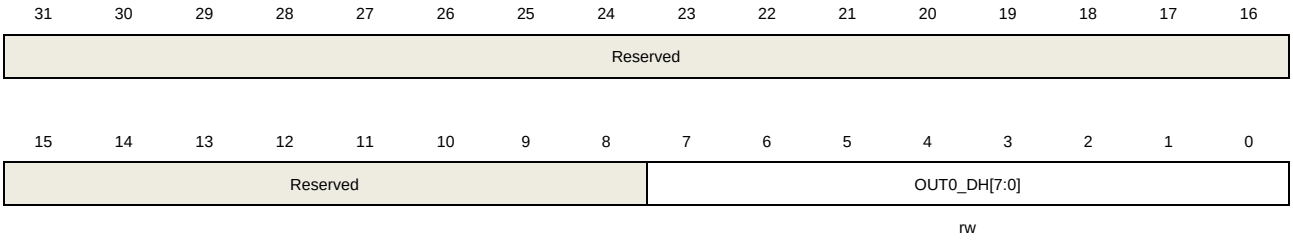
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

11.4.5. 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).



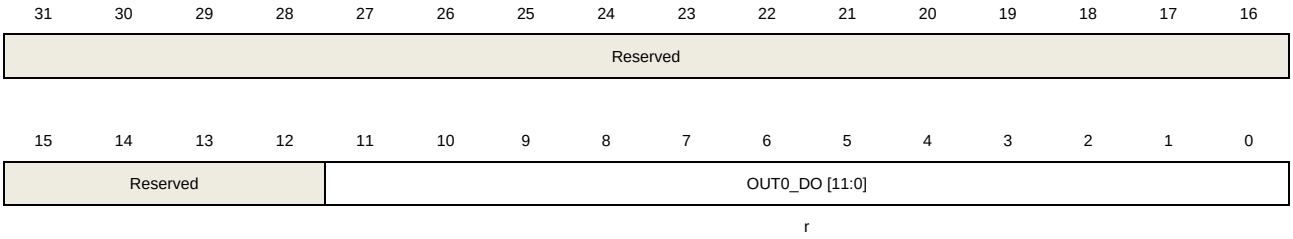
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

11.4.6. DACx_OUT0 data output register (DAC_OUT0_DO)

Address offset: 0x2C

Reset value: 0x0000 0000

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



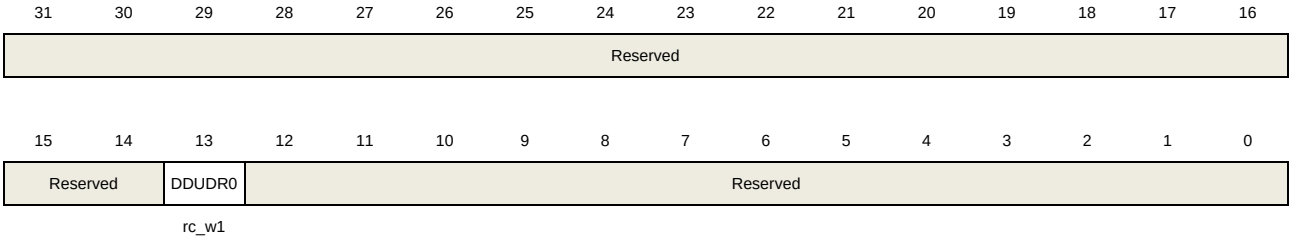
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DO [11:0]	DACx_OUT0 12-bit output data These bits, which are read only, storage the data that is being converted by DACx_OUT0.

11.4.7. DACx status register 0 (DAC_STAT0)

Address offset: 0x34

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.

12. Comparator (CMP)

12.1. 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.

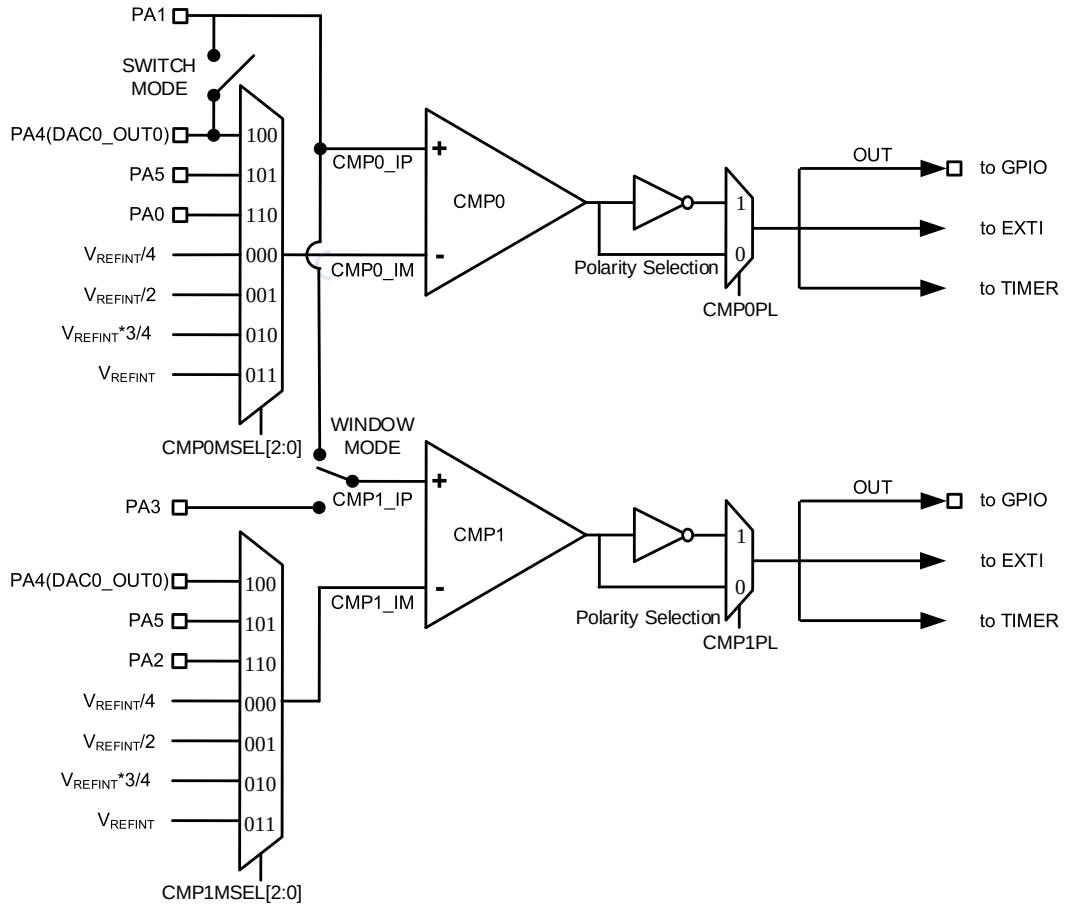
12.2. Characteristics

- 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.
- Window comparator.
- Outputs to I / O.
- Outputs to timers for triggering.
- Outputs to EXTI.

12.3. Function overview

The block diagram of CMP is shown below:

Figure 12-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

12.3.1. CMP clock

The clock of the CMP which is connected to APB bus, is synchronous with PCLK. It shares the common reset and clock enable bits with SYSCFG.

12.3.2. CMP I / O configuration

These I / Os must be configured in analog mode in the GPIOs registers before they are selected as CMP inputs.

Considering pin definitions in datasheet, and the CMP output must be connected to corresponding alternate I / Os.

The CMP output can be redirected internally and externally simultaneously.

The CMP output internally connect to the TIMER and the connections between them are as follows:

- CMP output to the TIMER input channel.
- CMP output to the TIMER break.
- CMP output to the TIMER OCPRE_CLR.

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 12-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 12-1. CMP inputs and outputs summary

	CMP0	CMP1
CMP non inverting inputs connected to I / Os	PA1 PA4	PA1 PA3
CMP inverting inputs connected to I / Os	PA0 PA4 PA5	PA2 PA4 PA5
CMP inverting inputs connected to internal signals	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0
CMP outputs connected to I / Os	PA0 PA6 PA11	PA2 PA7 PA12
CMP outputs connected to EXTI	•	
CMP outputs connected to internal signals	TIMER0_CH0 TIMER1_CH3 TIMER2_CH0 TIMER0 OCPRE_CLR TIMER1 OCPRE_CLR TIMER2 OCPRE_CLR	TIMER0_CH0 TIMER1_CH3 TIMER2_CH0 TIMER0 OCPRE_CLR TIMER1 OCPRE_CLR TIMER2 OCPRE_CLR
CMP outputs(motor control protection)	TIMER0 BRKIN	

12.3.3. 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 CMP_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.

12.3.4. CMP windows mode

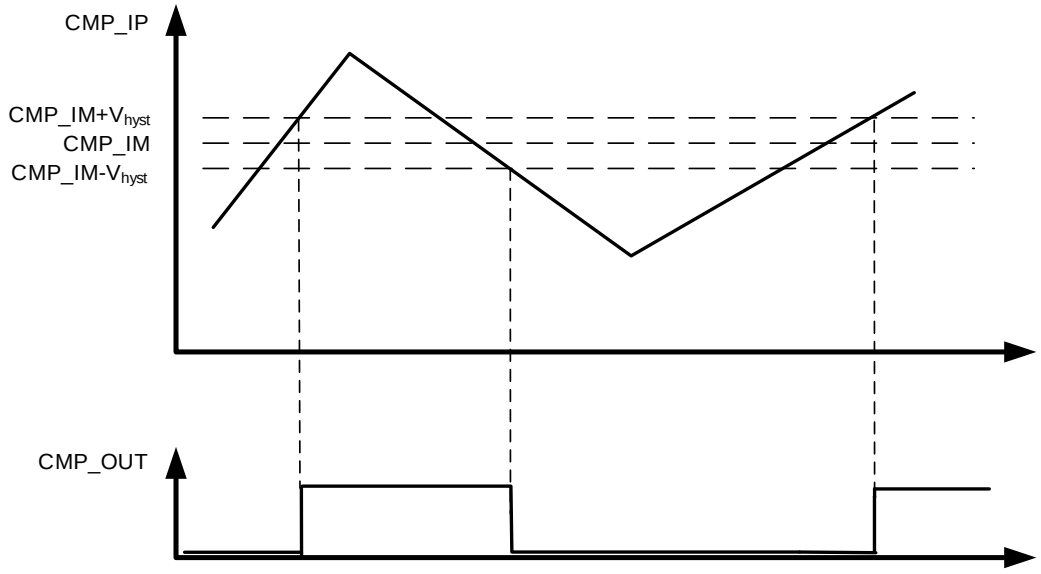
If the WNDEN bit in CMP_CS register is set, comparator windows mode is enabled, Input

plus of comparator 1 is connected with input plus of comparator 0. If the minus input of CMP0 and CMP1 is connected to different voltage, the voltage range from lower threshold to upper threshold, is monitored by analyzing the comparator 0 and comparator 1 output.

12.3.5. 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 CMP_CS register. This function could be shut down if it is unnecessary.

Figure 12-2. CMP hysteresis



12.3.6. CMP register write protection

The CMP control and status register (CMP_CS) can be protected from writing by setting CMPxLK bit to 1, and only be reset by the MCU reset. The CMP_CS[31:16] bits will be read-only if CMP1LK bit is set , and the CMP_CS[15:0] bits will be read-only if CMP0LK bit is set.

12.3.7. 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.

12.4. Register definition

CMP base address: 0x4001 001C

12.4.1. CMP Control/status register (CMP_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
CMP1LK	CMP1O	CMP1HST[1:0]		CMP1PL	CMP1OSEL[2:0]			WNDEN	CMP1MSEL[2:0]			CMP1M[1:0]		Reserved	CMP1EN
rwo	r	rw		rw	rw			rw	rw			rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0LK	CMP0O	CMP0HST[1:0]		CMP0PL	CMP0OSEL[2:0]			Reserved	CMP0MSEL[2:0]			CMP0M[1:0]		CMP0SW	CMP0EN
rwo	r	rw		rw	rw				rw			rw		rw	rw

Bits	Fields	Descriptions
31	CMP1LK	CMP1 lock This bit can set all control bits of CMP1 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP_CS[31:16] bits are read-write 1: CMP_CS[31:16] bits are read-only
30	CMP1O	CMP1 output state This bit is a copy of CMP1 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
29:28	CMP1HST[1:0]	CMP1 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
27	CMP1PL	Polarity of CMP1 output This bit is used to select the polarity of CMP1 output. 0: Output is not inverted 1: Output is inverted
26:24	CMP1OSEL[2:0]	CMP1 output selection These bits are used to select the destination of the CMP1 output. 000: No selection 001: TIMER0 break input 010: TIMER0 CH0 input capture

		011: TIMER0 OCPRE_CLR input 100: TIMER1 CH3 input capture 101: TIMER1 OCPRE_CLR input 110: TIMER2 CH0 input capture 111: TIMER2 OCPRE_CLR input **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.
23	WNDEN	Window mode enable This bit is used to select CMP1_IP source. 0: CMP1_IP is connected to CMP1 non-inverting input 1: CMP1_IP is connected to CMP0_IP
22:20	CMP1MSEL[2:0]	CMP1_IM input selection These bits are used to select the source connected to the CMP1_IM input of the CMP1. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT} * 3 / 4 011: V_{REFINT} 100: PA4 (DAC0_OUT0) 101: PA5 110: PA2 111: Reserved
19:18	CMP1M[1:0]	CMP1 mode These bits are used to control the operating mode of the CMP1 to adjust the speed /consumption. 00: High speed / full power 01: Medium speed / medium power 10: Low speed / low power 11: Very-low speed / ultra-low power
17	Reserved	Must be kept at reset value.
16	CMP1EN	CMP1 enable 0: CMP1 disabled 1: CMP1 enabled
15	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: CMP_CS[15:0] bits are read-write 1: CMP_CS[15:0] bits are read-only
14	CMP0O	CMP0 output state This bit is a copy of CMP0 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
13:12	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
11	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
10:8	CMP0OSEL[2:0]	CMP0 output selection These bits are used to select the destination of the CMP0 output. 000: No selection 001: TIMER0 break input 010: TIMER0 CH0 input capture 011: TIMER0 OCPRE_CLR input 100: TIMER1 CH3 input capture 101: TIMER1 OCPRE_CLR input 110: TIMER2 CH0 input capture 111: TIMER2 OCPRE_CLR input **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.
7	Reserved	Must be kept at reset value.
6:4	CMP0MSEL[2:0]	CMP0_IM input selection These bits are used to select the source connected to the CMP0_IM input of the CMP0. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT} * 3 / 4 011: V_{REFINT} 100: PA4 (DAC0_OUT0) 101: PA5 110: PA0 111: Reserved
3:2	CMP0M[1:0]	CMP0 mode These bits are used to control the operating mode. 00: High speed / full power 01: Medium speed / medium power 10: Low speed / low power 11: Very-low speed / ultra-low power

1	CMP0SW	CMP0 switch This bit is used to close a switch between CMP0 non-inverting input on PA1 and PA4 I / O. 0: Switch open 1: Switch closed
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

13. 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 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.

13.1. Free watchdog timer (FWDGT)

13.1.1. 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 causes a reset when the internal down counter reaches 0. The register write protection function in free watchdog can be enabled to prevent it from changing the configuration unexpectedly.

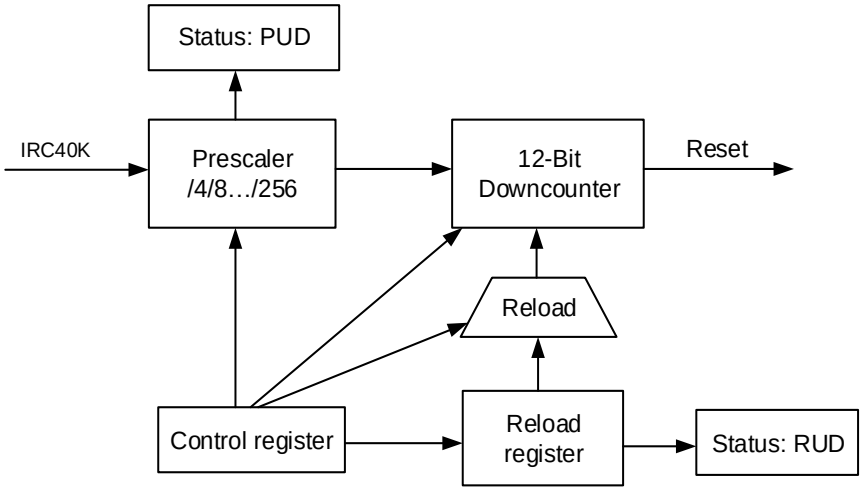
13.1.2. Characteristics

- Free-running 12-bit downcounter.
- Reset when the downcounter reaches 0, if the watchdog is enabled.
- Free clock source，FWDGT can operate even if the main clock fails such as in standby and Deep-sleep modes.
- Hardware free watchdog timer bit, automatically start the FWDGT at power on.
- FWDGT debug mode, the FWDGT can stop or continue to work in debug mode.

13.1.3. Function overview

The free watchdog timer consists of an 8-stage prescaler and a 12-bit down-counter. Refer to the ***Figure 13-1. Free watchdog timer block diagram*** for the functional blocks of the free watchdog timer module.

Figure 13-1. Free watchdog timer block diagram



The free watchdog timer is enabled by writing the value 0xCCCC in the control register (FWDGT_CTL), and the counter starts counting down. When the counter reaches the value 0x000, a reset is generated.

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 timer can automatically start at power on when the hardware free watchdog timer 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 in the control register. When an update of the prescaler (FWDGT_PSC) or watchdog counter window value (FWDGT_WND) or the reload value (FWDGT_RLD) is on going, the status bit in FWDGT_STAT register is set.

If the FWDGT_HOLD bit in DBG module is cleared, the FWDGT continues to work even the Cortex®-M3 core halted (Debug mode). While the FWDGT stops in Debug mode if the FWDGT_HOLD bit is set.

Table 13-1. Min / max FWDGT timeout period at 40 kHz (IRC40K)

Prescaler divider	PSC[2:0] bits	Min timeout (ms) RL[11:0]= 0x000	Max timeout (ms) RL[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: 1.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.
2. There must be an interval of 7 or more IRC40K clock cycles between two consecutive watchdog reload operations.

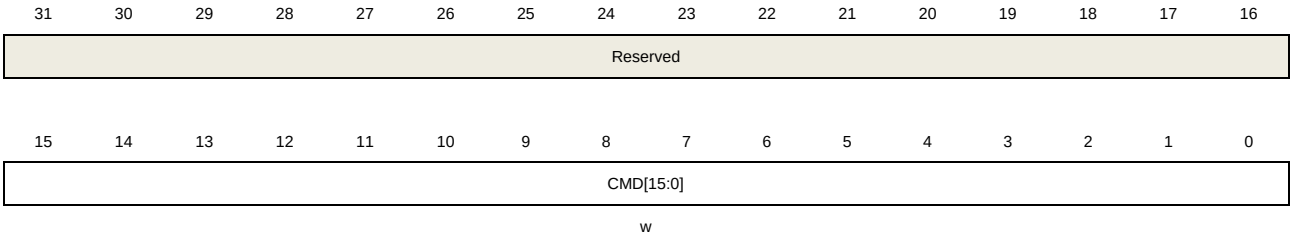
13.1.4. Register definition

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).

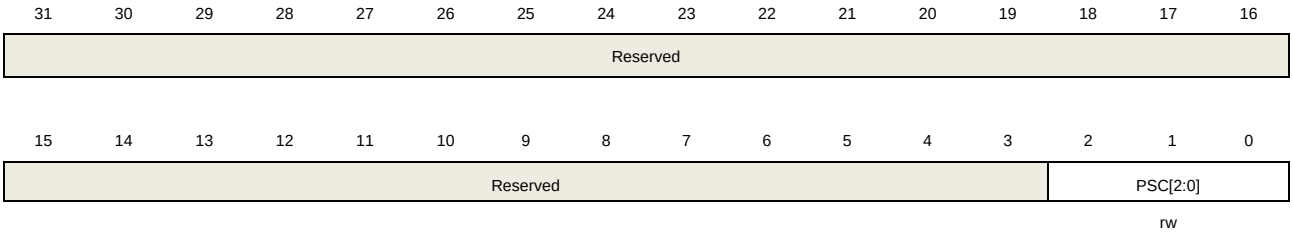


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. These bits have different fuctions when writing different values 0x5555: Disable the FWDGT_PSC, FWDGT_RLD and FWDGT_WND write protection. 0xCCCC: Start the free watchdog timer counter. When the counter reduces to 0, the free watchdog timer 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).



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. When a write operation to this register ongoing, 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

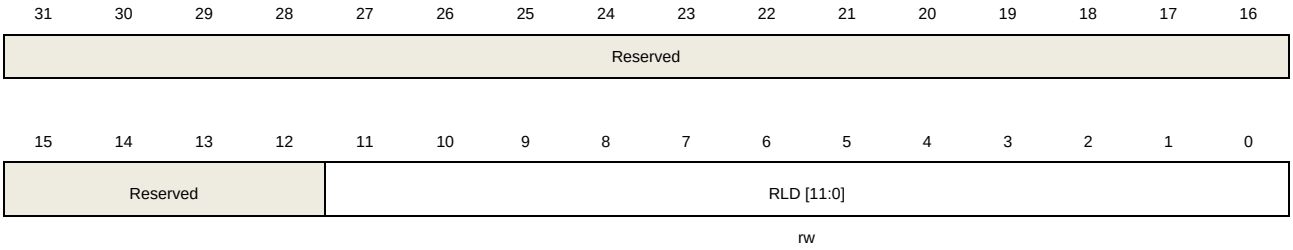
If several prescaler values are used by the application, it is mandatory to wait until PUD bit is reset before changing the prescaler value. However, after updating the prescaler value it is not necessary to wait until PUD is reset before continuing code execution, (for GD32F130xx and GD32F150xx, before 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).



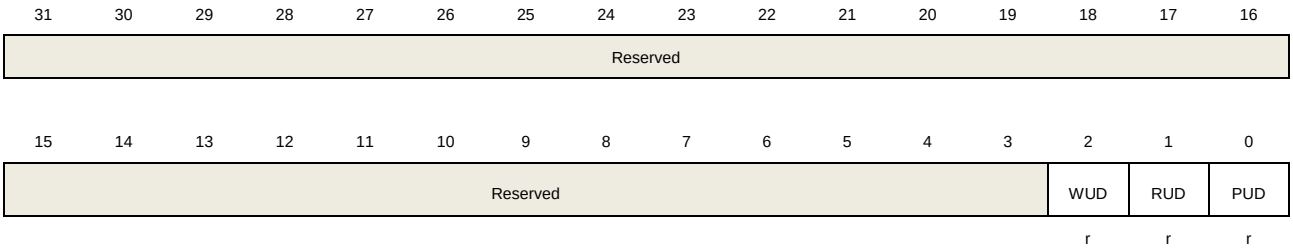
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT conter. These bits are write-protected. Write 0X5555 in the FWDGT_CTL register before writing these bits. When a write operation to this register ongoing, 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 is reset before changing the reload value. However, after updating the reload value it is not necessary to wait until RUD is reset before continuing code execution (for GD32F130xx and GD32F150xx, 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).



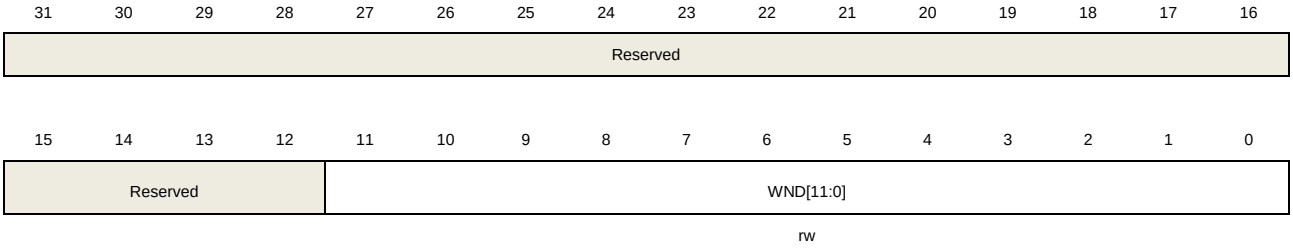
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
2	WUD	Watchdog counter window value update When a write operation to FWDGT_WND register ongoing, this bit is set and the value read from FWDGT_WND register is invalid.
1	RUD	Free watchdog timer counter reload value update When a write operation to FWDGT_RLD register ongoing, this bit is set and the value read from FWDGT_RLD register is invalid. This bit is reset by hardware after the update operation of FWDGT_RLD register.
0	PUD	Free watchdog timer prescaler value update When a write operation to FWDGT_PSC register ongoing, this bit is set and the value read from FWDGT_PSC register is invalid. This bit is reset by hardware after the update operation of FWDGT_PSC register.

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).



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 is 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.

13.2. Window watchdog timer (WWDGT)

13.2.1. 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 downcounter reduce progressively. The watchdog timer causes a reset when the counter reached 0x3F (the CNT[6] bit becomes cleared). The watchdog timer also cause a reset if 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.

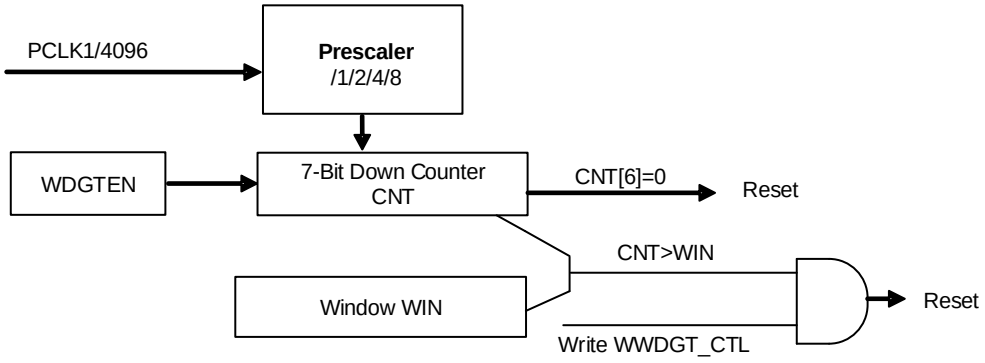
13.2.2. Characteristics

- Programmable free-running 7-bit downcounter.
- 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): if the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x40 or refreshes before it reaches the window value.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

13.2.3. Function overview

If the window watchdog timer is enable (set the WDGTEN bit in the WWDGT_CTL), the watchdog timer cause a reset when the counter reached 0x3F (the CNT[6] bit becomes cleared), or when the counter is refreshed before the counter reached the window register value.

Figure 13-2. Window watchdog timer block diagram



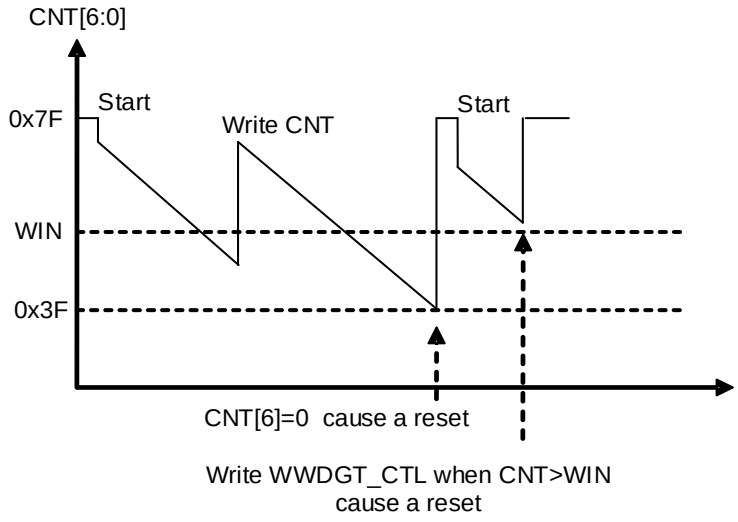
The window watchdog timer is always disabled after power on reset. The software starts the watchdog by setting the WDGTEN bit in the WWDGT_CTL register. Whenever window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F, it implies that the CNT[6] bit should be set. The CNT[5:0] determine the maximum time interval of 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 downcounter when 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 flag is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 13-3. Window watchdog timer timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (13\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB1 clock period measured in ms

Refer to the ***Table 13-2. Min-max timeout value at 36 MHz (fPCLK1)***for the minimum and maximum values of the t_{WWDGT}.

Table 13-2. Min-max timeout value at 36 MHz (fPCLK1)

Prescaler divider	PSC[1:0]	Min timeout value	Max timeout value

		CNT[6:0] =0x40	CNT[6:0]=0x7F
1 / 1	00	113 μs	7.28 ms
1 / 2	01	227 μs	14.56 ms
1 / 4	10	455 μs	29.12 ms
1 / 8	11	910 μs	58.25 ms

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M3 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

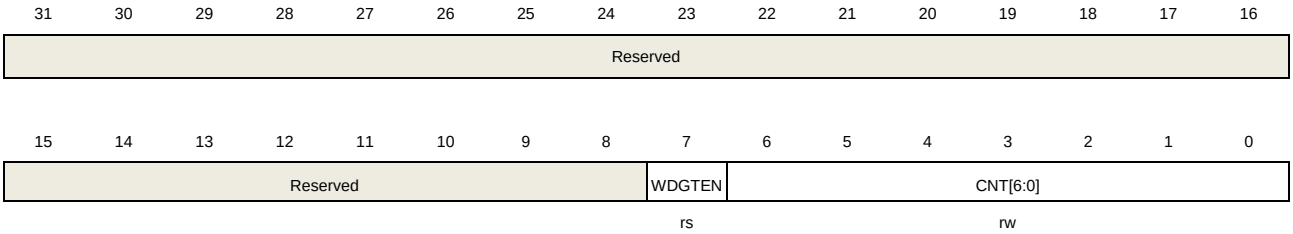
13.2.4. Register definition

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).

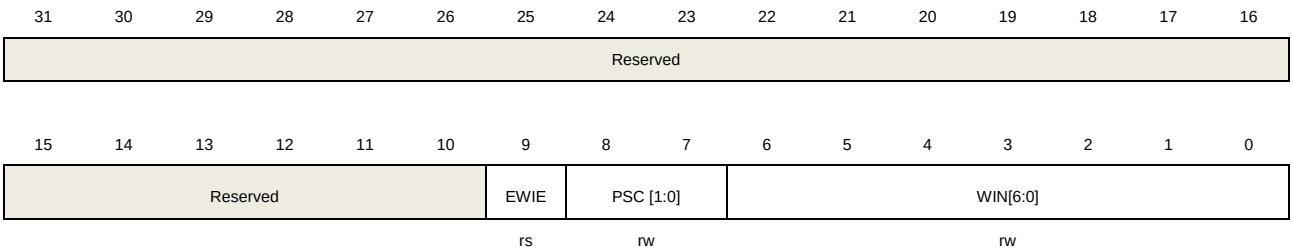


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	WDGTEN	Start the window watchdog timer. Cleared by a hardware reset. Writing 0 has no effect. 0: Window watchdog timer disabled 1: Window watchdog timer enabled
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset will occur when the value of this counter decreases from 0x40 to 0x3F. Writing this counter when the value of this counter is greater than the window value also cause 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

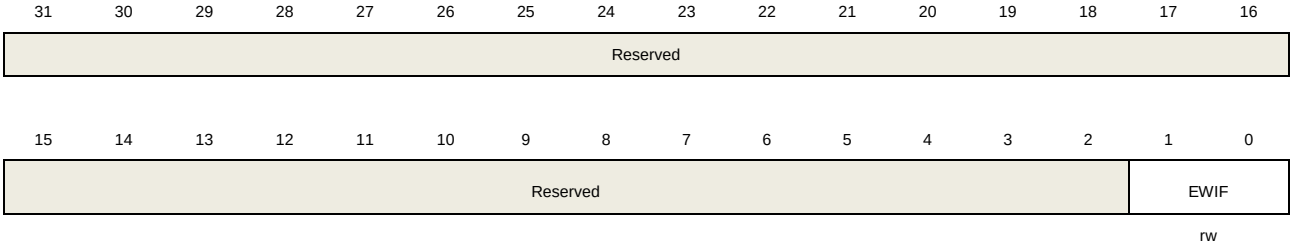
		reaches 0x40. It can be cleared by a hardware reset or software reset by setting the WWDGTRST bit of the RCU module. A write operation of 0 has no effect.
8:7	PSC[1:0]	Prescaler. The time base of the watchdog counter 00: PCLK1 / 4096 / 1 01: PCLK1 / 4096 / 2 10: PCLK1 / 4096 / 4 11: PCLK1 / 4096 / 8
6:0	WIN[6:0]	The Window value. A reset will 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).



Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	EWIF	Early wakeup interrupt flag. When the counter has reached 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 to it.

14. Real-time clock(RTC)

14.1. Overview

The RTC provides a time which includes hour/minute/second/sub-second and a calendar including year/month/day/week day. The time and calendar are expressed in BCD code except sub-second. Sub-second is expressed in binary code. Hour adjustment for daylight saving time. Working in power saving mode and smart wakeup is software configurable. Support improving the calendar accuracy using extern accurate low frequency clock.

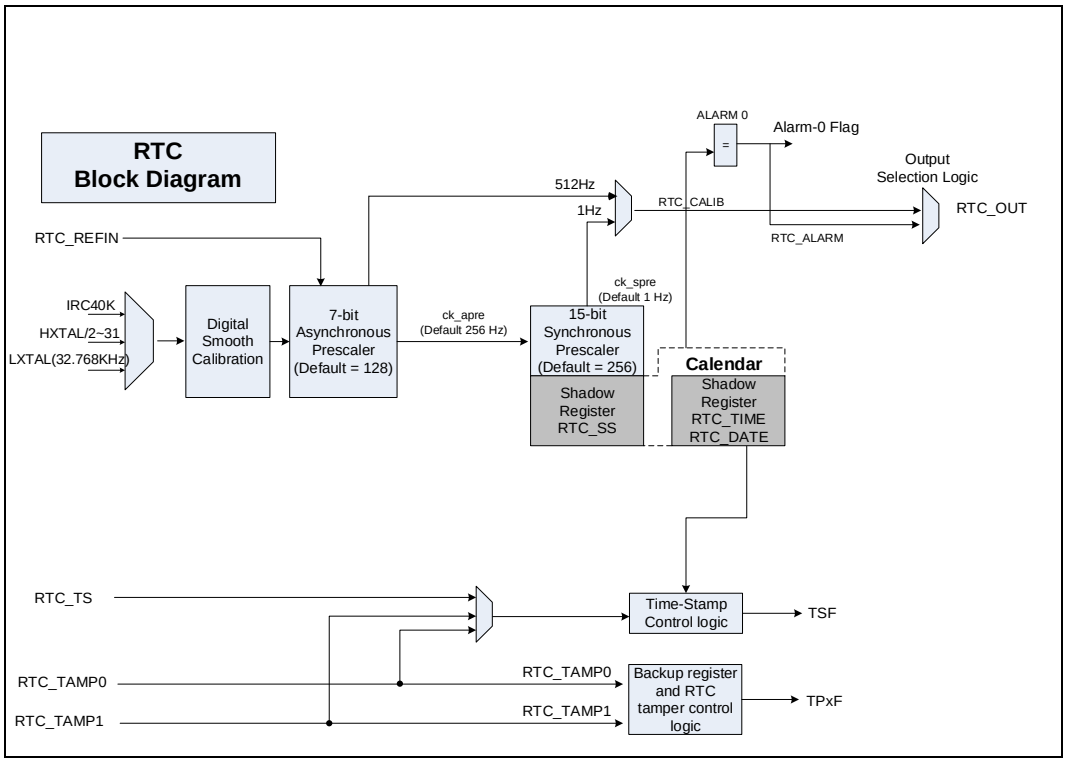
14.2. Characteristics

- Daylight saving compensation supported by software.
- External high-accurate low frequency (50Hz or 60Hz) clock used to achieve higher calendar accuracy performed by reference clock detection option function.
- Atomic clock adjustment (max adjustment accuracy is 0.95PPM) for calendar calibration performed by digital calibration function.
- Sub-second adjustment by shift function.
- Time-stamp function for saving event time.
- Two Tamper sources can be chosen and tamper type is configurable.
- Programmable calendar and one field maskable alarms.
- Maskable interrupt source:
  - Alarm 0
  - Time-stamp detection
  - Tamper detection
- Five 32-bit (20 bytes total) universal backup registers which can keep data under power saving mode. Backup register will be reset if tamper event detected.

14.3. Function overview

14.3.1. Block diagram

Figure 14-1. Block diagram of RTC



The RTC unit includes:

- Alarm event/interrupt
- Tamper event/interrupt
- 32-bit backup registers
- Optional RTC output function:
  - 512Hz (default prescale):RTC_OUT
  - 1Hz(default prescale):RTC_OUT
  - Alarm event(polarity is configurable):RTC_OUT
- Optional RTC input function:
  - time stamp event detection: RTC_TS
  - tamper 0 event detection: RTC_TAMP0
  - tamper 1 event detection: RTC_TAMP1
  - reference clock input: RTC_REFIN(50 or 60 Hz)

14.3.2. Clock source and prescalers

RTC unit has three independent clock sources: LXTAL, IRC40K and HXTAL with divided by 32.

In the RTC unit, there are two prescalers used for implementing the calendar and other functions. One prescaler is a 7-bit asynchronous prescaler and the other is a 15-bit synchronous prescaler. Asynchronous prescaler is mainly used for reducing power consumption. The asynchronous prescaler is recommended to set as high as possible if both prescalers are used.

The frequency formula of two prescalers is shown as below:

$$f_{ck_apre} = \frac{f_{rtcclk}}{FACTOR_A + 1} \quad (14\text{-}1)$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{FACTOR_S + 1} = \frac{f_{rtcclk}}{(FACTOR_A + 1)*(FACTOR_S + 1)} \quad (14\text{-}2)$$

The ck_apre clock is used to driven the RTC_SS down counter which stands for the time left to next second in binary format and when it reaches 0 it will automatically reload FACTOR_S value. The ck_spre clock is used to driven the calendar registers. Each clock will make second plus one.

14.3.3. Shadow registers introduction

BPSHAD control bit decides the location when APB bus accesses the RTC calendar register RTC_DATE, RTC_TIME and RTC_SS. By default, the BPSHAD is cleared, and APB bus accesses the shadow calendar registers. Shadow calendar registers is updated to real calendar register every two RTC clock and at the same time RSYNF bit is set once. This update mechanism is not performed in Deep-Sleep mode and Standby mode. When exiting these modes, software must clear RSYNF bit and wait it is asserted (the max wait time is 2 RTC clock) if software wants to read calendar register under BPSHAD=0 situation.

Note: When reading calendar registers (RTC_SS, RTC_TIME, RTC_DATE) under BPSHAD=0, the frequency of the APB clock (f_{apb}) must be at least 7 times the frequency of the RTC clock (f_{rtcclk}).

Systerm reset will reset the shadow calendar registers.

14.3.4. Configurable and field maskable alarm

RTC alarm function is divided into some fields and each has a maskable bit.

RTC alarm function can be enabled/disabled by ALRM0EN bit in RTC_CTL. If all the alarm fields value match the corresponding calendar value when ALRM0EN=1, the ALRM0F flag will be set.

Note: FACTOR_S in the RTC_PSC register must be larger than 3 if MSKS bit reset in RTC_ALRM0TD.

If a field is masked, the field is considered as matched in logic. If all the fields have been masked, the Alarm Flag will assert 3 RTC clock later after ALRM0EN is set.

14.3.5. RTC initialization and configuration

RTC register write protection

BKPWEN bit in the PMU_CTL register is cleared in default, so writing to RTC registers needs setting BKPWEN bit ahead of time.

After power-on reset, most of RTC registers are write protected. Unlocking this protection is the first step before writing to them.

Following below steps will unlock the write protection:

1. Write '0xCA' into the RTC_WPK register
2. Write '0x53' into the RTC_WPK register

Writing a wrong value to RTC_WPK will make write protection valid again. The state of write protection is not affected by system reset. Following registers are writing protected but others are not:

RTC_TIME, RTC_DATE, RTC_CTL, RTC_STAT, RTC_PSC, RTC_ALRM0TD, RTC_SHIFTCTL, RTC_HRFC, RTC_ALRM0SS.

Calendar initialization and configuration

The prescaler and calendar value can be programmed by following steps:

1. Enter initialization mode (by setting INITM=1) and polling INITF bit until INITF=1.
2. Program both the asynchronous and synchronous prescaler factors in RTC_PSC register.
3. Write the initial calendar values into the shadow calendar registers (RTC_TIME and RTC_DATE), and use the CS bit in the RTC_CTL register to configure the time format (12 or 24 hours).
4. Exit the initialization mode (by setting INITM=0).

About 4 RTC clock cycles later, real calendar registers will load from shadow registers and calendar counter restarts.

Note: Reading calendar register (BPSHAD=0) after initialization, software should confirm the RSYNF bit is already set to 1.

YCM flag indicates whether the calendar has been initialized by checking the year field of calendar.

Daylight saving Time

RTC unit supports daylight saving time adjust function through S1H, A1H and DSM bit.

S1H and A1H can subtract or add 1 hour to the calendar when the calendar is running.S1H and A1H operation can be tautologically set and DSM bit can be used to recording this adjustment operation. After setting the S1H/A1H, subtracting/adding 1 hour will perform when next second comes.

Alarm function operation process

To avoid unexpected alarm assertion and metastable state, alarm function has an operation flow:

1. Disable Alarm (by resetting ALRM0EN in RTC_CTL)
2. Set the Alarm registers needed(RTC_ALRM0TD/RTC_ALRM0SS)
3. Enable Alarm function (by setting ALRM0EN in the RTC_CTL)

14.3.6. Calendar reading

Reading calendar registers under BPSHAD=0

When BPSHAD=0, caldendar value is read from shadow registers. For the existence of synchronization mechanism, a basic request has to meet: the APB1 bus clock frequency must be equal to or greater than 7 times the RTC clock frequency. APB1 bus clock frequency lower than RTC clock frequency is not allowed in any case.

When APB1 bus clock frequency is not equal to or greater than 7 times the RTC clock frequency, the calendar reading flow should be obeyed:

1. reading calendar time register and date register twice
2. if the two values are equal, the value can be seen as the correct value
3. if the two values are not equal, a third reading should performed
4. the third value can be seen as the correct value

RSYNF is asserted once every 2 RTC clock and at this time point, the shadow registers will be updated to current time and date.

To ensure consistency of the 3 values (RTC_SS, RTC_TIME, and RTC_DATE), below consistency mechanism is used in hardware:

1. reading RTC_SS will lock the updating of RTC_TIME and RTC_DATE
2. reading RTC_TIME will lock the updating of RTC_DATE
3. reading RTC_DATE will unlock updating of RTC_TIME and RTC_DATE

If the software wants to read calendar in a short time interval(smaller than 2 RTCCLK periods), RSYNF must be cleared by software after the first calendar read, and then the software must wait until RSYNF is set again before next reading.

In below situations, software should wait RSYNF bit asserted before reading calendar registers(RTC_SS, RTC_TIME, RTC_DATE):

1. after a system reset
2. after an initialization
3. after shift function

Especially that software must clear RSYNF bit and wait it asserted before reading calendar register after wakeup from power saving mode.

Reading calendar registers under BPSHAD=1

When BPSHAD=1, RSYNF is cleared and maintains as 0 by hardware so reading calendar registers does not care about RSYNF bit. Current calendar value is read from real-time calendar counter directly. The benefit of this configuration is that software can get the real current time without any delay after wakeup from power saving mode (Deep-sleep /Standby Mode).

Because of no RSYNF bit periodic assertion, the results of the different calendar registers (RTC_SS/RTC_TIME/RTC_DATE) might not be coherent with each other when clock ck_apre edge occurs between two reading calendar registers.

In addition, if current calendar register is changing and at the same time the APB bus reading calendar register is also performing, the value of the calendar register read out might be not correct.

To ensure the correctness and consistency of the calendar value, software must perform reading operation as this: read all calendar registers continuously, if the last two values are the same, the data is coherent and correct.

14.3.7. Resetting the RTC

There are two reset sources used in RTC unit: system reset and backup domain reset.

System reset will affect calendar shadow registers and some bits of the RTC_STAT. When system reset is valid, the bits or registers mentioned before are reset to the default value.

Backup domain reset will affect the following registers and system reset will not affect them:

- RTC current real-time calendar registers
- RTC control register (RTC_CTL)
- RTC prescaler register (RTC_PSC)
- RTC high resolution frequency compensation register (RTC_HRFC)
- RTC shift register (RTC_SHIFTCTL)
- RTC timestamp registers (RTC_SSTS/RTC_TTS/RTC_DTS)
- RTC tamper and alternate function configuration register (RTC_TAMP)
- RTC backup registers (RTC_BKPx)
- RTC Alarm registers (RTC_ALRM0SS/RTC_ALRM0TD)

The RTC unit will go on running when system reset occurs or enter power saving mode, but if backup domain reset occurs, RTC will stop counting and all registers will reset.

14.3.8. RTC shift function

When there is a remote clock with higher degree of precision and RTC 1Hz clock(ck_spre)has an offset (in a fraction of a second) with the remote clock, RTC unit provides a function named shift function to remove this offset and thus make second precision higher.

RTC_SS register indicates the fraction of a second in binary format and is down counting

when RTC is running. Therefore by adding the SFS[14:0] value to the synchronous prescaler counter SSC[15:0] or by adding the SFS[14:0] value to the synchronous prescaler counter SSC[15:0] and at the same time set A1S bit can delay or advance the time when next second arrives.

The maximal RTC_SS value depends on the FACTOR_S value in RTC_PSC. The higher is FACTOR_S, the higher is adjust precision.

Because of the 1Hz clock(ck_spre) is generated by FACTOR_A and FACTOR_S, the higher FACTOR_S means the lower FACTOR_A, then more power consuming.

Note: Before using shift function, the software must check the MSB of SSC in RTC_SS(SSC[15]) and confirm it is 0.

After writing RTC_SHIFTCTL register, the SOPF bit in RTC_STAT will be set at once. When shift operation is complete, SOPF bit is cleared by hardware. System reset does not affect SOPF bit

Shift operation only works correctly when REFEN=0.

Software must not write to RTC_SHIFTCTL if REFEN=1.

14.3.9. RTC reference clock detection

RTC reference clock detection is another way to increase the precision of RTC second. To enable this function, you should have an external clock source (50Hz or 60 Hz) which is more precise than LXTAL clock source.

After enabling this function (REFEN=1), each 1Hz clock (ck_spre) edge is compared to the nearest RTC_REFIN clock edge. In most cases, the two clock edges are aligned every time. But when two clock edges are misaligned for the reason of LXTAL poor precision, the RTC reference clock detection function will shift the 1Hz clock edge a little to make next 1Hz clock edge aligned to reference clock edge.

When REFEN=1, a time window is applied at every second update time. Different detection state will use different window period.

7 ck_apre window is used when detecting the first reference clock edge and 3 ck_apre window is used for the edge aligned operation.

Whatever window used, the asynchronous prescaler counter will be forced to reload when the reference clock is detected in the window. When the two clock (ck_spre and reference clock) edges are aligned, this reload operation has no effect for 1Hz clock. But when the two clock edge are not aligned, this reload operation will shift ck_spre clock edge a bit to make the ck_spre(1Hz) clock edge aligned to the reference clock edge.

When reference detection function is running while the external reference clock is removed (no reference clock edge found in 3 ck_apre window), the calendar updating still can be performed by LXTAL clock only. If the reference clock is recovered later, detection function will use 7 ck_apre window to identify the reference clock and use 3 ck_apre window to adjust

the 1Hz clock (ck_spre) edge.

Note: Software must configure the FACTOR_A=0x7F and FACTOR_S=0xFF before enabling reference detection function (REFEN=1).

Reference detection function does not work in Standby Mode.

14.3.10. RTC smooth digital calibration

RTC smooth calibration function is a way to calibrate the RTC frequency based on RTC clock in a configurable period time.

This calibration is equally executed in a period time and the cycle number of the RTC clock in the period time will be added or subtracted. The resolution of the calibration is about 0.954PPM with the range from -487.1PPM to +488.5PPM.

The calibration period time can be configured to the $2^{20}/2^{19}/2^{18}$ RTC clock cycles which stands for 32/16/8 seconds if RTC input frequency is 32.768 KHz.

The High resolution frequency compensation register (RTC_HRFC) specifies the number of RTCCLK clock cycles to be calibrated during the period time:

So using CMSK can mask clock cycles from 0 to 511 and thus the RTC frequency can be reduced by up to 487.1PPM.

To increase the RTC frequency the FREQI bit can be set. If FREQI bit is set, there will be 512 additional cycles to be added during period time which means every $2^{11}/2^{10}/2^{9}$(32/16/8 seconds) RTC clock insert one cycle.

So using FREQI can increase the RTC frequency by 488.5PPM.

The combined using of CMSK and FREQI can adjust the RTC cycles from -511 to +512 cycles in the period time which means the calibration range is -487.1PPM to +488.5PPM with a resolution of about 0.954PPM.

When calibration function is running, the output frequency of calibration is calculated by the following formula:

$$f_{cal}=f_{rtcclk}\times\left(1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}\right) \quad (14\text{-}3)$$

Note: N=20/19/18 for 32/16/8 seconds window period

Calibration when FACTOR_A < 3

When asynchronous prescaler value (FACTOR_A) is set to less than 3, software should not set FREQI bit to 1 when using calibration function. FREQI setting will be ignored when FACTOR_A<3.

When the FACTOR_A is less than 3, the FACTOR_S value should be set to a value less than the nominal value. Assuming that RTC clock frequency is nominal 32.768 KHz, the corresponding FACTOR_S should be set as following rule:

FACTOR_A = 2: 2 less than nominal FACTOR_S (8189 with 32.768 KHz)

FACTOR_A = 1: 4 less than nominal FACTOR_S (16379 with 32.768 KHz)

FACTOR_A = 0: 8 less than nominal FACTOR_S (32759 with 32.768 KHz)

When the FACTOR_A is less than 3, CMSK is 0x100，the formula of calibration frequency is as follows:

$$f_{cal}=f_{rtcclk}\times(1+\frac{256\text{-}CMSK}{2^{N}+CMSK-256}) \quad (14\text{-}4)$$

Note: N=20/19/18 for 32/16/8 seconds window period

Verifying the RTC calibration

Calibration 1Hz output is provided to assist software to measure and verify the RTC precision.

Up to 2 RTC clock cycles measurement error may occur when measuring the RTC frequency over a limited measurement period. To eliminate this measurement error the measurement period should be the same as the calibration period.

- When the calibration period is 32 seconds(this is default configuration)

Using exactly 32s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 0.477PPM (0.5 RTCCLK cycles over 32s)

- When the calibration period is 16 seconds(by setting CWND16 bit)

In this configuration, CMSK[0] is fixed to 0 by hardware. Using exactly 16s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within0.954PPM (0.5 RTCCLK cycles over 16s)

- When the calibration period is 8 seconds(by setting CWND8 bit)

In this configuration, CMSK[1:0] is fixed to 0 by hardware. Using exactly 8s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 1.907PPM (0.5 RTCCLK cycles over 8s)

Re-calibration on-the-fly

When the INITF bit is 0, software can update the value of RTC_HRFC using following steps:

1) Wait the SCPF=0
2) Write the new value into RTC_HRFC register
3) After 3 ck_apre clocks, the new calibration settings take effect

14.3.11. Time-stamp function

Time-stamp function is performed on RTC_TS pin and is enabled by control bit TSEN.

When a time-stamp event occurs on RTC_TS pin, the calendar value will be saved in time-stamp registers (RTC_DTS/RTC_TTS/RTC_SSTS) and the time-stamp flag (TSF) is set to 1 by hardware. Time-stamp event can generate an interrupt if time-stamp interrupt enable (TSIE)

is set.

Time-stamp registers only record the calendar at the first time time-stamp event occurs which means that time-stamp registers will not change when TSF=1.

To extend the time-stamp event source, one optional feature is provided: tamper function can also be considered as time-stamp function if TPTS is set.

Note: When the time-stamp event occurs, TSF is set 2 ck_apre cycles delay because of synchronization mechanism.

14.3.12. Tamper detection

The RTC_TAMPx pin input can be used for tamper event detection under edge detection mode or level detection mode with configurable filtering setting.

RTC backup registers (RTC_BKPx)

The RTC backup registers are located in the backup domain that remains powered-on by V_{BAT} even if V_{DD} power is switched off. The wake up action from Standby Mode or system reset are not affect these registers.

These registers are only reset by detected tamper event and backup domain reset.

Tamper detection function initialization

RTC tamper detection function can be independently enabled on tamper input pin by setting corresponding TPxEN bit. Tamper detection configuration is set before enable TPxEN bit. When the tamper event is detected, the corresponding flag (TPxF) will assert. Tamper event can generate an interrupt if tamper interrupt enable (TPIE) is set. Any tamper event will reset all backup registers (RTC_BKPx).

Timestamp on tamper event

The TPTS bit can control whether the tamper detection function is used as time-stamp function. If the bit is set to 1, the TSF bit will be set when the tamper event detected as if “enable” the time-stamp function. Whatever the TPTS bit is, the TPxF will assert when tamper event detected.

Edge detection mode on tamper input detection

When FLT bit is set to 0x0, the tamper detection is set to edge detection mode and TPxEG bit determines the rising edge or falling edge is the detecting edge. When tamper detection is under edge detection mode, the internal pull-up resistors on the tamper detection input pin are deactivated.

Because of detecting the tamper event will reset the backup registers (RTC_BKPx), writing to the backup register should ensure that the tamper event reset and the writing operation will

not occur at the same time, a recommend way to avoid this situation is disable the tamper detection before writing to the backup register and re-enable tamper detection after finish writing.

Note: Tamper detection is still running when V_{DD} power is switched off if tamper is enabled.

Level detection mode with configurable filtering on tamper input detection

When FLT bit is not reset to 0x0, the tamper detection is set to level detection mode and FLT bit determines the consecutive number of samples (2, 4 or 8) needed for valid level. When DISPU is set to 0x0(this is default), the internal pull-up resistance will pre-charge the tamper input pin before each sampling and thus larger capacitance is allowed to connect to the tamper input pin. The pre-charge duration is configured through PRCH bit. Higher capacitance needs long pre-charge time.

The time interval between each sampling is also configurable. Through adjusting the sampling frequency (FREQ), software can balance between the power consuming and tamper detection latency.

14.3.13. Calibration clock output

Calibration clock can be output on the RTC_OUT if COEN bit is set to 1.

When the COS bit is set to 0(this is default) and asynchronous prescaler is set to 0x7F (FACTOR_A), the frequency of RTC_CALIB is $f_{rtcclk}/64$. When the RTCCLK is 32.768KHz, RTC_CALIB output is corresponding to 512Hz. It's recommend to using rising edge of RTC_CALIB output because there may be a light jitter on falling edge.

When the COS bit is set to 1, the RTC_CALIB frequency is:

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (14\text{-}5)$$

When the RTCCLK is 32.768 KHz, RTC_CALIB output is corresponding to 1Hz if prescaler are default values.

14.3.14. Alarm output

When OS control bits are not reset, RTC_ALARM alternate function output is enabled. This function will directly output the content of alarm flag in RTC_STAT.

The OPOL bit in RTC_CTL can configure the polarity of the alarm output which means that the RTC_ALARM output is the opposite of the corresponding flag bit or not.

14.3.15. RTC power saving mode management

Table 14-1. RTC power saving mode management

Mode	Active in Mode	Exit Mode

Sleep	Yes	RTC Interrupts
Deep-Sleep	Yes: if clock source is LXTAL or IRC40K	RTC Alarm/ Tamper Event/ Timestamp Event
Standby	Yes: if clock source is LXTAL or IRC40K	RTC Alarm/ Tamper Event/ Timestamp Event

14.3.16. RTC interrupts

All RTC interrupts are connected to the EXTI controller.

Below steps should be followed if you want to use the RTC alarm/tamper/timestamp:

1) Configure and enable the corresponding interrupt line to RTC alarm/tamper/timestamp event of EXTI and set the rising edge for triggering
2) Configure and enable the RTC alarm/tamper/timestamp global interrupt
3) Configure and enable the RTC alarm/tamper/timestamp function

Table 14-2. RTC interrupts control

Interrupt	Event flag	Control Bit	Exit Sleep	Exit Deep-sleep	Exit Standby
Alarm 0	ALRM0F	ALRM0IE	Y	Y(*)	Y(*)
Timestamp	TSF	TSIE	Y	Y(*)	Y(*)
Tamper 0	TP0F	TPIE	Y	Y(*)	Y(*)
Tamper 1	TP1F	TPIE	Y	Y(*)	Y(*)

* Only active when RTC clock source is LXTAL or IRC40K.

14.4. Register definition

RTC base address: 0x4000 2800

14.4.1. Time register (RTC_TIME)

Address offset: 0x00

System reset: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register is write protected and can only be written in initialization state.

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									PM	HRT[1:0]		HRU[3:0]			
									rw	rw		rw			

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

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	PM	AM/PM mark 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	Reserved	Must be kept at reset value
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code
7	Reserved	Must be kept at reset value
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

14.4.2. Date register (RTC_DATE)

Address offset: 0x04

System reset: 0x0000 2101 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register is write protected and can only be written in initialization state.

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								YRT[3:0]				YRU[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]			MONT	MONU[2:0]				Reserved		DAYT[1:0]		DAYU[3:0]			
rw			rw	rw						rw		rw			

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:20	YRT[3:0]	Year tens in BCD code
19:16	YRU[3:0]	Year units in BCD code
15:13	DOW[2:0]	Days of the week 0x0: Reserved 0x1: Monday … 0x7: Sunday
12	MONT	Month tens in BCD code
11:8	MONU[2:0]	Month units in BCD code
7:6	Reserved	Must be kept at reset value
5:4	DAYT[1:0]	Day tens in BCD code
3:0	DAYU[3:0]	Day units in BCD code

14.4.3. Control register (RTC_CTL)

Address offset: 0x08

System reset: not affected

Backup domain reset: 0x0000 0000

This register is writing protected.

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								COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
								rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	Reserved		ALRM0IE	TSEN	Reserved		ALRM0EN	Reserved	CS	BPSHAD	REFEN	TSEG	Reserved		
rw			rw	rw			rw		rw	rw	rw	rw			

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value

23	COEN	Calibration output enable 0: Disable calibration output 1: Enable calibration output
22:21	OS[1:0]	Output selection This bit is used for selecting flag source to output 0x0: Disable output RTC_ALARM 0x1: Enable alarm0 flag output 0x2: Reserved 0x3: Reserved
20	OPOL	Output polarity This bit is used to invert output RTC_ALARM 0: Disable invert output RTC_ALARM 1: Enable invert output RTC_ALARM
19	COS	Calibration output selection Valid only when COEN=1 and prescalers are at default values 0: Calibration output is 512 Hz 1: Calibration output is 1Hz
18	DSM	Daylight saving mark This bit is flexible used by software. Often can be used to recording the daylight saving hour adjustment.
17	S1H	Subtract 1 hour(winter time change) One hour will be subtracted from current time if it is not 0 0: No effect 1: 1 hour will be subtracted at next second change time.
16	A1H	Add 1 hour(summer time change) One hour will be added from current time 0: No effect 1: 1 hour will be added at next second change time
15	TSIE	Time-stamp interrupt enable 0: Disable time-stamp interrupt 1: Enable time-stamp interrupt
14:13	Reserved	Must be kept at reset value
12	ALRM0IE	RTC alarm-0 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
11	TSEN	Time-stamp function enable 0: Disable time-stamp function 1: Enable time-stamp function

10:9	Reserved	Must be kept at reset value
8	ALRM0EN	Alarm-0 function enable 0: Disable alarm function 1: Enable alarm function
7	Reserved	Must be kept at reset value
6	CS	Clock System 0: 24-hour format 1: 12-hour format **Note:** Can only be written in initialization state
5	BPSHAD	Shadow registers bypass control 0: Reading calendar from shadow registers 1: Reading calendar from current real-time calendar **Note:** If frequency of APB1 clock is less than seven times the frequency of RTCCLK, this bit must set to 1.
4	REFEN	Reference clock detection function enable 0: Disable reference clock detection function 1: Enable reference clock detection function **Note:** Can only be written in initialization state and FACTOR_S must be 0x00FF
3	TSEG	Valid event edge of time-stamp 0: rising edge is valid event edge for time-stamp event 1: falling edge is valid event edge for time-stamp event
2:0	Reserved	Must be kept at reset value

14.4.4. Status register (RTC_STAT)

Address offset: 0x0C
System reset: Only INITM, INITF and RSYNF bits are set to 0. Others are not affected
Backup domain reset: 0x0000 0007

This register is writing protected except RTC_STAT[14:8].
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															SCPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TP1F	TP0F	TSOVRF	TSF	Reserved		ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	Reserved		ALRM0WF
	rc_w0	rc_w0	rc_w0	rc_w0			rc_w0	rw	r	rc_w0	r	r			r

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value

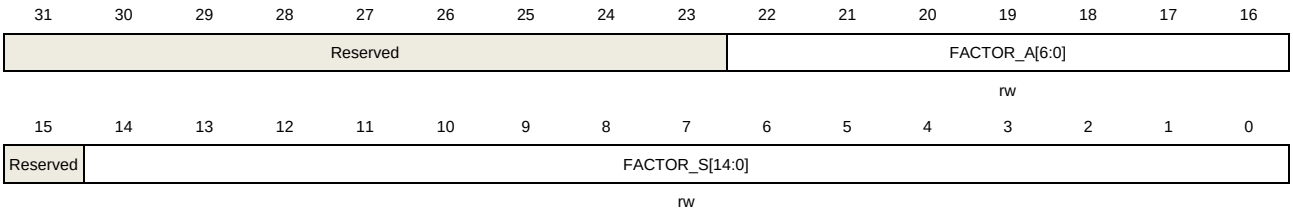
16	SCPF	Smooth calibration pending flag Set to 1 by hardware when software writes to RTC_HRFC and set to 0 by hardware when smooth calibration window is completed.
15	Reserved	Must be kept at reset value
14	TP1F	RTC_TAMP1 detected flag Set to 1 by hardware when tamper detection is found on tamper1 input pin. Software can clear this bit by writing 0 into this bit.
13	TP0F	RTC_TAMP0 detected flag Set to 1 by hardware when tamper detection is found on tamper0 input pin. Software can clear this bit by writing 0 into this bit.
12	TSOVRF	Time-stamp overflow flag This bit is set by hardware when a time-stamp event is detected if TSF bit is set before. Cleared by software writing 0.
11	TSF	Time-stamp flag Set by hardware when time-stamp event is detected. Cleared by software writing 0.
10:9	Reserved	Must be kept at reset value
8	ALRM0F	Alarm-0 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 0 setting value. Cleared by software writing 0.
7	INITM	Enter initialization mode 0: Free running mode 1: Enter initialization mode for setting calendar time/date and prescaler. Counter will stop under this mode.
6	INITF	Initialization state flag Set to 1 by hardware, calendar registers and prescaler can be programmed in this state. 0:Calendar registers and prescaler register cannot be changed 1:Calendar registers and prescaler register can be changed
5	RSYNF	Register synchronization flag Set to 1 by hardware every 2 RTCCLK which will copy current calendar time/date into shadow register. Initialization mode(INITM), shift operation pending flag(SOPF) or bypass mode(BPSHAD) will clear this bit. This bit is also can be cleared by software writing 0. 0:Shadow register are not yet synchronized 1:Shadow register are synchronized
4	YCM	Year configuration mark

		Set by hardware if the year field of calendar date register is not the default value 0. 0:Calendar has not been initialized 1:Calendar has been initialized
3	SOPF	Shift function operation pending flag 0:No shift operation is pending 1:Shift function operation is pending
2:1	Reserved	Must be kept at reset value
0	ALRM0WF	Alarm 0 configuration can be write flag Set by hardware if alarm register can be written after ALRM0EN bit has reset. 0:Alarm registers programming is not allowed 1:Alarm registers programming is allowed

14.4.5. Prescaler register (RTC_PSC)

Address offset: 0x10
System reset: not effected
Backup domain reset: 0x007F 00FF

This register is write protected and can only be written in initialization state.
This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22:16	FACTOR_A[6:0]	Asynchronous prescaler factor ck_apre frequency = RTCCLK frequency/(FACTOR_A+1)
15	Reserved	Must be kept at reset value
14:0	FACTOR_S[14:0]	Synchronous prescaler factor ck_spre frequency = ck_apre frequency/(FACTOR_S+1)

14.4.6. Alarm 0 time and date register (RTC_ALRM0TD)

Address offset: 0x1C
System reset: not effect
Backup domain reset: 0x0000 0000

This register is write protected and can only be written in initialization state.

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

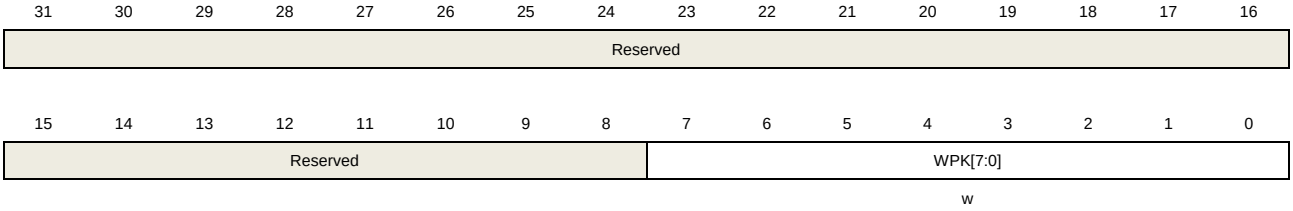
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

Bits	Fields	Descriptions
31	MSKD	Alarm date mask bit 0:Not mask date/day field 1:Mask date/day field
30	DOWS	Day of the week selected 0:DAYU[3:0] indicates the date units 1: DAYU[3:0] indicates the week day and DAYT[1:0] has no means.
29:28	DAYT[1:0]	Date tens in BCD code
27:24	DAYU[3:0]	Date units or week day in BCD code
23	MSKH	Alarm hour mask bit 0:Not mask hour field 1:Mask hour field
22	PM	AM/PM flag 0:AM or 24-hour format 1:PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	MSKM	Alarm minutes mask bit 0:Not mask minutes field 1:Mask minutes field
14:12	MNT[2:0]	Minutes tens in BCD code
11:8	MNU[3:0]	Minutes units in BCD code
7	MSKS	Alarm second mask bit 0:Not mask second field 1:Mask second field
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

14.4.7. Write protection key register (RTC_WPK)

Address offset: 0x24
Reset value: 0x0000 0000

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

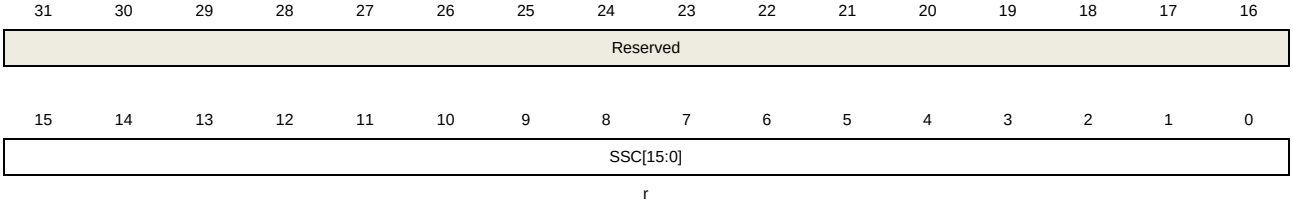


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	WPK[7:0]	Key for write protection

14.4.8. Sub second register (RTC_SS)

Address offset: 0x28
System reset: 0x0000 0000 when BPSHAD = 0.
Not affected when BPSHAD = 1.

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler. Second fraction value is calculated by the below formula: Second fraction = (FACTOR_S - SSC) / (FACTOR_S + 1)

14.4.9. Shift function control register (RTC_SHIFTCTL)

Address offset: 0x2C
System reset: not effect
Backup domain reset: 0x0000 0000

This register is writing protected and can only be wrote when SOPF=0.

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
A1S	Reserved														
w															

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

Bits	Fields	Descriptions
31	A1S	One second add 0:Not add 1 second 1:Add 1 second to the clock/calendar. This bit is jointly used with SFS field to add a fraction of a second to the clock.
30:15	Reserved	Must be kept at reset value
14:0	SFS[14:0]	Subtract a fraction of a second The value of this bit will add to the counter of synchronous prescaler. When only using SFS, the clock will delay because the synchronous prescaler is a down counter: Delay (seconds) = SFS / (FACTOR_S + 1) When jointly using A1S and SFS, the clock will advance: Advance (seconds) = (1 - (SFS / (FACTOR_S + 1)))

Note: Writing to this register will cause RSYNF bit to be cleared.

14.4.10. Time of time stamp register (RTC_TTS)

Address offset: 0x30

Backup domain reset: 0x0000 0000

System reset: no effect

This register will record the calendar time when TSF is set to 1.

Reset TSF bit will also clear this 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									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MNT[2:0]			MNU[3:0]				Reserved	SCT[2:0]			SCU[3:0]			
	r			r					r			r			

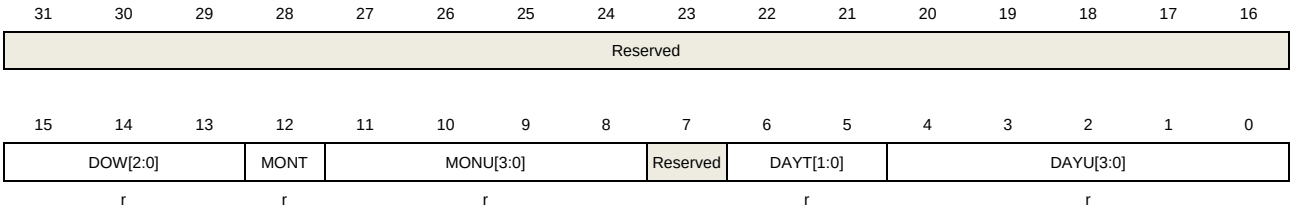
Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value
22	PM	AM/PM mark

		0:AM or 24-hour format 1:PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	Reserved	Must be kept at reset value
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code
7	Reserved	Must be kept at reset value
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

14.4.11. Date of time stamp register (RTC_DTS)

Address offset: 0x34
Backup domain reset: 0x0000 0000
System reset: no effect

This register will record the calendar date when TSF is set to 1.
Reset TSF bit will also clear this register.
This register has to be accessed by word(32-bit).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:13	DOW[2:0]	Days of the week
12	MONT	Month tens in BCD code
11:8	MONU[3:0]	Month units in BCD code
7	Reserved	Must be kept at reset value
6:5	DAYT[1:0]	Day tens in BCD code
4:0	DAYU[3:0]	Day units in BCD code

14.4.12. Sub second of time stamp register (RTC_SSTS)

Address offset: 0x38
Backup domain reset: 0x0000 0000
System reset: no effect

This register will record the calendar date when TSF is set to 1.
Reset TSF bit will also clear this 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
SSC[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF is set to 1.

14.4.13. High resolution frequency compensation register (RTC_HRFC)

Address offset: 0x3C
Backup domain reset: 0x0000 0000
System Reset: no effect

This register is write protected.
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
FREQI	CWND8	CWND16	Reserved								CMSK[8:0]				
rw	rw	rw									rw				

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	FREQI	Increase RTC frequency by 488.5PPM 0: No effect 1: One RTCCLK pulse is inserted every 2^{11} pulses. This bit should be used in conjunction with CMSK bit. If the input clock frequency is 32.768KHz, the number of RTCCLK pulses added during 32s calibration window is

		(512 * FREQI) - CMSK
14	CWND8	Frequency compensation window 8 second selected 0:No effect 1:Calibration window is 8 second **Note:** When CWND8=1, CMSK[1:0] are stuck at “00”.
13	CWND16	Frequency compensation window 16 second selected 0:No effect 1:Calibration window is 16 second **Note:** When CWND16=1, CMSK[0] are stuck at “0”.
12:9	Reserved	Must be kept at reset value
8:0	CMSK[8:0]	Calibration mask number The number of mask pulse out of 2^{20} RTCCLK pulse. This feature will decrease the frequency of calendar with a resolution of 0.9537 PPM.

14.4.14. Tamper register (RTC_TAMP)

Address offset: 0x40
Backup domain reset: 0x0000 0000
System reset: no effect

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								PC15MDE	PC15VAL	PC14MDE	PC14VAL	PC13MDE	PC13VAL	Reserved	
								rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]		FLT[1:0]		FREQ[2:0]			TPTS	Reserved		TP1EG	TP1EN	TPIE	TP0EG	TP0EN
rw	rw		rw		rw			rw			rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23	PC15MDE	PC15 Mode 0:No effect 1:Force PC15 to push-pull output if LXTAL is disable
22	PC15VAL	PC15 Value Only valid when LXTAL is disabled and PC15MDE=1,PC15 output this bit data.
21	PC14MDE	PC14 Mode 0:No effect 1:Force PC14 to push-pull output if LXTAL is disable
20	PC14VAL	PC14 Value

		Only valid when LXTAL is disabled and PC14MDE=1,PC14 output this bit data.
19	PC13MDE	PC13 Mode 0:No effect 1:Force PC13 to push-pull output if all RTC alternate functions are disabled.
18	PC13VAL	PC13 value or alarm output type value When PC13 is used to output alarm: 0:PC13 is in open-drain output type 1:PC13 is in push-pull output type When all RTC alternate functions are disabled and PC13MDE=1: 0:PC13 output 0 1:PC13 output 1
17:16	Reserved	Must be kept at reset value
15	DISPU	RTC_TAMPx pull up disable bit 0:Enable inner pull-up before sampling for pre-charge RTC_TAMPx pin 1:Disable pre-charge duration
14:13	PRCH[1:0]	Pre-charge duration time of RTC_TAMPx This setting determines the pre-charge time before each sampling. 0x0:1 RTC clock 0x1:2 RTC clock 0x2:4 RTC clock 0x3:8 RTC clock
12:11	FLT[1:0]	RTC_TAMPx filter count setting This bit determines the tamper sampling type and the number of consecutive sample. 0x0: Detecting tamper event using edge mode. Pre-charge duration is disabled automatically 0x1: Detecting tamper event using level mode.2 consecutive valid level samples will make an effective tamper event 0x2:Detecting tamper event using level mode.4 consecutive valid level samples will make an effective tamper event 0x3:Detecting tamper event using level mode.8 consecutive valid level samples will make an effective tamper event
10:8	FREQ[2:0]	Sampling frequency of tamper event detection 0x0: Sample once every 32768 RTCCLK(1Hz if RTCCLK=32.768KHz) 0x1: Sample once every 16384 RTCCLK(2Hz if RTCCLK=32.768KHz) 0x2: Sample once every 8192 RTCCLK(4Hz if RTCCLK=32.768KHz) 0x3: Sample once every 4096 RTCCLK(8Hz if RTCCLK=32.768KHz) 0x4: Sample once every 2048 RTCCLK(16Hz if RTCCLK=32.768KHz) 0x5: Sample once every 1024 RTCCLK(32Hz if RTCCLK=32.768KHz) 0x6: Sample once every 512 RTCCLK(64Hz if RTCCLK=32.768KHz)

		0x7: Sample once every 256 RTCCLK(128Hz if RTCCLK=32.768KHz)
7	TPTS	Make tamper function used for timestamp function 0:No effect 1:TSF is set when tamper event detected even TSEN=0
6:5	Reserved	Must be kept at reset value
4	TP1EG	Tamper 1 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
3	TP1EN	Tamper 1 detection enable 0:Disable tamper 1 detection function 1:Enable tamper 1 detection function
2	TPIE	Tamper detection interrupt enable 0:Disable tamper interrupt 1:Enable tamper interrupt
1	TP0EG	Tamper 0 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
0	TP0EN	Tamper 0 detection enable 0:Disable tamper 0 detection function 1:Enable tamper 0 detection function

Note: It's strongly recommended that reset the TPxEN before change the tamper configuration.

14.4.15. Alarm 0 sub second register (RTC_ALRM0SS)

Address offset: 0x44
Backup domain reset: 0x0000 0000
System reset: no effect

This register is write protected and can only be wrote when ALRM0EN=0 or INITM=1.
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				MSKSSC[3:0]				Reserved							

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

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27:24	MSKSSC[3:0]	Mask control bit of SSC 0x0: Mask alarm sub second setting. The alarm asserts at every second time point if all the rest alarm fields are matched. 0x1: SSC[0] is to be compared and all others are ignored 0x2: SSC[1:0] is to be compared and all others are ignored 0x3: SSC[2:0] is to be compared and all others are ignored 0x4: SSC[3:0] is to be compared and all others are ignored 0x5: SSC[4:0] is to be compared and all others are ignored 0x6: SSC[5:0] is to be compared and all others are ignored 0x7: SSC[6:0] is to be compared and all others are ignored 0x8: SSC[7:0] is to be compared and all others are ignored 0x9: SSC[8:0] is to be compared and all others are ignored 0xA: SSC[9:0] is to be compared and all others are ignored 0xB: SSC[10:0] is to be compared and all others are ignored 0xC: SSC[11:0] is to be compared and all others are ignored 0xD: SSC[12:0] is to be compared and all others are ignored 0xE: SSC[13:0] is to be compared and all others are ignored 0xF: SSC[14:0] is to be compared and all others are ignored **Note:** The bit 15 of synchronous counter (SSC[15] in RTC_SS) is never compared.
23:15	Reserved	Must be kept at reset value
14:0	SSC[14:0]	Alarm sub second value This value is the alarm sub second value which is to be compared with synchronous prescaler counter SSC. Bit number is controlled by MSKSSC bits.

14.4.16. Backup registers (RTC_BKPx) (x=0..4)

Address offset: 0x50~0x60
Backup domain reset: 0x0000 0000
System reset: no effect

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[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

rw

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data These registers can be wrote or read by software. The content remains valid even in power saving mode because they can powered-on by V_{BAT}. Tamper detection flag TPxF assertion will reset these registers. Also when the FMC readout protection disables will reset these registers.

15. Timer (TIMERx)

Table 15-1. Timers (TIMERx) are devided into six sorts

TIMER	TIMER0	TIMER1/2	TIMER13	TIMER14	TIMER15/16	TIMER5
TYPE	Advanced	General-L0	General-L2	General-L3	General-L4	Basic
Prescaler	16-bit	16-bit	16-bit	16-bit	16-bit	16-bit
Counter	16-bit	32-bit(TIMER1) 16-bit(TIMER2)	16-bit	16-bit	16-bit	16-bit
Count mode	UP,DOWN, Center-aligned	UP,DOWN, Center-aligned	UP ONLY	UP ONLY	UP ONLY	UP ONLY
Repetition	●	×	×	●	●	×
CH Capture/ Compare	4	4	1	2	1	0
Complementary & Dead-time	●	×	×	●	●	×
Break	●	×	×	●	●	×
Single Pulse	●	●	×	●	●	●
Quadrature Decoder	●	●	×	×	×	×
Master-slave management	●	●	×	●	×	×
Inter connection	●(1)	●(2)	×	●(3)	×	TRGO TO DAC
DMA	●	●	×	●	●	●(4)
Debug Mode	●	●	●	●	●	●

(1) TIMER0 **ITI0:** TIMER14_TRGO **ITI1:** TIMER1_TRGO **ITI2:** TIMER2_TRGO **ITI3:** 0

(2) TIMER1 **ITI0:** TIMER0_TRGO **ITI1:** TIMER14_TRGO **ITI2:** TIMER2_TRGO **ITI3:** 0
TIMER2 **ITI0:** TIMER0_TRGO **ITI1:** TIMER1_TRGO **ITI2:** TIMER14_TRGO **ITI3:** 0

(3) TIMER14 **ITI0:** TIMER1_TRGO **ITI1:** TIMER2_TRGO **ITI2:** 0 **ITI3:** 0

(4) Only update events will generate DMA request. Note that TIMER5 do not have DMA configuration registers.

15.1. Advanced timer (TIMERx,x=0)

15.1.1. Overview

The advanced timer module (TIMER0) is a four-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 issuitable for motor control applications.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counters incrementing in unison.

15.1.2. Characteristics

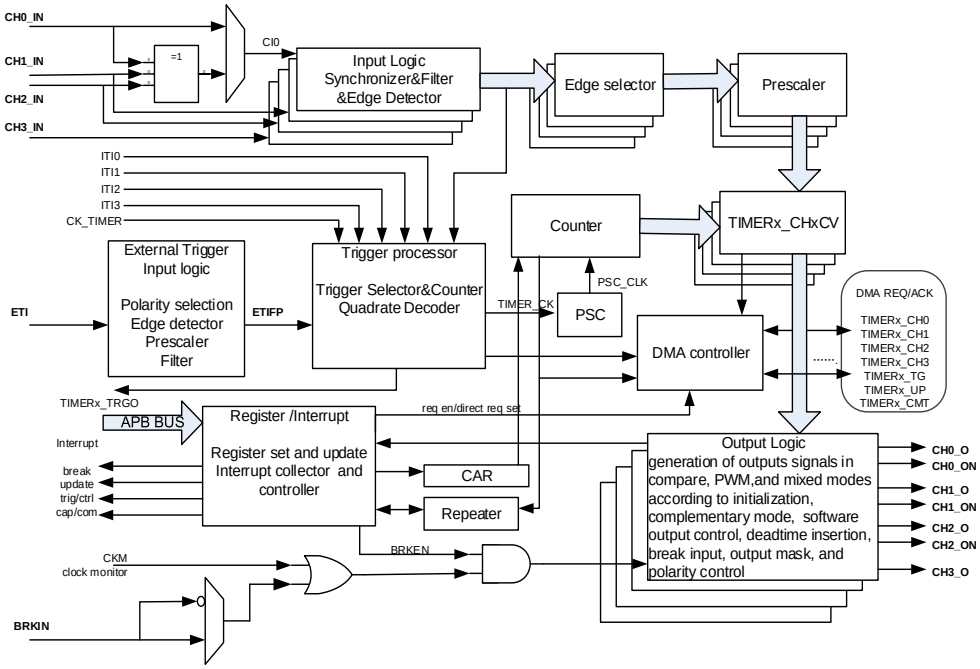
- Total channel num: 4.
- Counter width: 16 bit.
- Source of counter clock is selectable:
  internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: count up, count down, count up/down.
- Quadrature Decoder: used to track motion and determine both rotation direction and position.
- Hall sensor: for 3-phase motor control.
- Programmable prescaler: 16 bit.The factor can be changed on the go.
- Each channel is user-configurable:
  input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request on: update, trigger event, compare/capture event, and break input.
- Daisy chaining of timer modules allows a single timer to initiate multiple timers.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer master-slave management.

15.1.3. Block diagram

Figure 15-1. Advanced timer block diagram provides details of the internal configuration of

the advanced timer.

Figure 15-1. Advanced timer block diagram



15.1.4. Function overview

Clock source configuration

The advanced 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]).

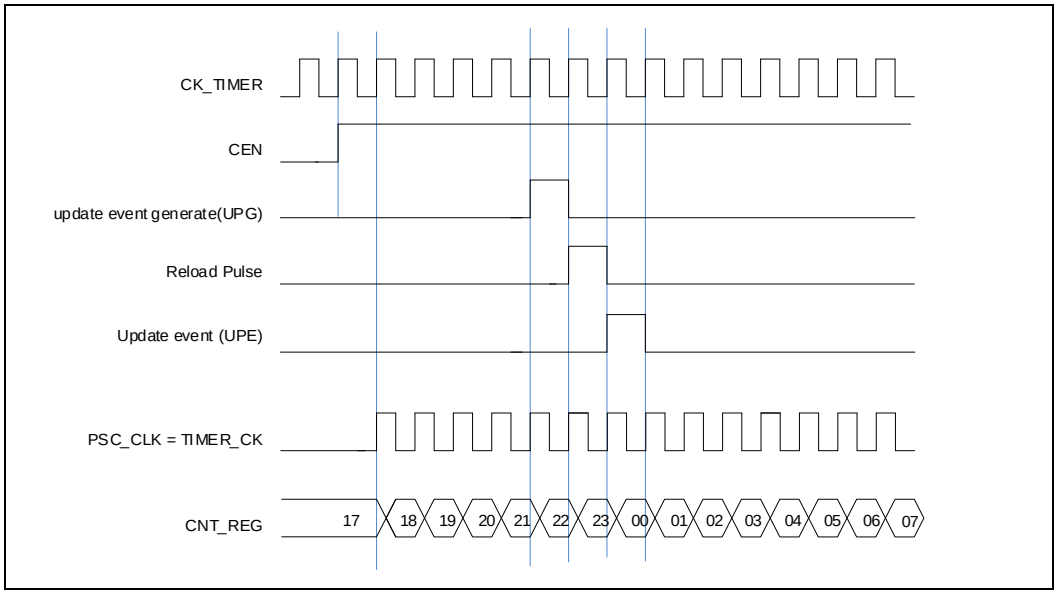
- SMC [2:0] == 3'b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

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.

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, details as follows. 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 15-2. 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_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.

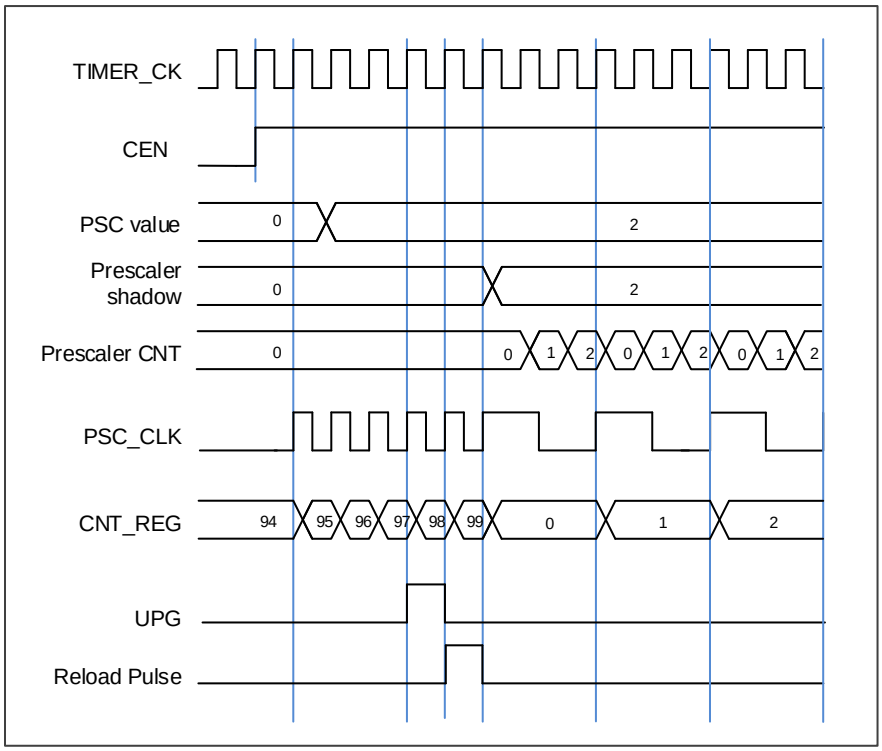
- SMC1== 1'b1 (external clock mode 1). External input ETI is selected as timer clock source (ETI)

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 to set the SMC [2:0] to 0x7 and the TRGS [2:0] to 0x7 respectively. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as 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 prescale 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 15-3. 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 generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter, counter auto reload register, prescaler register) are updated.

Figure 15-4. Timing chart of up counting mode, PSC=0/2

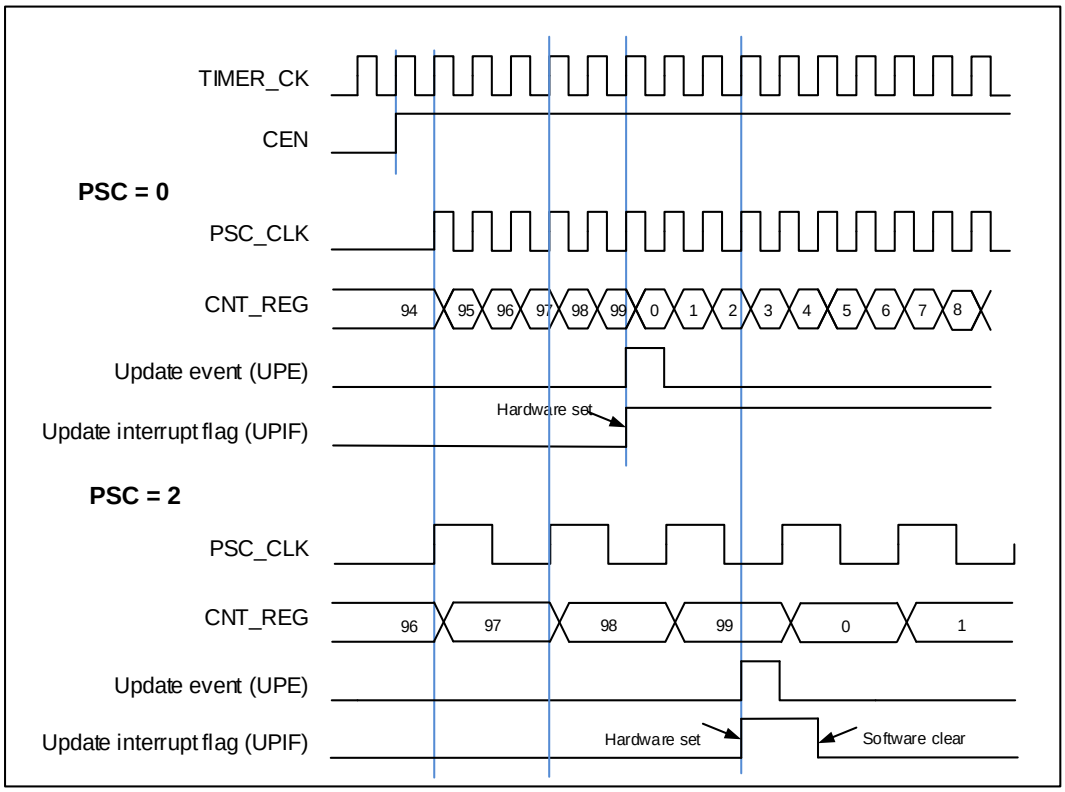
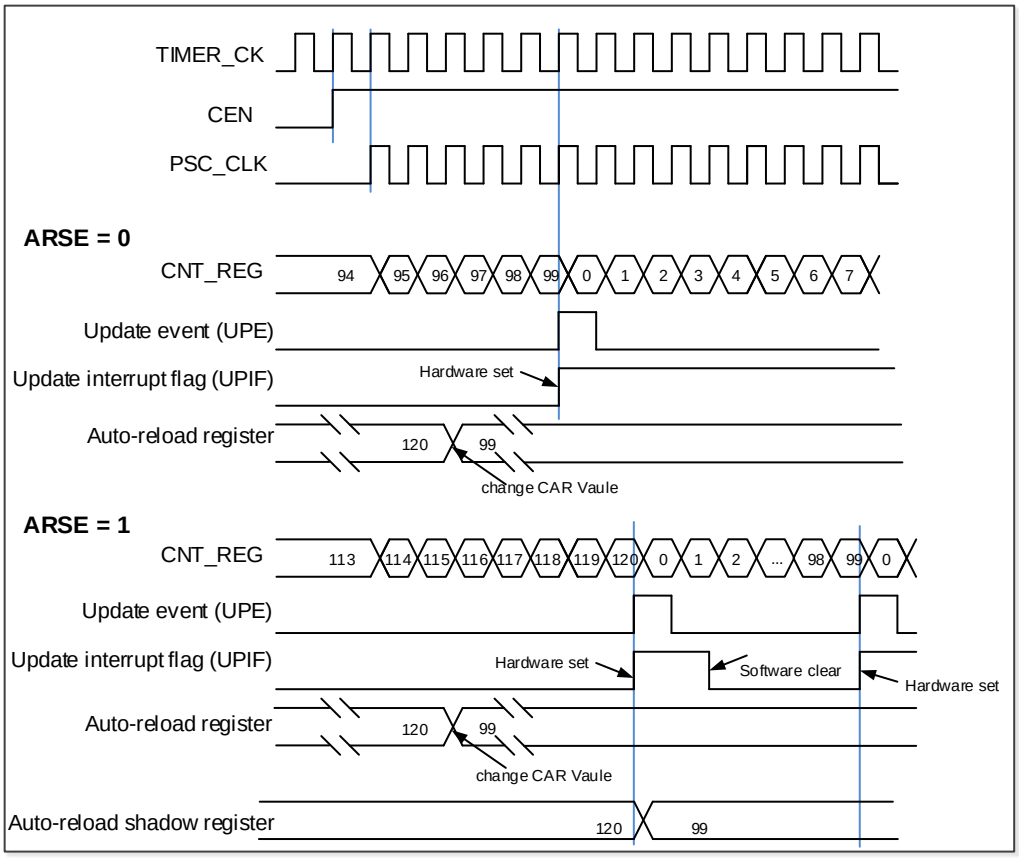


Figure 15-5. Timing chart of up counting mode, change TIMERx_CAR on the go



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 generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter, counter auto reload register, prescaler register) are updated.

Figure 15-6. Timing chart of down counting mode, PSC=0/2 and *Figure 15-7. 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 15-6. Timing chart of down counting mode, PSC=0/2

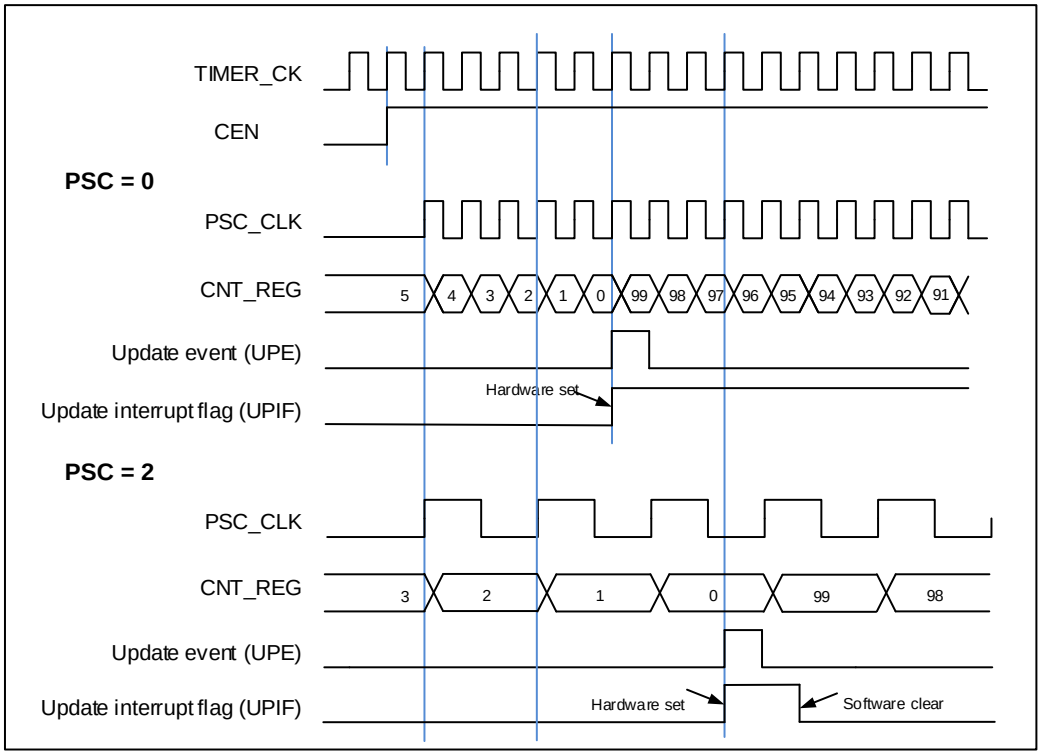
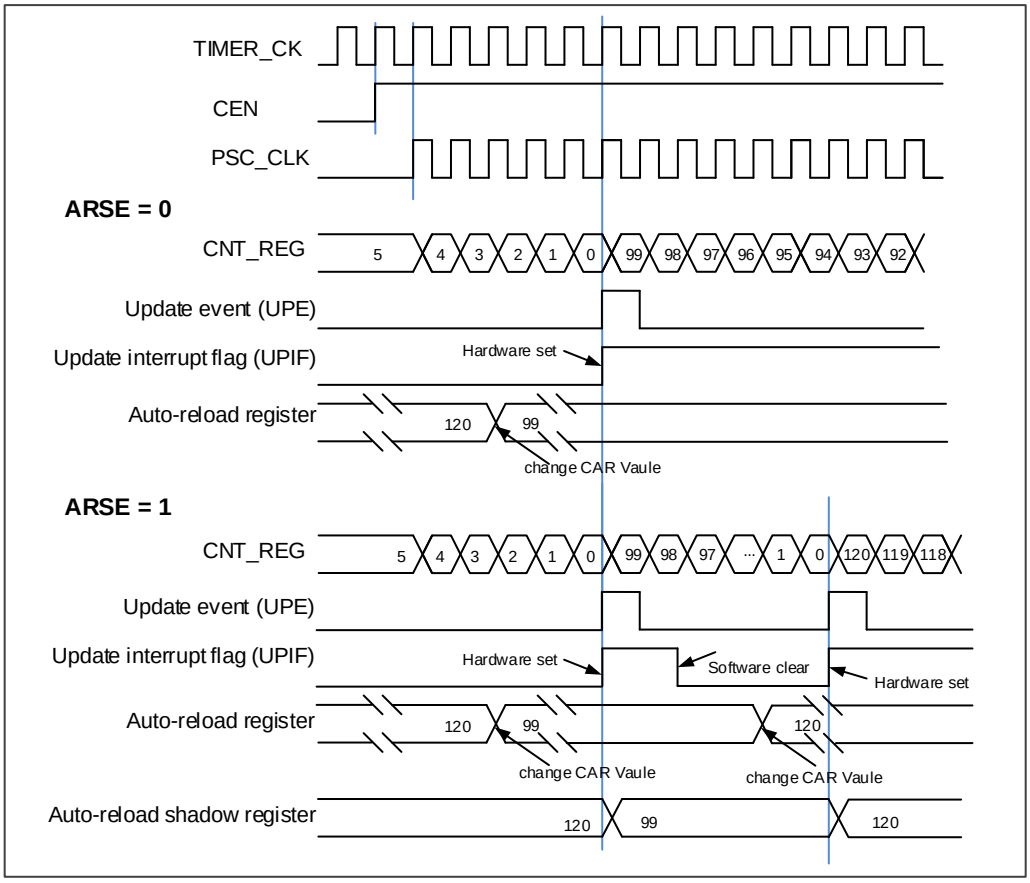


Figure 15-7. Timing chart of down counting mode, change TIMERx_CAR on the go



Counter center-aligned counting

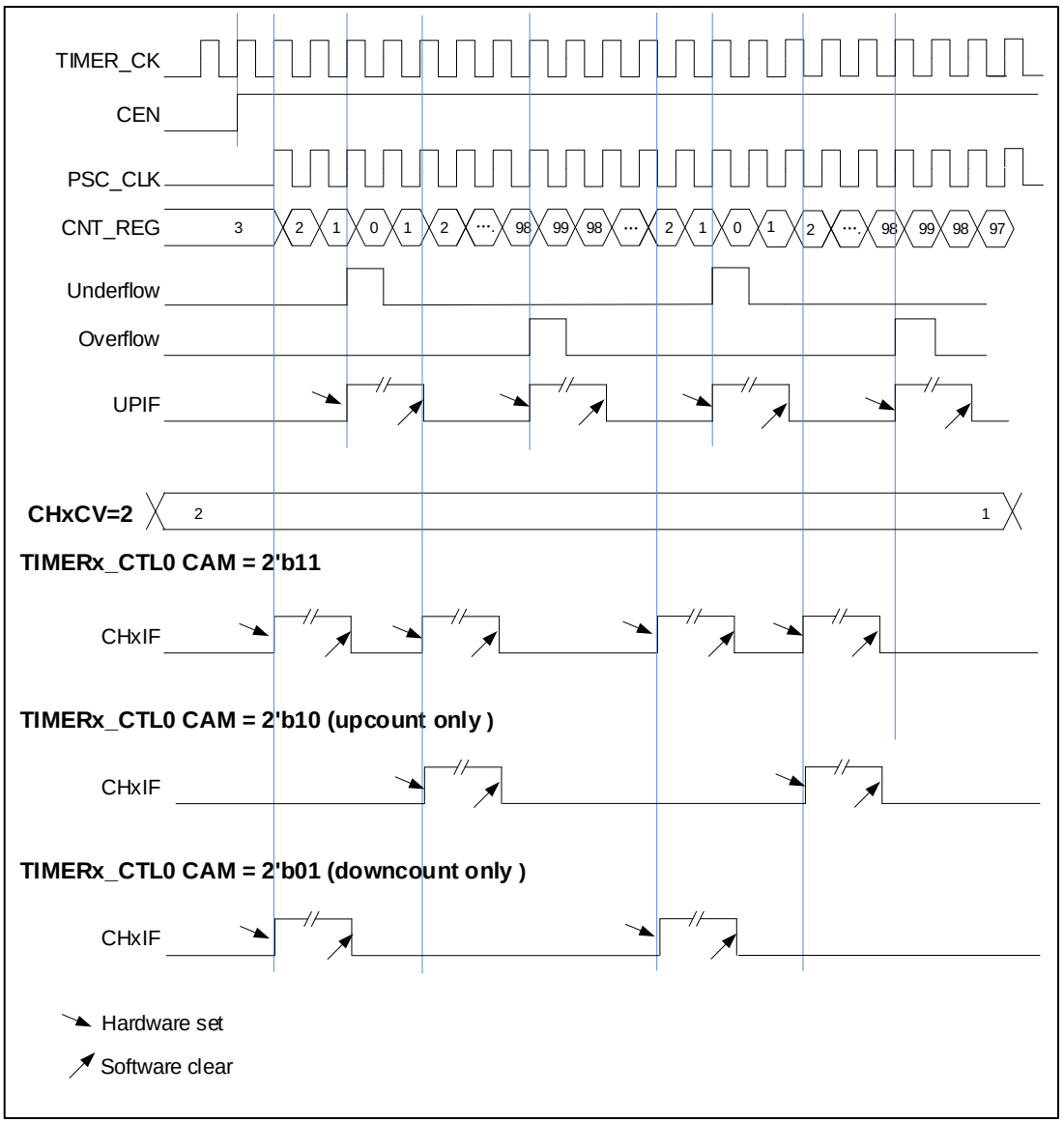
In this mode, the counter counts up from 0 to the counter-reload value and then counts down to 0 alternatively. The Timer module generates an overflow event when the counter counts to the counter-reload value subtract 1 in the up-counting direction and generates an underflow event when the counter counts 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 generates an update event irrespective of whether the counter is counting up or down in the center-align counting mode.

The UPIF bit in the TIMERx_INTF register can 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 15-8. Center-aligned counter timechart.***

Figure 15-8. Center-aligned counter timechart show some examples of the counter behavior for different clock frequencies when TIMERx_CAR=0x99. TIMERx_PSC=0x0

Figure 15-8. Center-aligned counter timechart



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 generator 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 15-9. Repetition counter timing chart of center-aligned counting mode

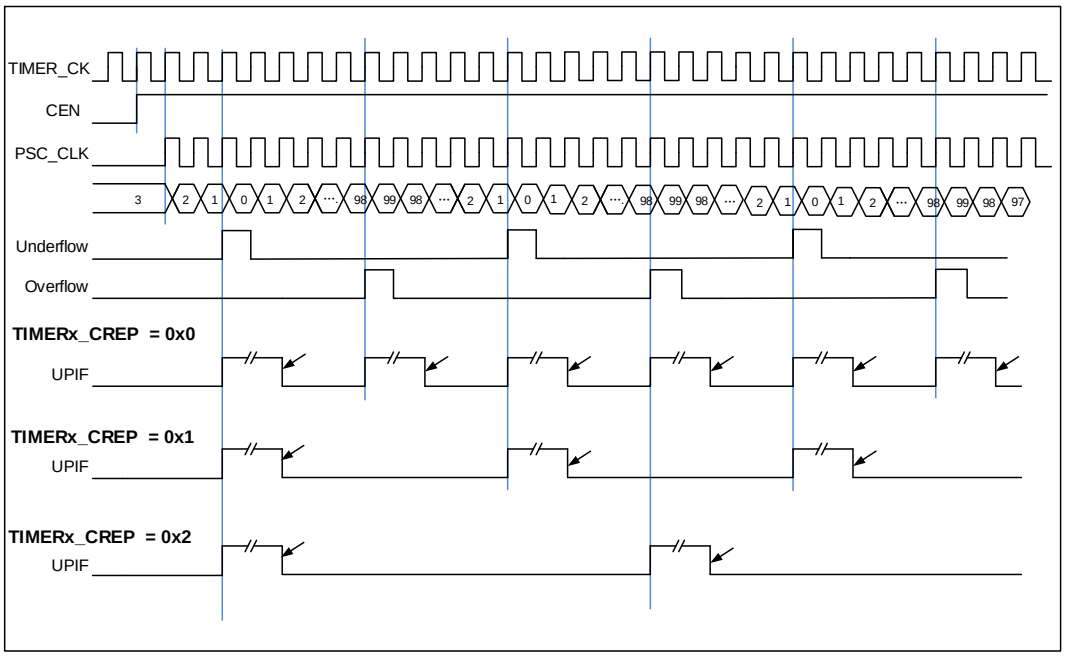


Figure 15-10. Repetition counter timing chart of up counting mode

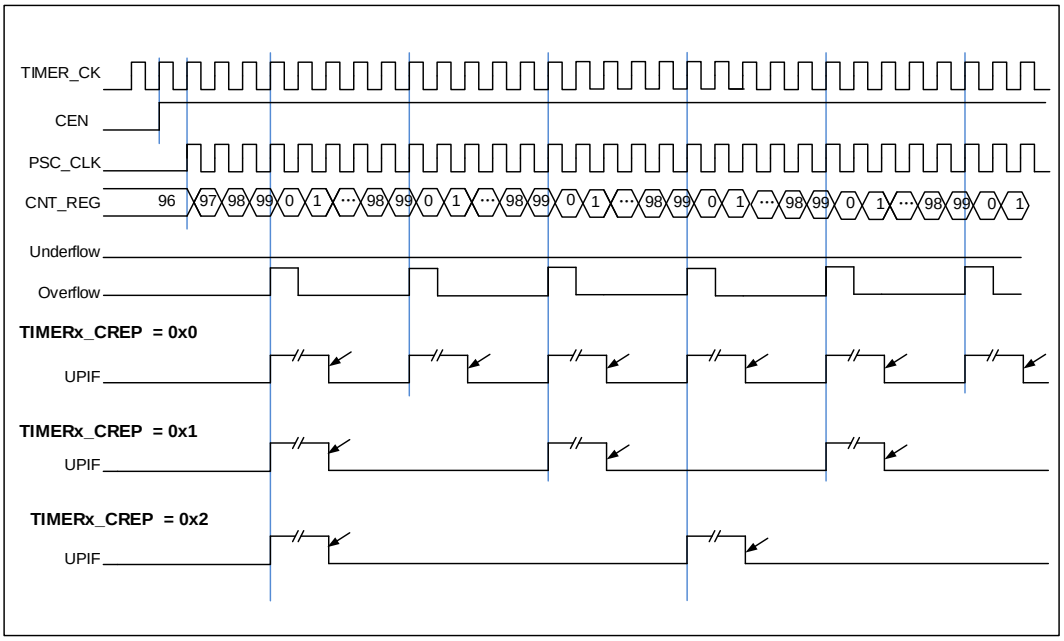
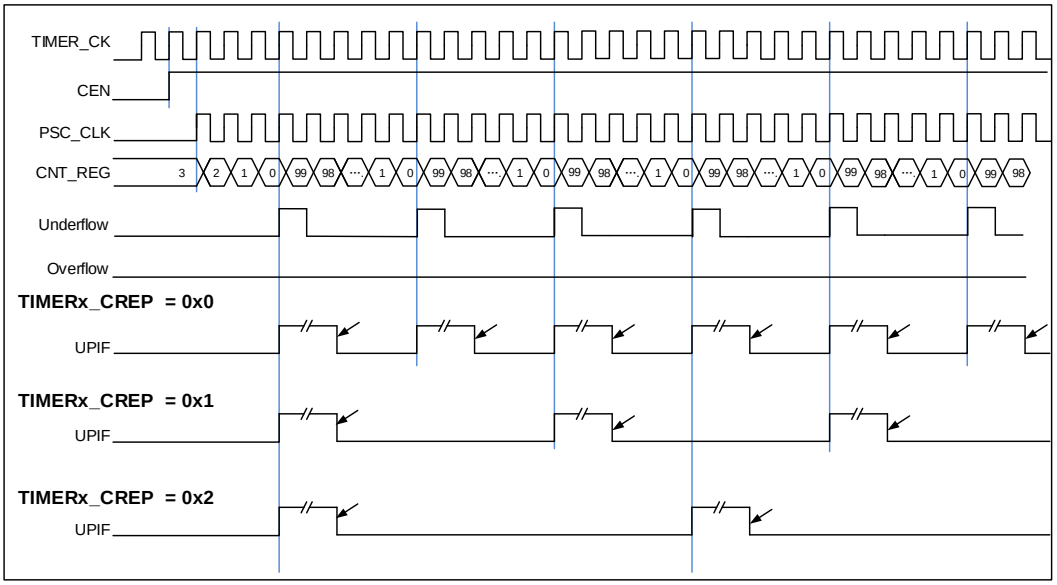


Figure 15-11. Repetition counter timing chart of down counting mode



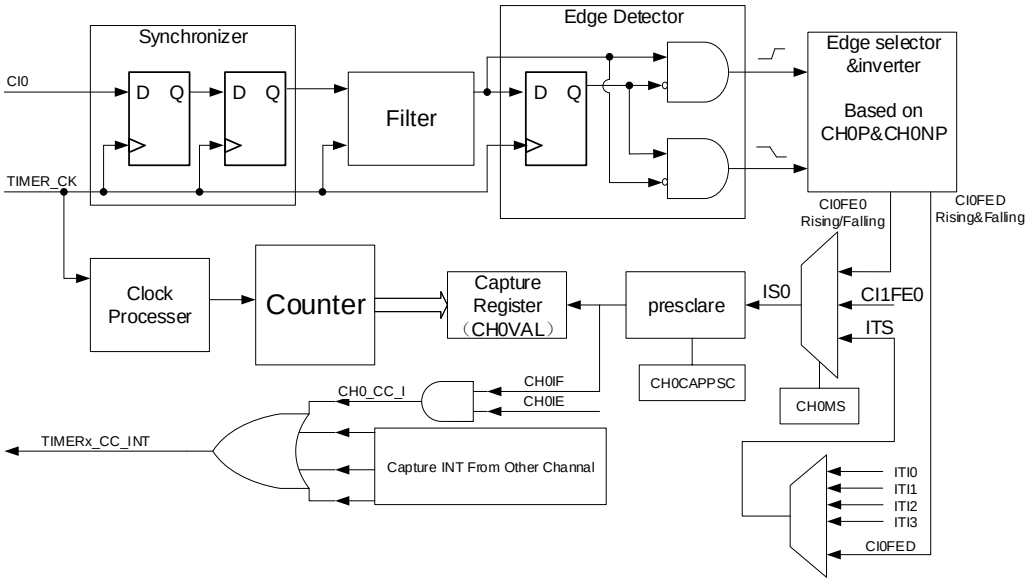
Input capture and output compare channels

The advanced 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 enabled by CHxIE = 1.

Figure 15-12. Channel input capture principle



One of channels' input signals (CIx) can be chosen from the TIMERx_CHx signal or the Excusive-OR function of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals. First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and falling edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So the process can be divided to 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 request or interrupt 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 connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

- **Channel output compare function**

Figure 15-13. Channel output compare principle (with complementary output, x=0, 1, 2)

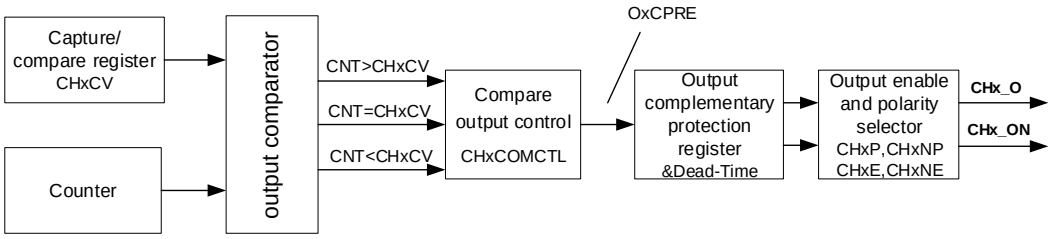


Figure 15-14. Channel output compare principle (CH3_O)

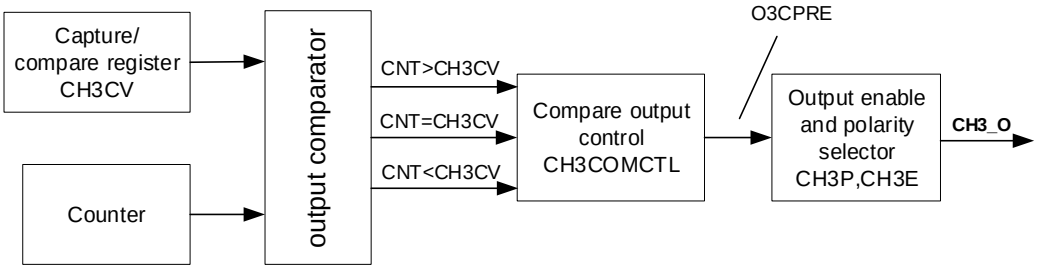


Figure 15-13. Channel output compare principle (with complementary output, x=0, 1, 2) and *Figure 15-14. Channel output compare principle (CH3_O)* show the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O/CHx_ON and the OxCPRE signal is described as below: The active level of OxCPRE is high, the output level of CHx_O/CHx_ON depends on OxCPRE signal, CHxP/CHxNP bit and CHxE/CHxNE bit (please refer to the TIMERx_CHCTL2 register for more details). For examples:

1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxE=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 CHxNP=0 (the active level of CHx_ON is low, contrary to OxCPRE), CHxNE=1 (the output of CHx_ON is enabled).
If the output of OxCPRE is active(high) level, the output of CHx_O is active(low) level;
If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(high) level.

When CH0_O and CH0_ON are output at the same time, the specific outputs of CH0_O and CH0_ON are related to the relevant bits (ROS, IOS, POE and DTCFG bits) in the TIMERx_CCHP register.

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 CxCDE=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/CHxNP.
- Enable the output by CHxEN.

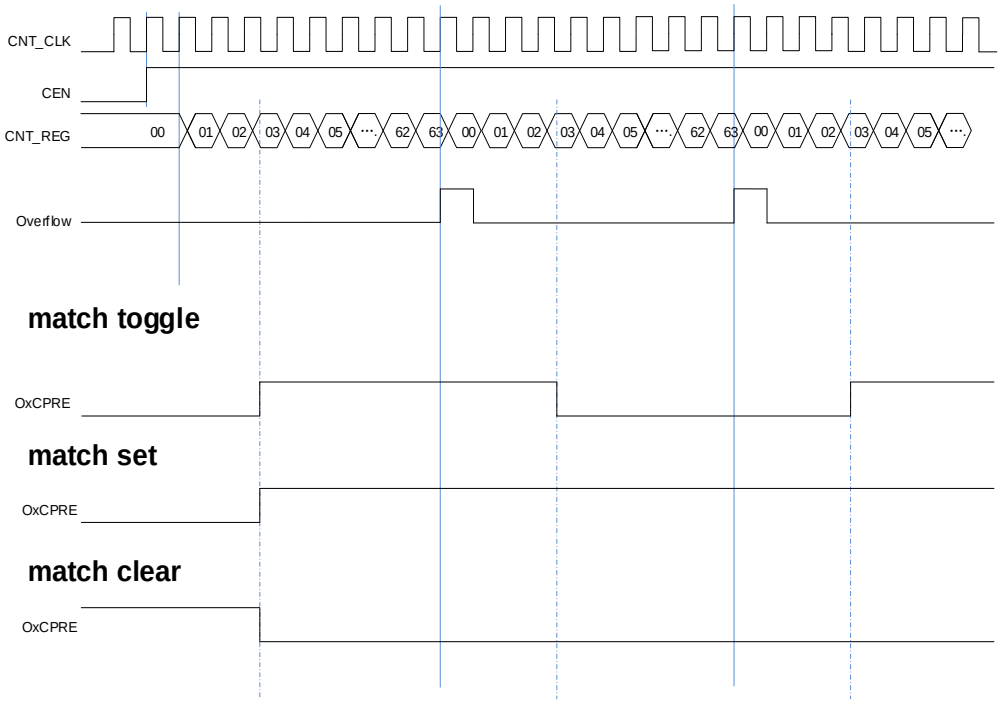
Step3: Interrupt/DMA request enable configuration by CHxIE/CxCDE.

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 15-15. Output-compare under three modes shows the three compare modes: toggle/set/clear. CAR=0x63, CHxVAL=0x3.

Figure 15-15. Output-compare under three modes



Output PWM function

In the output PWM function (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, we can also divide PWM into EAPWM (Edge aligned PWM) and CAPWM (Centre aligned PWM).

The EAPWM period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. ***Figure 15-16. EAPWM timechart*** shows the EAPWM output and interrupts waveform.

The CAPWM period is determined by 2*TIMERx_CAR, and duty cycle is by 2*TIMERx_CHxCV. ***Figure 15-17. CAPWM timechart*** shows the CAPWM output and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).
And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3'b110).

Figure 15-16. EAPWM timechart

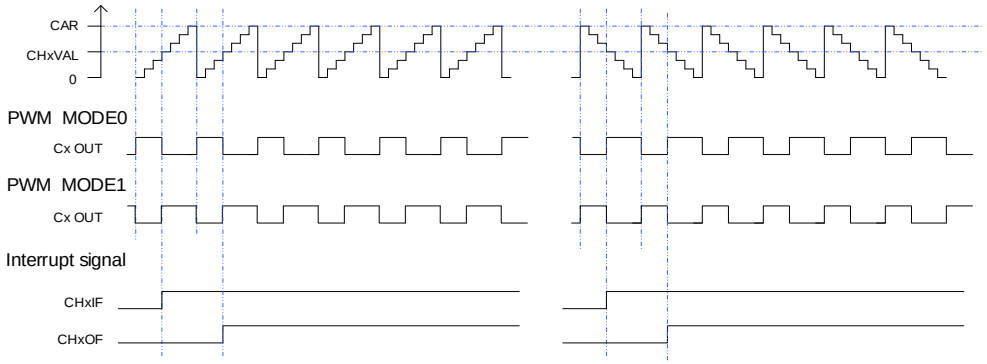
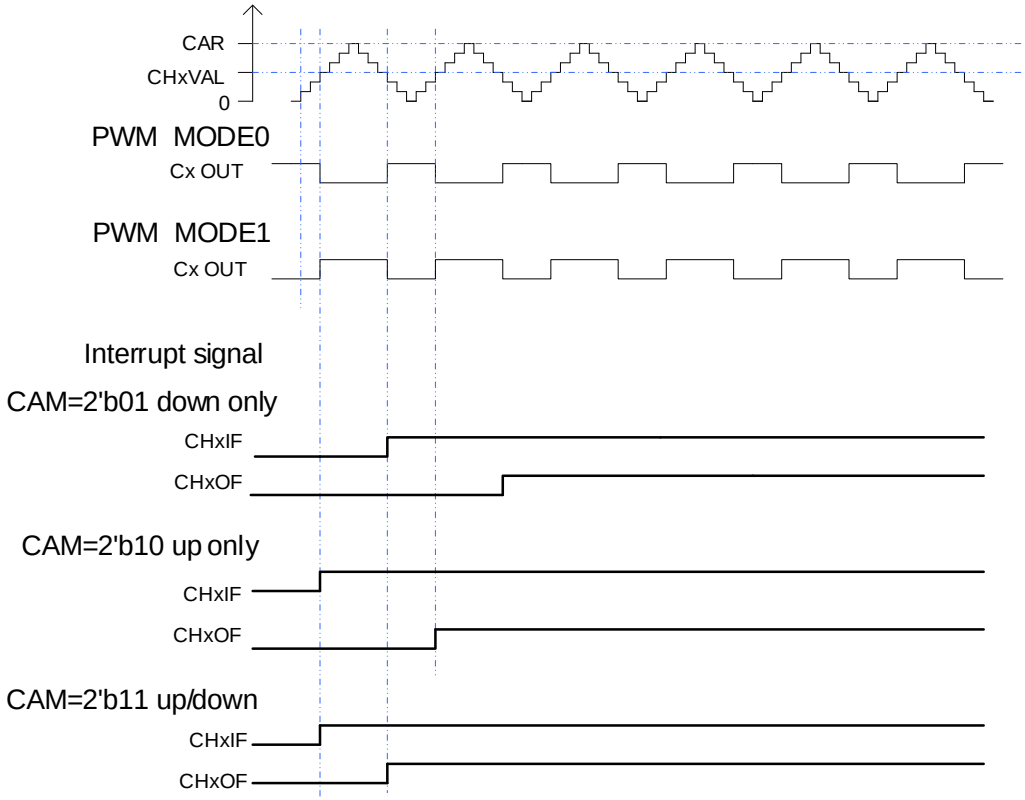


Figure 15-17. CAPWM timechart



Channel output prepare signal

As is shown in ***Figure 15-13. Channel output compare principle (with complementary output, x=0, 1, 2),*** when the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL filed. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting 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. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

The OxCPRE signal can be forced to 0 when the ETIFE signal is derived from the external ETI pin and when it is set to a high level by setting the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register. The OxCPRE signal will not return to its active level until the next update event occurs.

Channel output complementary PWM

Function of complementary is for a pair of CHx_O and CHx_ON. Those two output signals cannot be active at the same time. The TIMERx has 4 channels, but only the first three channels have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register and the POEN, ROS, IOS, ISOx and ISOxN bits in the TIMERx_CCHP and TIMERx_CTL1 registers. The outputs polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 15-2. Complementary outputs controlled by parameters

<table>
<tr><th colspan="5">Complementary Parameters</th><th colspan="2">Output Status</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON output disable(1).</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">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)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">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.</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="6">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O output disable.</td><td>CHx_ON =OxCPRE⊕(4)CHxNP
CHx_ON output enable.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON = LOW
CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON =(!OxCPRE)(5)⊕CHxNP.
CHx_ON output enable.</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O output “off-state”.</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O output “off-state”</td><td>CHx_ON =OxCPRE⊕CHxNP
CHx_ON output enable</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON =(!OxCPRE)⊕CHxNP
CHx_ON output enable.</td></tr>
</table>

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 both CHxEN and CHxNEN are 1'b1, and set POEN is also necessary. The field named DTCFG defines the dead time delay that can be used for all channels expect for channel 3. The detail about the delay time, refer to the register TIMERx_CCHP.

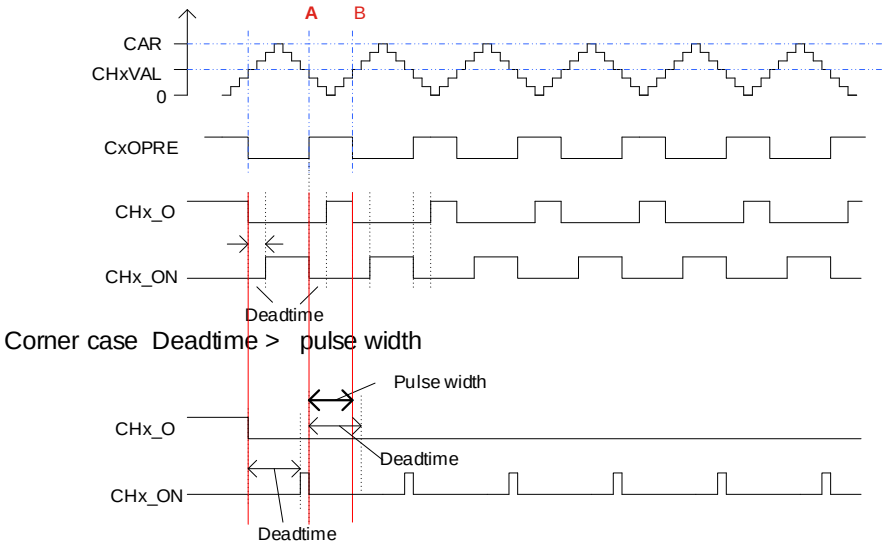
The dead time delay insertion ensures that no two complementary signals drive the active state at the same time.

When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the ***Figure 15-18. Channel output complementary PWM with dead-time insertion*** CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at once. Similarly, At point B when counter match (counter = CHxVAL) occurs again, OxCPRE is cleared, CHx_O signal will be cleared at once, while CHx_ON signal remains at the low value 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 CHx_O duty cycle, then the CHx_O signal is always the inactive value. (as show in the ***Figure 15-18. Channel output complementary PWM with dead-time insertion)***
- The dead time delay is greater than or equal to the CHx_ON duty cycle, then the CHx_ON signal is always the inactive value.

Figure 15-18. Channel output complementary PWM with dead-time insertion



Break mode

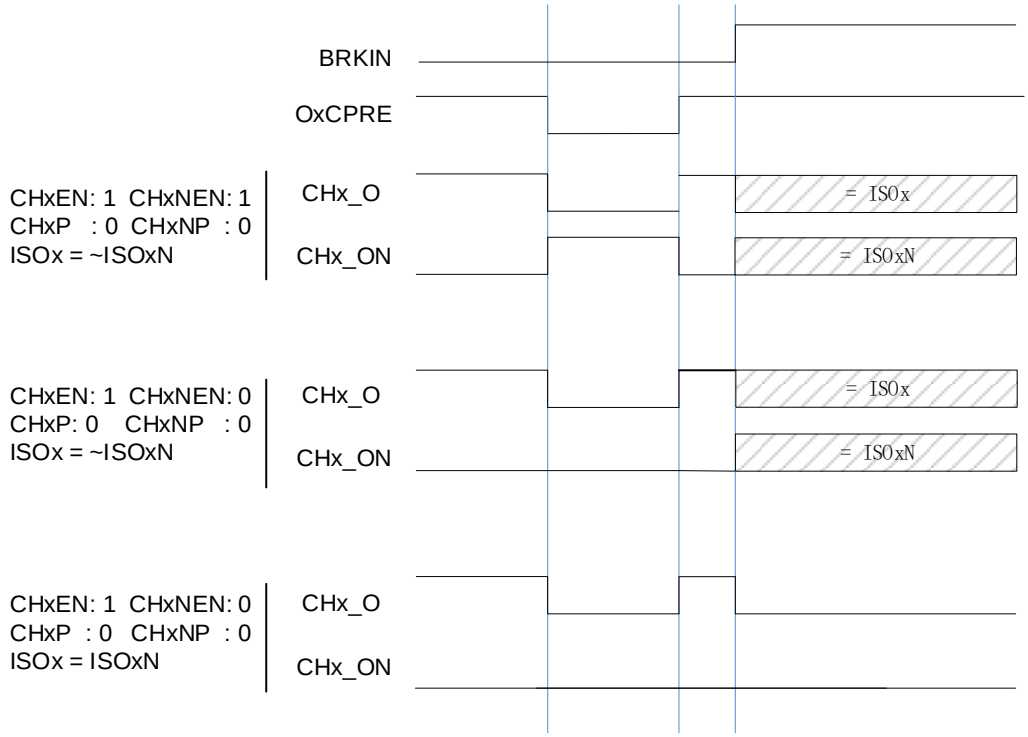
In this mode, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot

be set both to active level when break occurs. The break sources are input break pin and HXTAL stuck event by Clock Monitor (CKM) in RCU. The break function enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is setting by the BRKP bit in TIMERx_CCHP.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order 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 is set. If BRKIE is 1, an interrupt generated.

Figure 15-19. Output behavior in response to a break(The break high active)



Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0 and CI1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact to generate the counter value. The DIR bit is modified during each input source transition. The input source can be either CI0 only, CI1 only or both CI0 and CI1, the selection mode by setting the SMC [2:0] to 0x01, 0x02 or 0x03. The mechanism for changing the counter direction is shown in **Table 15-3. Counting direction versus quadrature decode signals***Table 15-3. Counting direction versus quadrature decode signals.* The quadrature decoder can be regarded as an external clock with a directional selection. This means that the counter counts continuously in

the interval between 0 and the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 15-3. Counting direction versus quadrature decode signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
CI0 only counting	CI1FE1=High	Down	Up	-	-
	CI1FE1=Low	Up	Down	-	-
CI1 only counting	CI0FE0=High	-	-	Up	Down
	CI0FE0=Low	-	-	Down	Up
CI0 and CI1 counting	CI1FE1=High	Down	Up	X	X
	CI1FE1=Low	Up	Down	X	X
	CI0FE0=High	X	X	Up	Down
	CI0FE0=Low	X	X	Down	Up

Note:"-" means "no counting"; "X" means impossible.

Figure 15-20. Example of counter operation in quadrature decoder interface mode

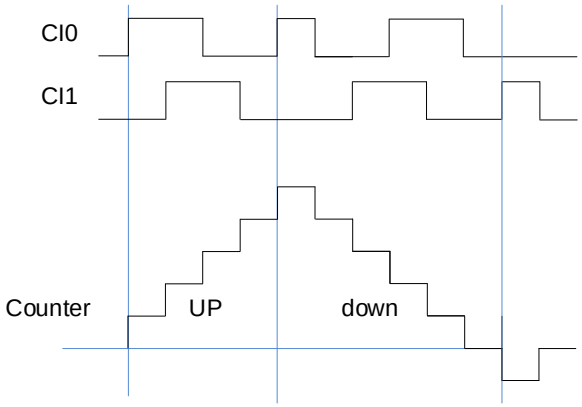
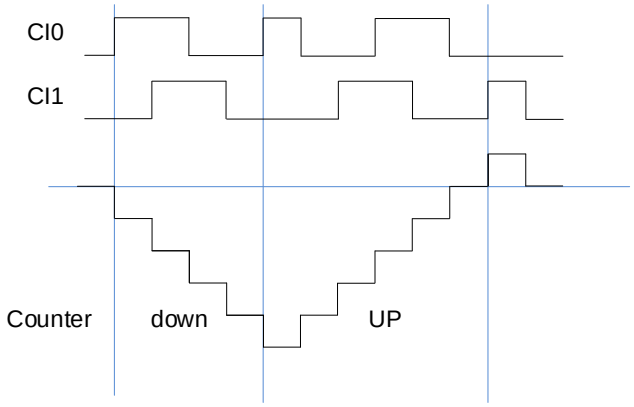


Figure 15-21. Example of quadrature decoder interface mode with CI0FE0 polarity inverted



Hall sensor function

Hall sensor is generally used to control BLDC Motor; advanced timer can support this function.

Figure 15-22. Hall sensor is used to BLDC motor show how to connect. And we can see we need two timers. First TIMER_in(Advanced/GeneralL0 TIMER) should accept three Rotor Position signals from Motor.

Each of the 3 sensors provides a pulse that applied to an input capture pin, can then be analyzed and both speed and position can be deduced.

By the internal connection such as TRGO-ITIx, TIMER_in and TIMER_out can be connected. TIMER_out will generate PWM signal to control BLDC motor's speed based on the ITRx. Then, the feedback circuit is finished, also you change configuration to fit your request.

About the TIMER_in, it need have input XOR function, so you can choose from Advanced/GeneralL0 TIMER.

And TIMER_out need have functions of complementary and Dead-time, so only advanced timer can be chosen. Else, based on the timers' internal connection relationship, pair's timers can be selected. For example:

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)
TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

And so on.

After getting appropriate timers combination, and wire connection, we need to configure timers. Some key settings include:

- Enable XOR by setting TI0S, then, each of input signal change will make the CI0 toggle. CH0VAL will record the value of counter at that moment.
- Enable ITIx connected to commutation function directly by setting CCUC and CCSE.
- Configuration PWM parameter based on your request.

Figure 15-22. Hall sensor is used to BLDC motor

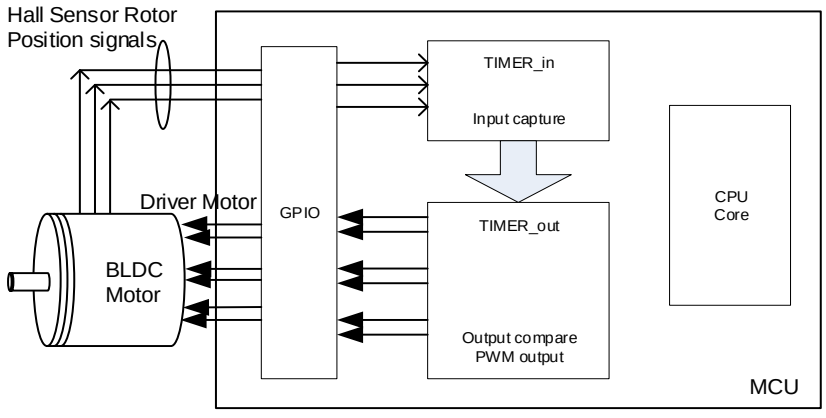
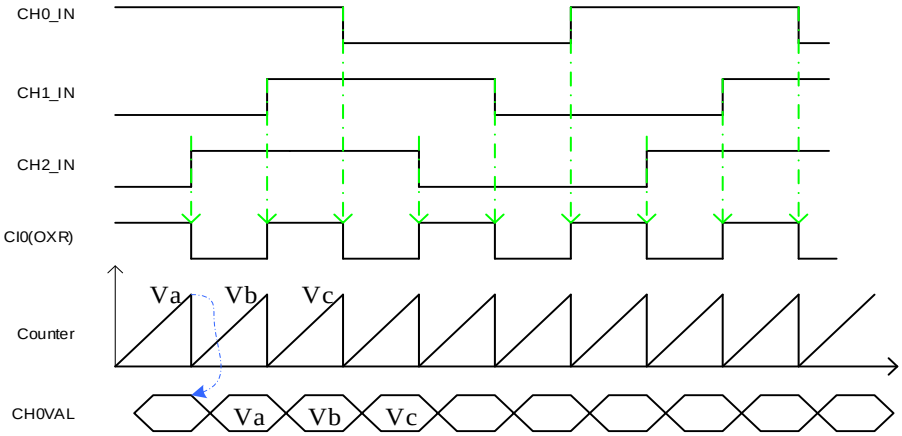
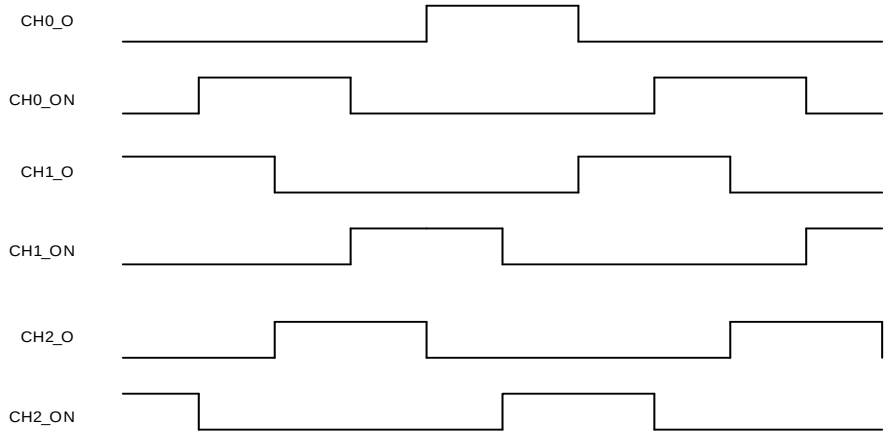


Figure 15-23. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



Advanced TIMER_out under output compare mode(PWM with Dead-time)

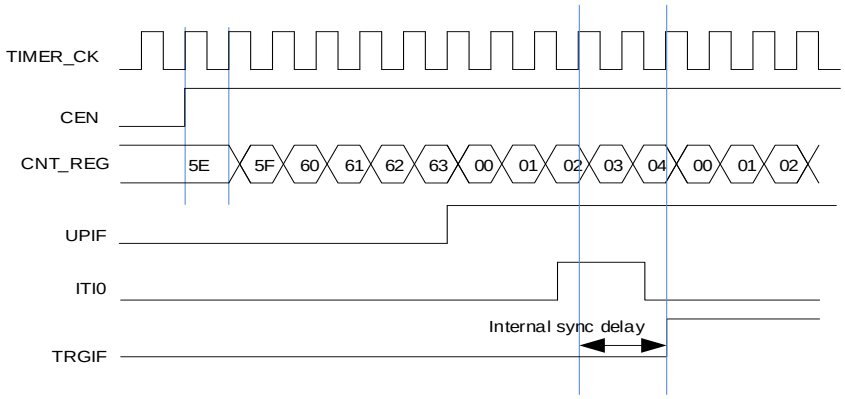
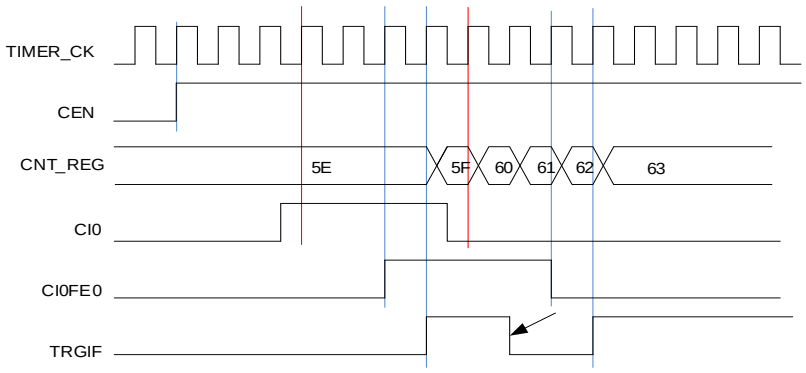


Master-slave management

The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode which is selected by the SMC [2:0] in the TIMERx_SMCFG register. The trigger input of these modes can be selected by the TRGS [2:0] in the TIMERx_SMCFG register.

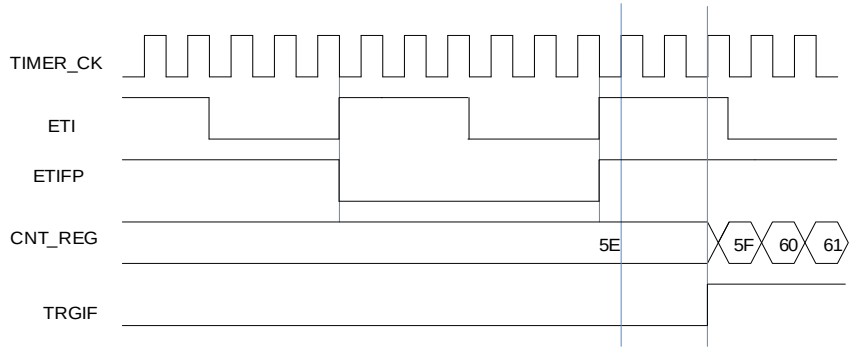
Table 15-4. Slave mode example table

	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 110: CI1FE1	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion. If ETIFP is selected as the trigger source,	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

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
		111: ETIFP	configure the ETP for polarity selection and inversion.	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[2:0] = 3'b000 ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 15-24. Restart mode 			
Exam2	**Pause mode** The counter will be paused when the trigger input is low, and it will start when the trigger input is high.	TRGS[2:0]=3'b101 CI0FE0 is selected.	TI0S=0 (Non-xor) [CH0NP=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.	Filter is bypassed in this example.
	Figure 15-25. Pause mode 			

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
Exam3	**Event mode** The counter will start to count when a rising edge of trigger input comes.	TRGS[2:0] =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 15-26. Event mode



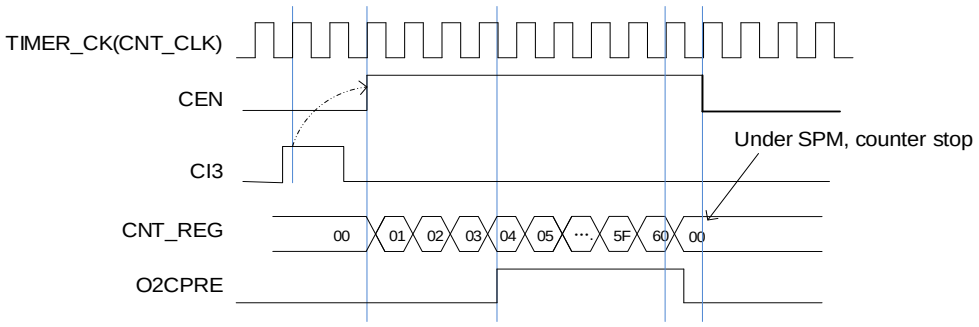
Single pulse mode

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 operate in the single pulse mode, it is not necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the trigger signals edge or by setting the CEN bit to 1 using software. Setting the CEN bit to 1 or a trigger from the trigger signals 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 using software, the counter will be stopped and its value held.

In the single pulse mode, the trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0/1 register. 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. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 15-27. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timers interconnection

Timer can be configured as interconnection, that is, one timer which operate in the master mode outputs TRGO signal to control another timer which operate in the slave mode, TRGO include reset evevt, start evevt, update evevt, capture/compare pulse evevt, compare evevt. slave timer received the ITIx and performs the corresponding mode, include internal clock mode, quadrature decoder mode, restart mode, pause mode, event mode, external clock mode.

Figure 15-28. TIMER0 Master/Slave mode timer example shows the timer0 trigger selection when it is configured in slave mode.

Figure 15-28. TIMER0 Master/Slave mode timer example

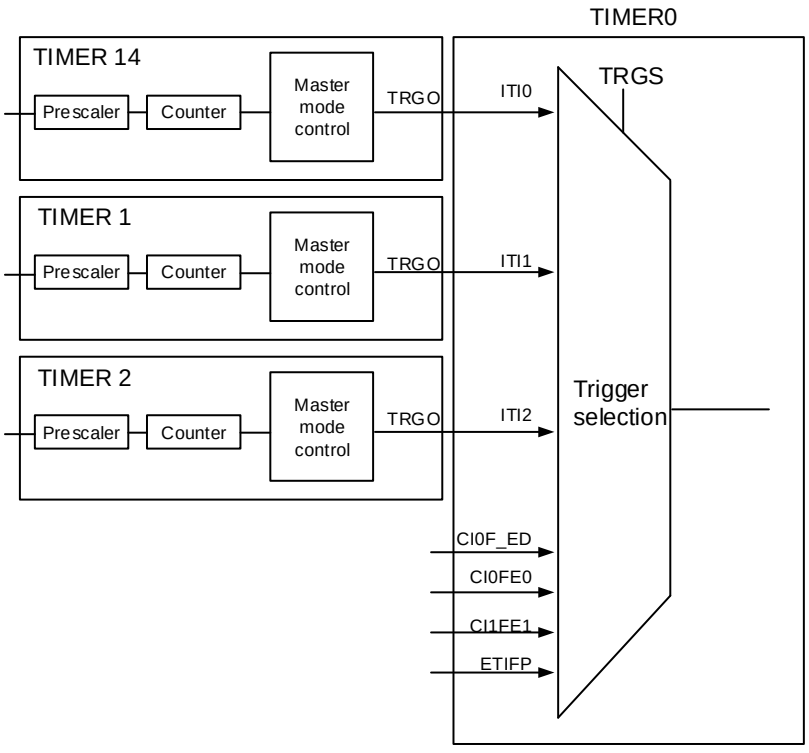


Table 15-5. Input trigger of Timer0 shows the input trigger.

Table 15-5. Input trigger of Timer0

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER0	TIMER14	TIMER1	TIMER2	reserved

Note: '-' means no interconnection.

Other interconnection examples:

TIMER2 as prescaler for TIMER0

We configure TIMER2 as a prescaler for TIMER0. Refer to *Figure 15-28. TIMER0 Master/Slave mode timer example* for connections. Do as follow:

1. Configure TIMER2 in master mode and select its Update Event (UPE) as trigger output (MMC=010 in the TIMER2_CTL1 register). Then TIMER2 drives a periodic signal on each counter overflow.
2. Configure the TIMER2 period (TIMER2_CAR registers).
3. Select the TIMER0 input trigger source from TIMER2 (TRGS=010 in the TIMER0_SMCFG register).
4. Configure TIMER0 in external clock mode 1 (SMC=111 in TIMER0_SMCFG register).
5. Start TIMER0 by writing '1 in the CEN bit (TIMER0_CTL0 register).
6. Start TIMER2 by writing '1 in the CEN bit (TIMER2_CTL0 register).

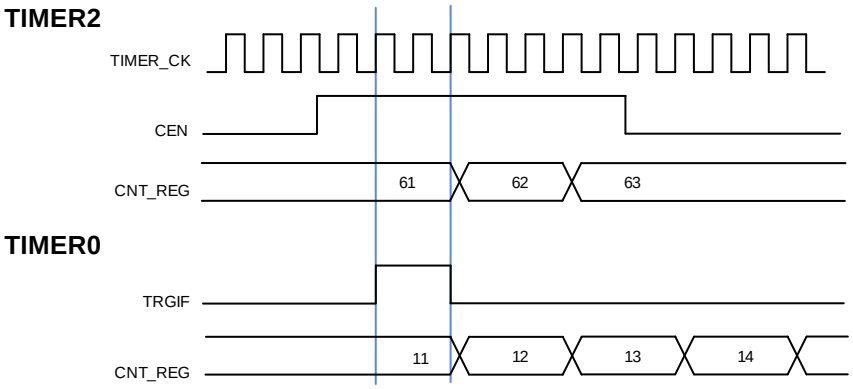
Start TIMER0 with TIMER2's Enable/Update signal

First, we enable TIMER0 with the enable out of TIMER2. Refer to *Figure 15-29. Triggering TIMER0 with Enable of TIMER2* TIMER0 starts counting from its current value on the divided internal clock after trigger by TIMER2 enable output.

When TIMER0 receives the trigger signal its CEN bit is set and the counter counts until we disable TIMER0. Both counter clock frequencies are divided by 3 by the prescaler compared to TIMER_CK ($f_{CNT_CLK} = f_{TIMER_CK}$ /3). Do as follow:

1. Configure TIMER2 master mode to send its enable signal as trigger output(MMC=001 in the TIMER2_CTL1 register)
2. Configure TIMER0 to select the input trigger from TIMER2 (TRGS=010 in the TIMER0_SMCFG register).
3. Configure TIMER0 in event mode (SMC=110 in TIMER0_SMCFG register).
4. Start TIMER2 by writing 1 in the CEN bit (TIMER2_CTL0 register).

Figure 15-29. Triggering TIMER0 with Enable of TIMER2



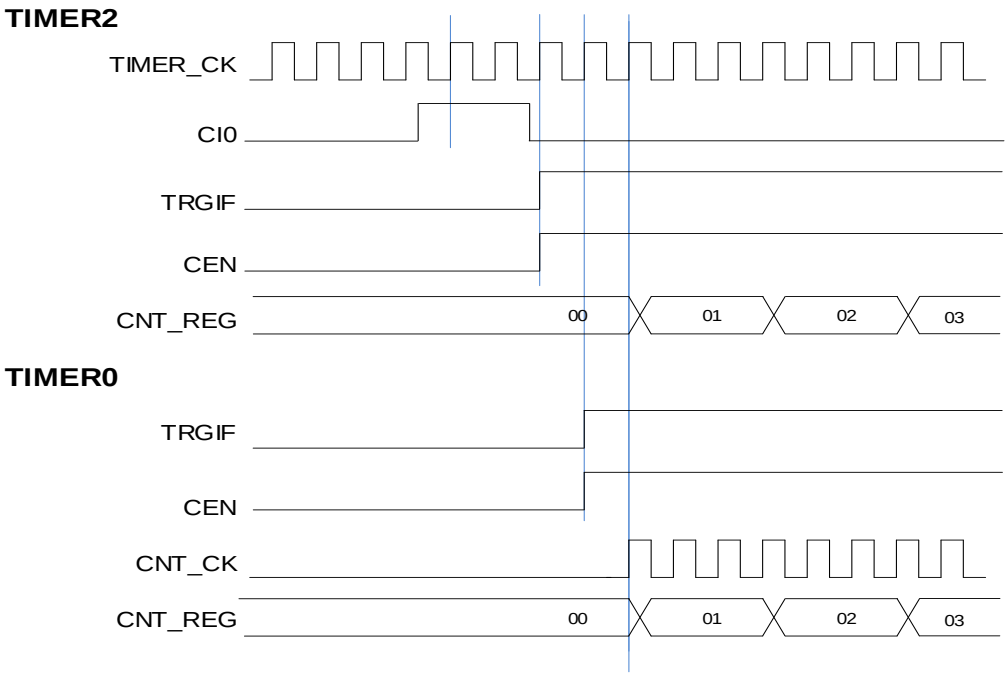
Using an external trigger to start 2 timers synchronously

We configure the start of TIMER0 is triggered by the enable of TIMER2, and TIMER2 is triggered by its CI0 input rises edge. To ensure 2 timers start synchronously, TIMER2 must be configured in Master/Slave mode. Do as follow:

1. Configure TIMER2 slave mode to get the input trigger from CI0 (TRGS=100 in the TIMER2_SMCFG register).
2. Configure TIMER2 in event mode (SMC=110 in the TIMER2_SMCFG register).
3. Configure the TIMER2 in Master/Slave mode by writing MSM=1 (TIMER2_SMCFG register).
4. Configure TIMER0 to get the input trigger from TIMER2 (TRGS=010 in the TIMER0_SMCFG register).
5. Configure TIMER0 in event mode (SMC=110 in the TIMER0_SMCFG register).

When a rising edge occurs on TIMER2's CI0, two timer counters starts counting synchronously on the internal clock and both TRGIF flags are set.

Figure 15-30. Triggering TIMER0 and TIMER2 with TIMER2'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®-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

15.1.5. TIMERx registers(x=0)

TIMER0 base address: 0x4001 2C00

Control register 0 (TIMERx_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						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 aligns 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 under center-aligned 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 under center-aligned 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 under center-aligned 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, 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 quadrature decoder mode, this bit is read only.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. 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 update interrupts or DMA requests: 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, pause mode and quadrature 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	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

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value
14	ISO3	Idle state of channel 3 output Refer to ISO0 bit
13	ISO2N	Idle state of 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 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 channel 0 complementary output 0: When POEN bit is reset, CH0_ON is set low. 1: When POEN bit is reset, CH0_ON 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 CH0_ON 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, CH1 and 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 resert source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a conter 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 enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below: 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by 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, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event coming. 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
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]	OCRC	SMC[2:0]
rw	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 rising edge or high level . 1: ETI is active at falling edge or low level .
14	SMC1	Part of SMC for enable External clock mode1 In external clock mode 1, the counter is clocked by any active edge on the ETIF 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 clock source of the timer will be ETIFP if external clock mode 0 and external clock mode 1 are configured at the same time. **Note:** External clock mode 0 enable is in this register's SMC[2:0] bit-filed.
13:12	ETPSC[1:0]	The prescaler of external trigger The frequency of external trigger signal ETIFP must not be at higher than 1/4 of TIMER_CK frequency. When the external trigger signal is a fast clock, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disable. 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 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 disable

1: Master-slave mode enable

6:4 TRGS[2:0]

Trigger selection

This bit-field specifies which signal is selected as the trigger input, which is used to synchronize the counter.

000: ITI0

001: ITI1

010: ITI2

011: ITI3

100: CI0F_ED

101: CI0FE0

110: CI1FE1

111: ETIFP

These bits must not be changed when slave mode is enabled.

3 OCRC

OCPRE clear source selection

0: OCPRE_CLR_INT is connected to the OCPRE_CLR input

1: OCPRE_CLR_INT is connected to ETIF

2:0 SMC[2:0]

Slave mode control

000: Disable 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 CI1FE1 edge, while the direction depends on CI0FE0 level.

010: Quadrature decoder mode 1.The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

011: Quadrature decoder mode 2.The counter counts on both CI0FE0 and CI1FE1 edge, while the direction depends on each other.

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.

Because CI0F_ED outputs 1 pulse for each transition on CI0F, and the pause mode checks the level of the trigger signal, when CI0F_ED is selected as the trigger input, the pause mode must not be used.

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	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
31:15	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
Reserved															

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
31:13	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 channel's commutation event occurs, and cleared by software 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4	CH3IF	Channel 3 's capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 's capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 's capture/compare interrupt flag

		Refer to CH0IF description
1	CH0IF	Channel 0 ‘s capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When 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 on an update event 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								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	BRKG	Break event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is 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 is 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, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1).
0: No affect
1: Generate channel's c/c control update event

4 CH3G Channel 3's capture or compare event generation
Refer to CH0G description

3 CH2G Channel 2's capture or compare event generation
Refer to CH0G description

2 CH1G Channel 1's capture or compare event generation
Refer to CH0G description

1 CH0G Channel 0's capture or compare event generation
This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high.
0: No generate a channel 1 capture or compare event
1: Generate a channel 1 capture or compare event

0 UPG Update event generation
This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) 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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	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
31:16	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	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
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 mode 01: Channel 1 is programmed as input mode, IS1 is connected to CI1FE1 10: Channel 1 is programmed as input mode, IS1 is connected to CI0FE1 11: Channel 1 is programmed as input mode, IS1 is connected to ITS. **Note:** When CH1MS[1:0]=11, it is necessary to select an internal trigger input 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 disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. 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 mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(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 disable 1: Channel 0 output compare shadow enable 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
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 mode 01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0 10: Channel 0 is programmed as input mode, IS0 is connected to CI1FE0 11: Channel 0 is programmed as input mode, IS0 is connected to ITS **Note:** When CH0MS[1:0]=11, it is necessary to select an internal trigger input 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:

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 clear. 00: Prescaler disable, 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
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	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	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMFEN description
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 mode 01: Channel 3 is programmed as input mode, IS3 is connected to CI3FE3 10: Channel 3 is programmed as input mode, IS3 is connected to CI2FE3 11: Channel 3 is programmed as input mode, IS3 is connected to ITS. **Note:** When CH3MS[1:0]=11, it is necessary to ensure that 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 disable 1: Channel 2 output compare clear enable
6:4	CH2COMCTL[2:0]	Channel 2 compare output control

		This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O2CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH2CV. 010: Clear the channel output. 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 high when the counter is smaller than TIMERx_CH2CV, and low otherwise. When counting down, O2CPRE is low when the counter is larger than TIMERx_CH2CV, and high otherwise. 111: PWM mode 1. When counting up, O2CPRE is low when the counter is smaller than TIMERx_CH2CV, and high otherwise. When counting down, O2CPRE is high when the counter is larger than TIMERx_CH2CV, and low otherwise. 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-filed in TIMERx_CCHP register is 11 and CH2MS bit-filed is 00(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 disable 1: Channel 2 output compare shadow enable 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
2	CH2COMFEN	Channel 2 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM1 or PWM2 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the result of the comparison. 0: Channel 2 output quickly compare disable. 1: Channel 2 output quickly compare enable.
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 mode

01: Channel 2 is programmed as input mode, IS2 is connected to CI2FE2

10: Channel 2 is programmed as input mode, IS2 is connected to CI3FE2

11: Channel 2 is programmed as input mode, IS2 is connected to ITS.

Note: When CH2MS[1:0]=11, it is necessary to select an internal trigger input 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 clear. 00: Prescaler disable, 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 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		CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:14	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 output polarity Refer to CH0NP description
10	CH2NEN	Channel 2 complementary output enable Refer to CH0NEN description
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 output polarity

		Refer to CH0NP description
6	CH1NEN	Channel 1 complementary output enable Refer to CH0NEN description
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 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 complementary output high level is active level 1: Channel 0 complementary output low level is active level When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of input signal. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled 1: Channel 0 complementary output enabled
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 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed 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-filed indicates the current counter value. Writing to this bit-filed 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-filed 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-filed specifies the auto reload value of the counter.

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-filed specifies the update event generation rate. Each time the repetition counter counting down to zero, an update event is generated. The update rate of the shadow registers is also affected by this bit-filed 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 or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN 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: The 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-filed in TIMERx_CCHP register is 00.
13	BRKP	Break polarity This bit specifies the polarity of the BRKIN input signal. 0: BRKIN input active low 1; BRKIN input active high
12	BRKEN	Break enable This bit can be set to enable the BRKIN and CCS clock failure event inputs. 0: Break inputs disabled 1; Break inputs enabled This bit can be modified only when PROT [1:0] bit-filed 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 15-2. Complementary outputs controlled by parameters*** 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 15-2. Complementary outputs controlled by parameters***

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-filed specifies the write protection property of registers.

00: protect disable. No write protection.

01: PROT mode 0.The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected.

10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected.

11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN bits in TIMERx_CHCTL0/1 registers (if the related channel is configured in output) are writing protected.

This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.

7:0 DTCFG[7:0]

Dead time configure

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:

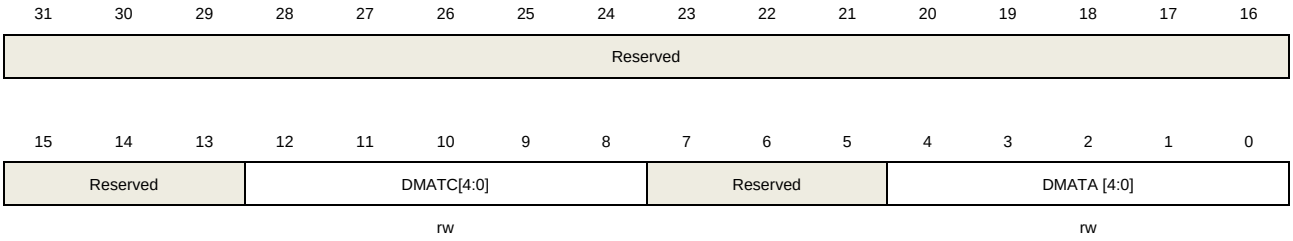
1. t_{DTS_CK} is the period of DTS_CK which is configured by CKDIV[1:0] in TIMERx_CTL0.

2. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0x48
Reset value: 0x0000 0000

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

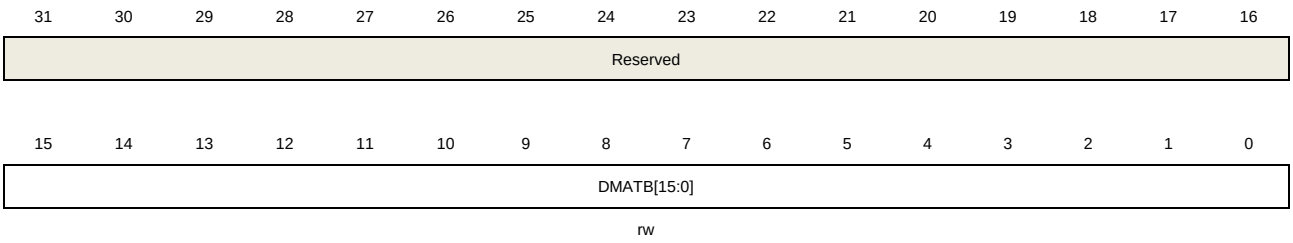


Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed is defined the number of DMA will access(R/W) the register of TIMERx_DMATB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed define 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.

DMA transfer buffer register (TIMERx_DMATB)

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
15:0	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.

15.2. General level0 timer (TIMERx, x=1, 2)

15.2.1. Overview

The general level0 timer module (TIMER1, 2) is a four-channel timer that supports input capture, output compare. They can generate PWM signals to control motor or be used for power management applications. The general level0 time reference is a 16-bit or 32-bit counter that can be used as an unsigned counter.

In addition, the general level0 timers can be programmed and be used to count or time external events that drive other timers.

Timers are completely independent, but there may be synchronized to provide a larger timer with their counters incrementing in unison.

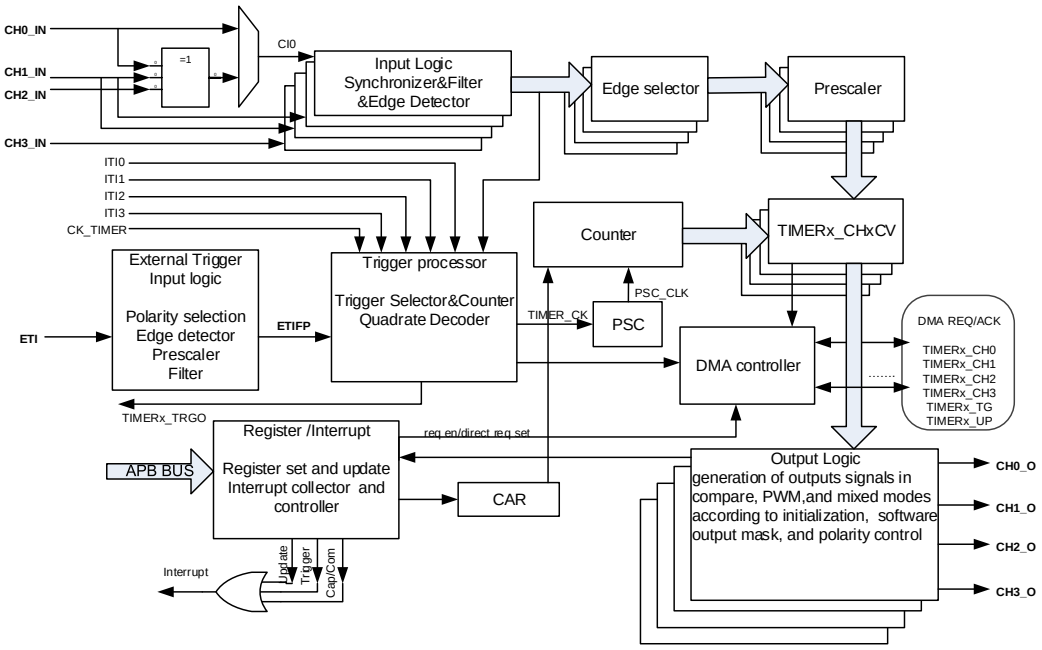
15.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16bit (TIMER2), 32bit (TIMER1).
- Source of count clock is selectable:
  internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: count up, count down, count up/down.
- Quadrature decoder: used to track motion and determine both rotation direction and position.
- Hall sensor: for 3-phase motor control.
- Programmable prescaler: 16 bit. Factor can be changed on the go.
- Each channel is user-configurable:
  Input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Auto-reload function.
- Interrupt output or DMA request on: update, trigger event, and compare/capture event.
- Daisy chaining of timer modules to allow a single timer to initiate multiple timing events.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer master-slave management.

15.2.3. Block diagram

Figure 15-31. General Level 0 timer block diagram provides details on the internal configuration of the general level0 timer.

Figure 15-31. General Level 0 timer block diagram



15.2.4. 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]).

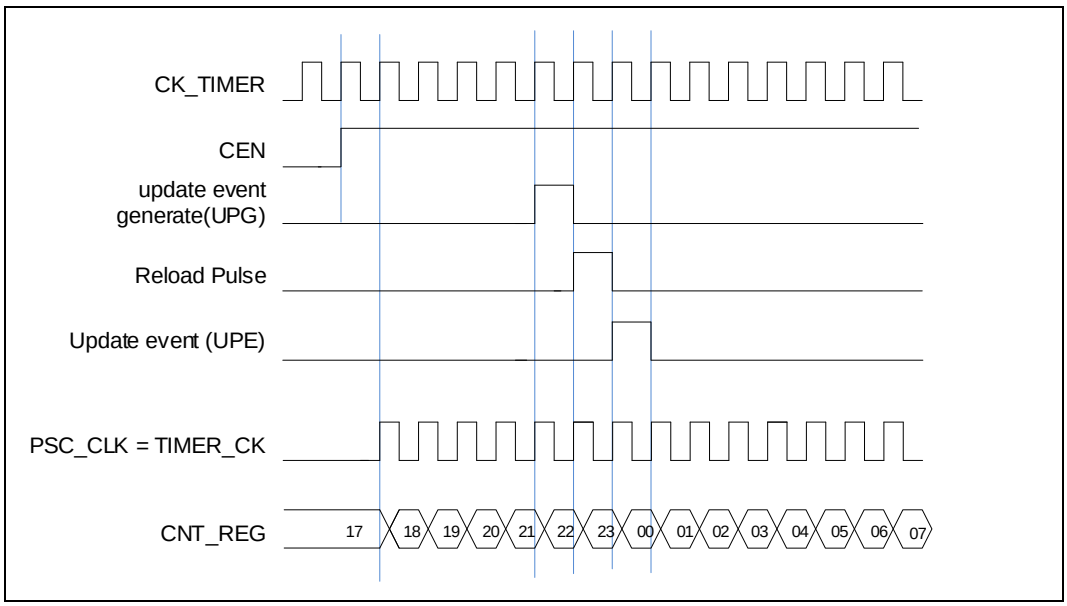
- SMC [2:0] == 3'b000. Internal timer clock CK_TIMER which is from module RCU.

The default internal clock source is the CK_TIMER used to drive 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, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU.

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 and described as follows. 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 15-32. Timing chart of internal clock divided by 1



- SMC [2:0] == 3'b111(external clock mode 0). External input pin source

The TIMER_CK, driven counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CI0/TIMERx_CI1. 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.

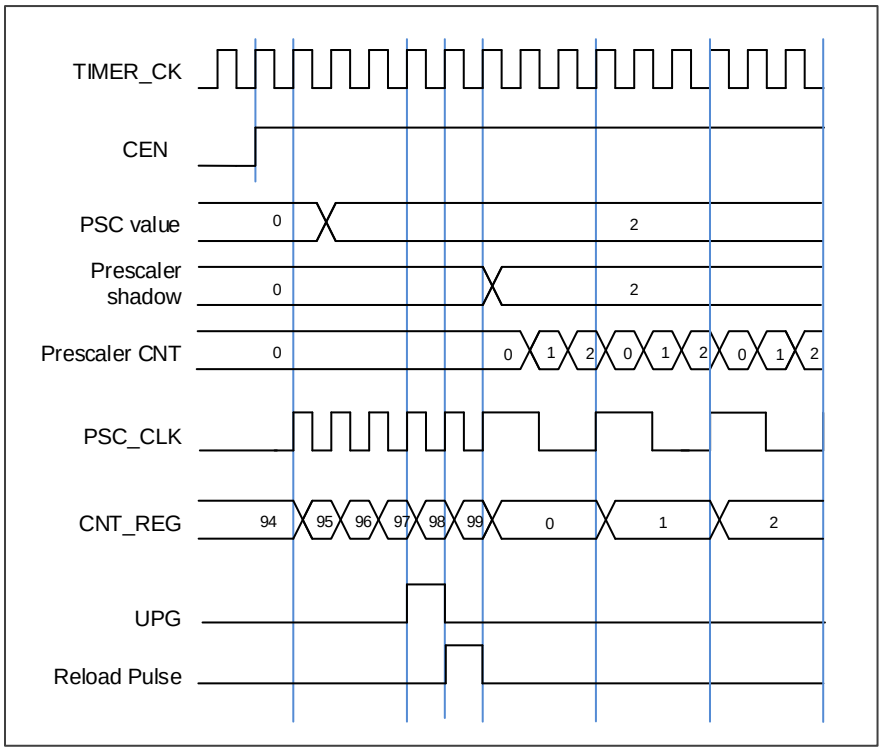
- SMC1== 1'b1(external clock mode 1). External input pin source (ETI)

The TIMER_CK, driven 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 set the SMC [2:0] to 0x7 and the TRGS [2:0] to 0x7 respectively. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the clock source is selected to come from the ETI signal, the trigger controller including the edge detection circuitry will generate a clock pulse during 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 prescale 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 15-33. 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 (counter auto reload register, prescaler register) are updated.

Figure 15-34. Timing chart of up counting mode, PSC=0/2 and ***Figure 15-35. Timing chart of up counting mode, change TIMERx_CAR on the go.*** show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 15-34. Timing chart of up counting mode, PSC=0/2

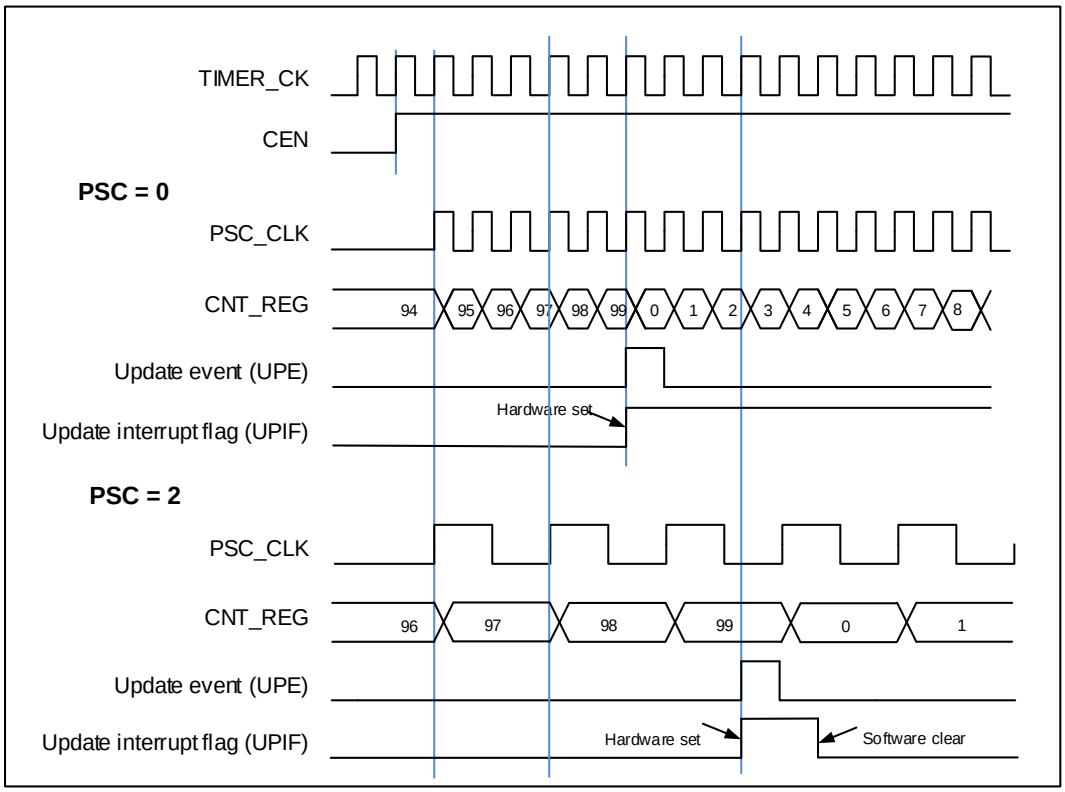
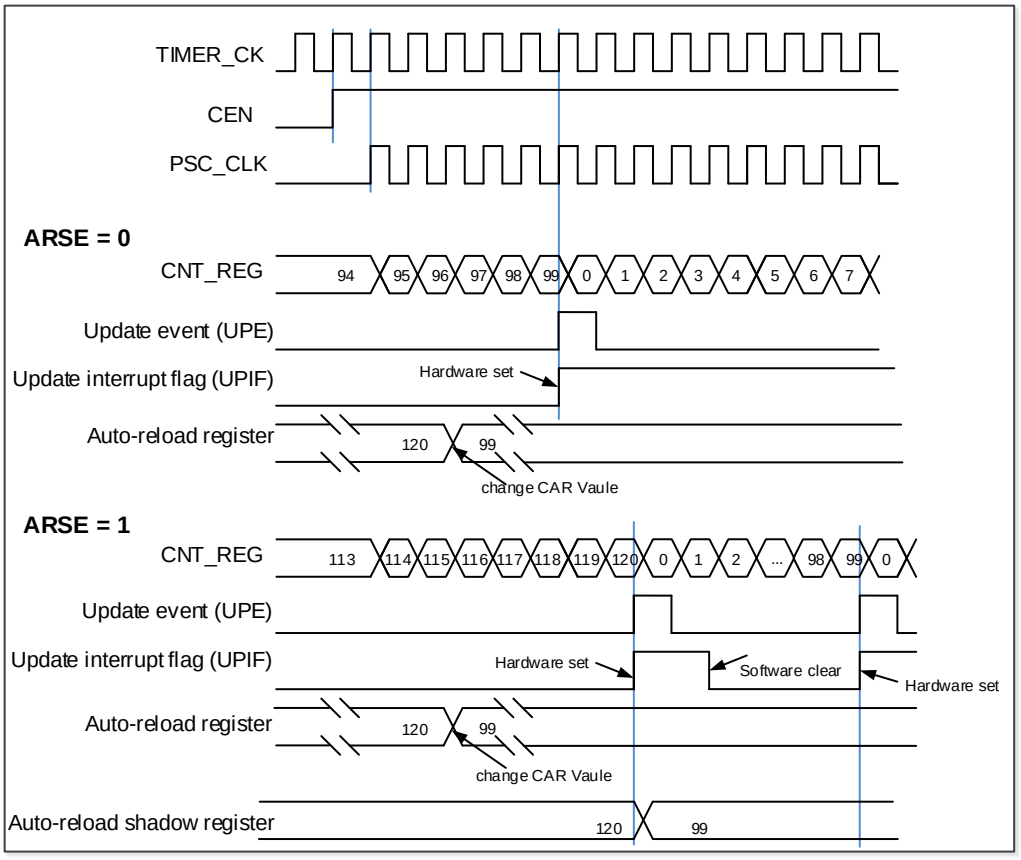


Figure 15-35. Timing chart of up counting mode, change TIMERx_CAR on the go.



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 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 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 (counter auto reload register, prescaler register) are updated.

Figure 15-36. Timing chart of down counting mode, PSC=0/2 and *Figure 15-37. 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 15-36. Timing chart of down counting mode, PSC=0/2

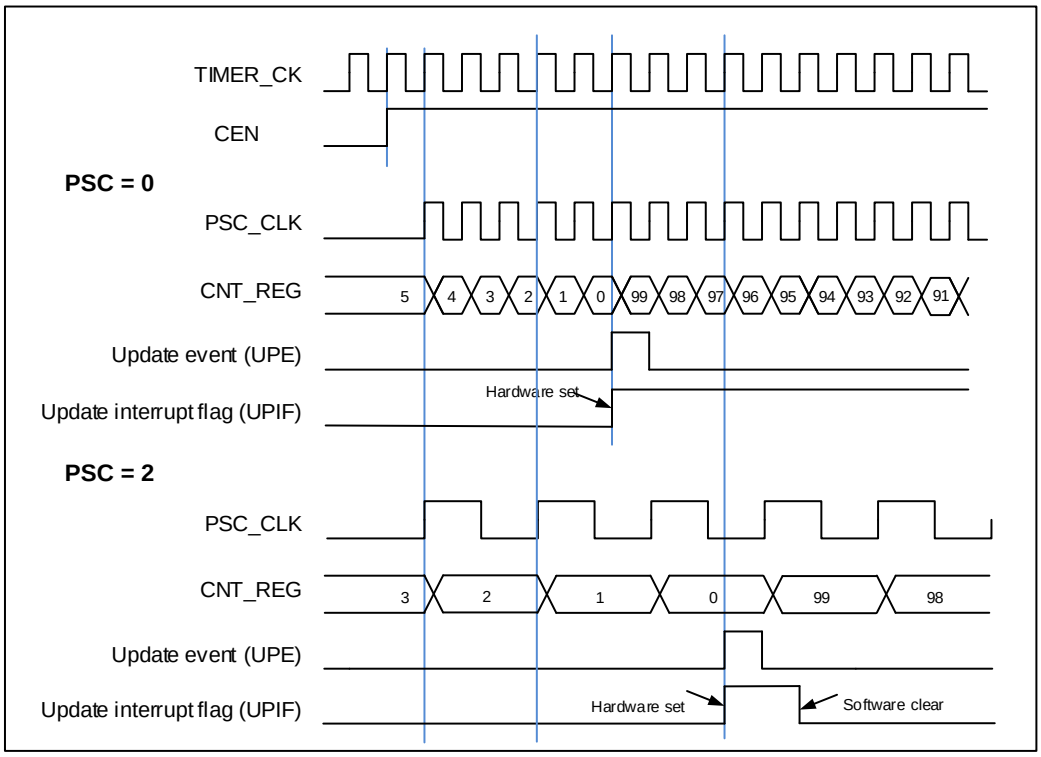
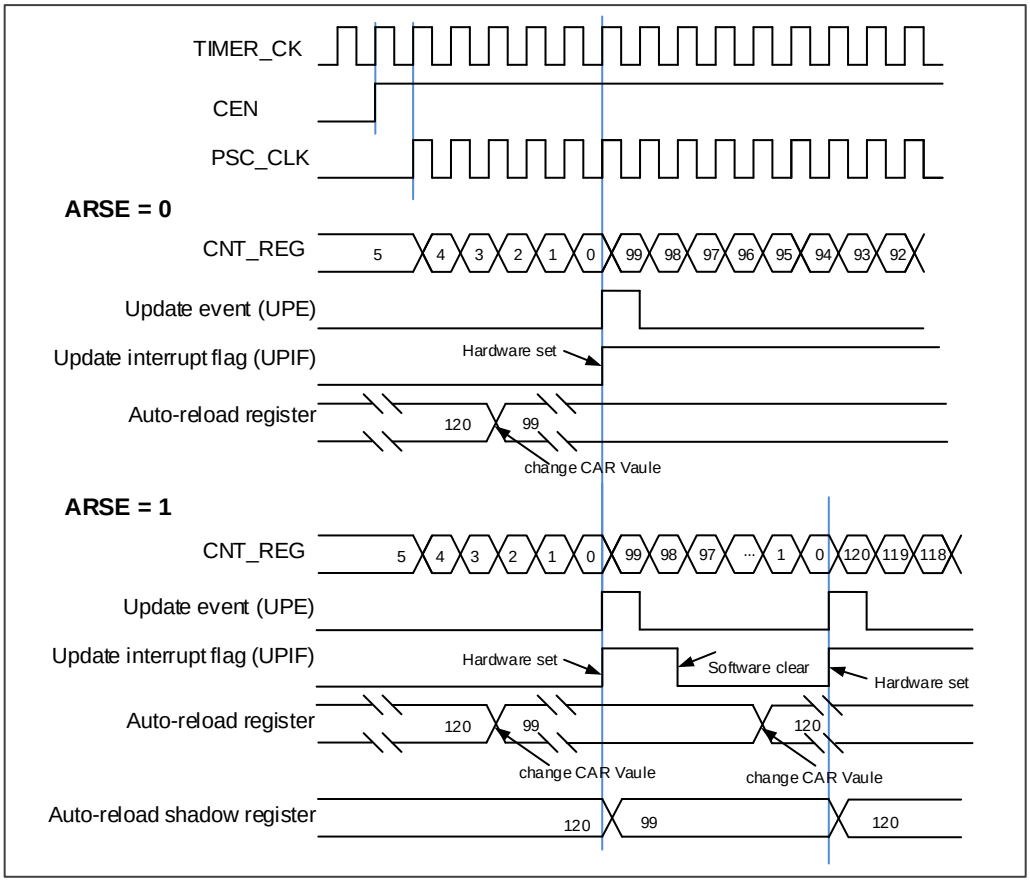


Figure 15-37. Timing chart of down counting mode, change TIMERx_CAR on the go.



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 irrespective of whether the counter is counting up or down in the center-align counting mode and generates an update event.

The UPIF bit in the TIMERx_SWEVG register can be set to 1 when an underflow event at count-down (CAM in TIMERx_CTL0 is “2’b01”), an overflow event at count-up (CAM in TIMERx_CTL0 is “2’b10”) or both of them occur (CAM in TIMERx_CTL0 is “2’b11”).

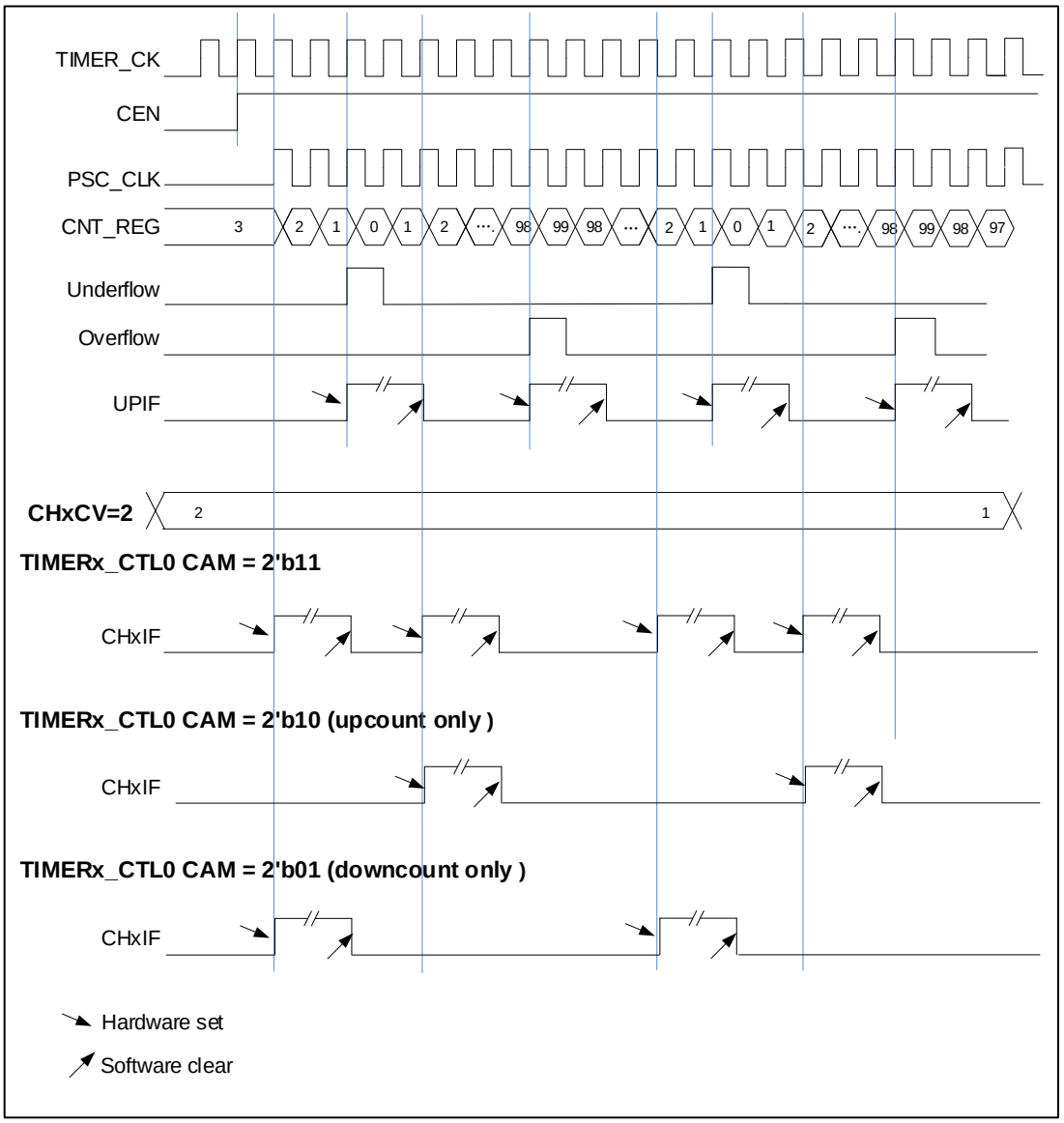
If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter autoreload register, prescaler register) are updated.

Figure 15-38. Center-aligned counter timechart show some examples of the counter

behavior for different clock frequencies when TIMERx_CAR=0x99. TIMERx_PSC=0x0

Figure 15-38. Center-aligned counter timechart



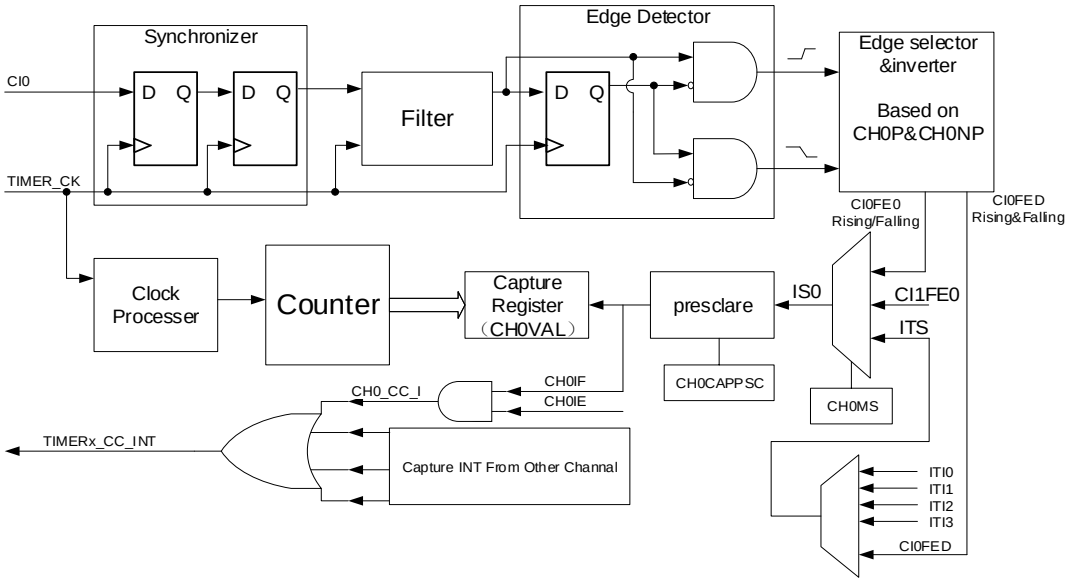
Input capture and output compare channels

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 enabled by CHxIE = 1.

Figure 15-39. Channel input capture principle



One of channels' input signals (CIx) can be chosen from the TIMERx_CHx signal or the Excusive-OR function of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals. First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and fall edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So the process can be divided to several steps as below:

Step1: Filter Configuration. (CHxCAPFLT in TIMERx_CHCTL0)

Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge Selection. (CHxP/CHxNP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source Selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel 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 enable; you can got the interrupt and DMA request.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2)

Result: When you wanted input signal is got, TIMERx_CHxCV will be set by counter's value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt and DMA request will be asserted based on the your configuration of CHxIE and CHxDEN in

TIMERx_DMAINTEN

Direct generation: If you want to generate a DMA request or interrupt, you can set CHxG by software 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 connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

- **Channel output compare function**

Figure 15-40. Channel output compare principle (x=0,1,2,3)

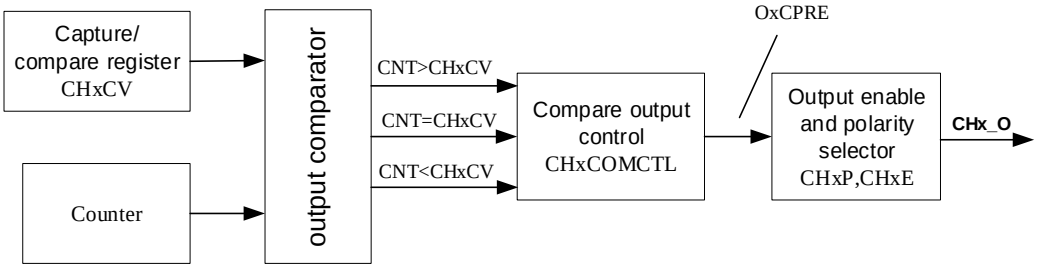


Figure 15-40. Channel output compare principle (x=0,1,2,3) shows the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O and the OxCPRE signal is described as blew: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE 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), CHxE=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 CHxVAL 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 CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be assert, if CHxDEN =1.

So the process can be divided to 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.

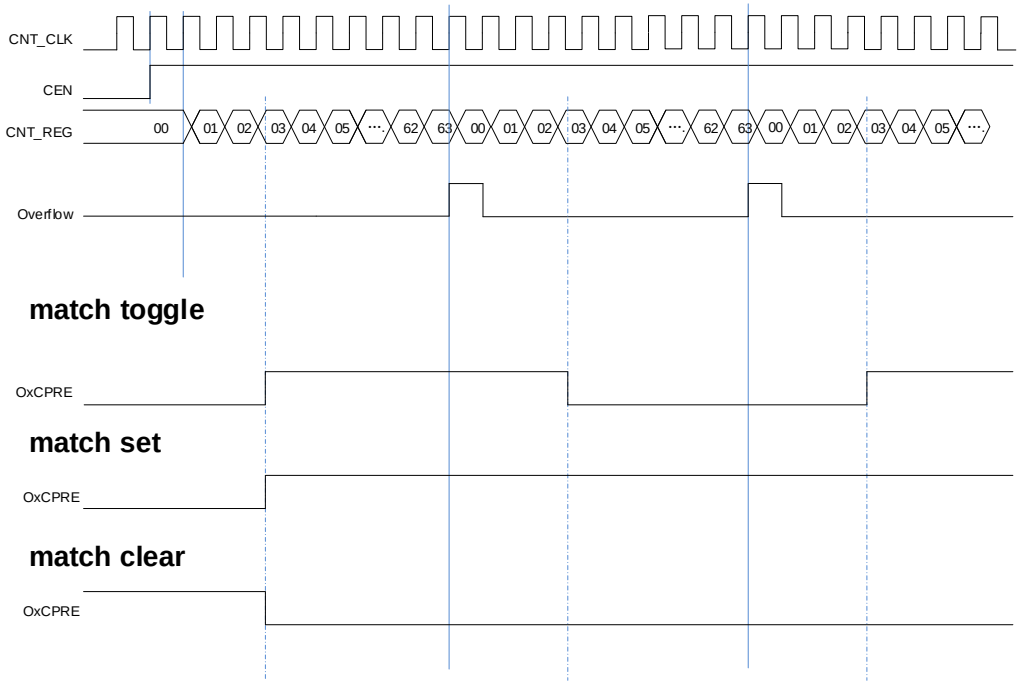
Step3: Interrupt/DMA-request enables configuration by CHxIE/CxCDE.

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 15-41. Output-compare under three modes show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 15-41. Output-compare under three modes



Output PWM function

In the PWM output function (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 15-42. Timing chart of EAPWM*** 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 15-43. Timing chart of CAPWM*** 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 inactive 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 active in PWM mode 1 (CHxCOMCTL=3'b111).

Figure 15-42. Timing chart of EAPWM

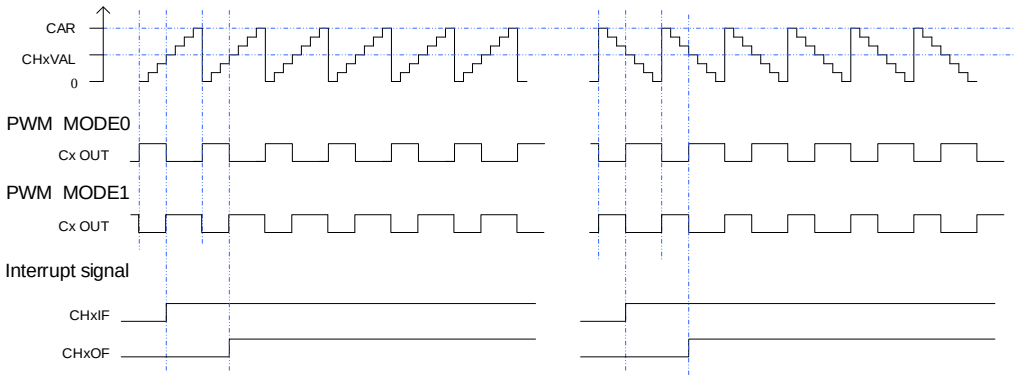
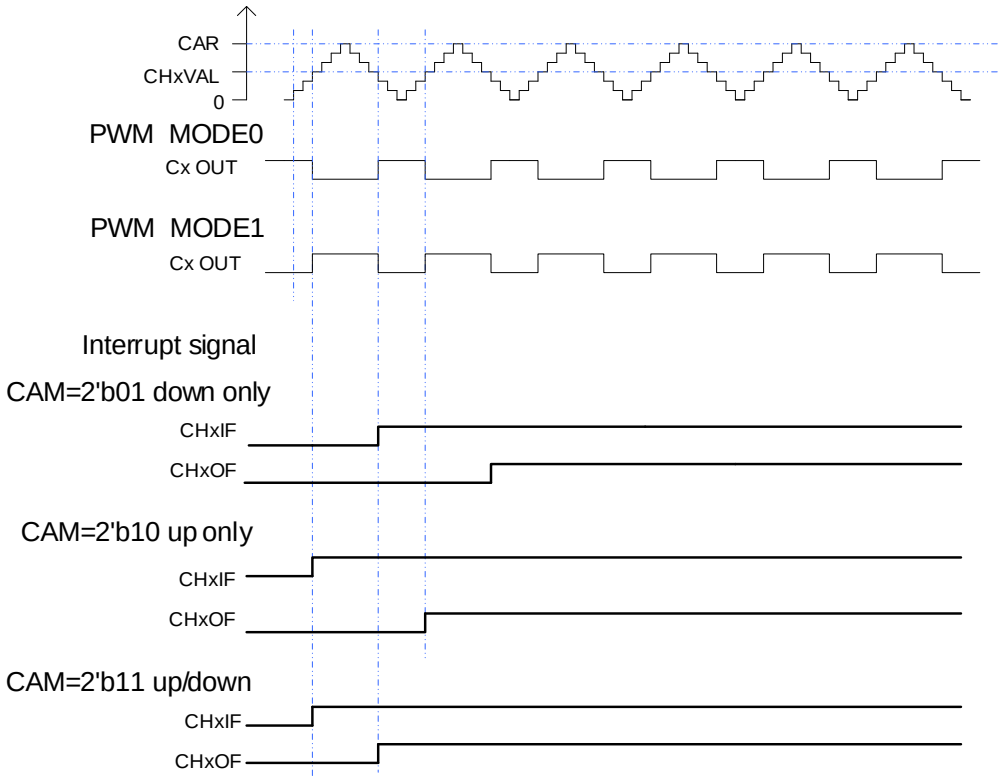


Figure 15-43. Timing chart of CAPWM



Channel output prepare signal

As is shown in ***Figure 15-40. 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

Refer to ***Quadrature decoder***.

Master-slave management

Refer to ***Master-slave management***.

Single pulse mode

Refer to ***Single pulse mode.***

Timers interconnection

Refer to ***Advanced timer (TIMERx, x=0)***.

Table 15-6. TIMERx(x=1,2) interconnection

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER1	TIMER0	TIMER14	TIMER2	Reserved
TIMER2	TIMER0	TIMER1	TIMER14	Reserved

Timer DMA mode

Timer's DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB; Of course, you have to enable a DMA

request which will be asserted by some internal interrupt event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB 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), then the timer's DMA request is finished. 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 DMASAR+0x4, DMASAR+0x8, DMASAR+0xc at the next 3 accesses to TIMERx_DMATB. In one word, one time DMA internal interrupt event assert, DMATC+1 times request will be send by TIMERx.

If one more time DMA request event coming, TIMERx will repeat the process as above.

Timer debug mode

When the Cortex®-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

15.2.5. TIMERx registers(x=1, 2)

TIMER1 base address: 0x4000 0000
TIMER2 base address: 0x4000 0400

Control register 0 (TIMERx_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						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 aligns 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 under center-aligned 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 under center-aligned 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 under center-aligned 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, 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 quadrature decoder mode, this bit is read only.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. 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 update interrupts or DMA requests: 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, pause mode and quadrature 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, CH1 and 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 resert source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a conter 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: DMA request of channel x is sent when channel x event occurs. 1: DMA request of channel x is sent when update event occurs.
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]			OCRC	SMC[2:0]		
rw	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 rising edge or high level . 1: ETI is active at falling edge or low level .
14	SMC1	Part of SMC for enable External clock mode1 In external clock mode 1, the counter is clocked by any active edge on the ETIF 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 clock source of the timer will be ETIFP if external clock mode 0 and external clock mode 1 are configured at the same time. **Note:** External clock mode 0 enable is in this register's SMC[2:0] bit-filed.
13:12	ETPSC[1:0]	The prescaler of external trigger The frequency of external trigger signal ETIFP must not be at higher than 1/4 of TIMER_CK frequency. When the external trigger signal is a fast clock, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disable. 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 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 disable

1: Master-slave mode enable

6:4 TRGS[2:0]

Trigger selection

This bit-field specifies which signal is selected as the trigger input, which is used to synchronize the counter.

000: ITI0

001: ITI1

010: ITI2

011: ITI3

100: CI0F_ED

101: CI0FE0

110: CI1FE1

111: ETIFP

These bits must not be changed when slave mode is enabled.

3 OCRC

OCPRE clear source selection

0: OCPRE_CLR_INT is connected to the OCPRE_CLR input

1: OCPRE_CLR_INT is connected to ETIF

2:0 SMC[2:0]

Slave mode control

000: Disable 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 edge, while the direction depends on each other.

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.

Because CI0F_ED outputs 1 pulse for each transition on CI0F, and the pause mode checks the level of the trigger signal, when CI0F_ED is selected as the trigger input, the pause mode must not be used.

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	31
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	CH3IF	CH1IF	CH0IF	UPIF
			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 ‘s capture/compare interrupt enable Refer to CH0IF description
3	CH2IF	Channel 2 ‘s capture/compare interrupt enable Refer to CH0IF description
2	CH1IF	Channel 1 ‘s capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 ‘s capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When 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 on an update event 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 is 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's capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2's capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1's capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high.

		0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) 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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	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
31:16	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	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
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 mode 01: Channel 1 is programmed as input mode, IS1 is connected to CI1FE1 10: Channel 1 is programmed as input mode, IS1 is connected to CI0FE1

		11: Channel 1 is programmed as input mode, IS1 is connected to ITS. **Note:** When CH1MS[1:0]=11, it is necessary to select an internal trigger input 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 disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. 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 mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(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 disable 1: Channel 0 output compare shadow enable 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.

2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
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 mode 01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0 10: Channel 0 is programmed as input mode, IS0 is connected to CI1FE0 11: Channel 0 is programmed as input mode, IS0 is connected to ITS **Note:** When CH0MS[1:0]=11, it is necessary to select an internal trigger input 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:

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 clear. 00: Prescaler disable, 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	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	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	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMFEN description
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 mode 01: Channel 3 is programmed as input mode, IS3 is connected to CI3FE3 10: Channel 3 is programmed as input mode, IS3 is connected to CI2FE3 11: Channel 3 is programmed as input mode, IS3 is connected to ITS. **Note:** When CH3MS[1:0]=11, it is necessary to ensure that 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 disable 1: Channel 2 output compare clear enable
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O2CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH2CV. 010: Clear the channel output. 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 high when the counter is smaller than TIMERx_CH2CV, and low otherwise. When counting down, O2CPRE is low when the counter is larger than TIMERx_CH2CV, and high otherwise. 111: PWM mode 1. When counting up, O2CPRE is low when the counter is smaller than TIMERx_CH2CV, and high otherwise. When counting down, O2CPRE is high when the counter is larger than TIMERx_CH2CV, and low otherwise.

		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-filed in TIMERx_CCHP register is 11 and CH2MS bit-filed is 00(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 disable 1: Channel 2 output compare shadow enable 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
2	CH2COMFEN	Channel 2 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM1 or PWM2 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the result of the comparison. 0: Channel 2 output quickly compare disable. 1: Channel 2 output quickly compare enable.
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 mode 01: Channel 2 is programmed as input mode, IS2 is connected to CI2FE2 10: Channel 2 is programmed as input mode, IS2 is connected to CI3FE2 11: Channel 2 is programmed as input mode, IS2 is connected to ITS. **Note:** When CH2MS[1:0]=11, it is necessary to select an internal trigger input 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 clear.

00: Prescaler disable, 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 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
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 output 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 output 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 output 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 is configured in output mode, this bit should be keep reset value. When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed 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 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed 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) (x=1)

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
CNT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CNT[31:0]	This bit-filed indicates the current counter value. Writing to this bit-filed can change the value of the counter.

Counter register (TIMERx_CNT) (x=2)

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-filed indicates the current counter value. Writing to this bit-filed 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-filed will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR) (x=1)

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
CARL[31:16]															

rw

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

rw

Bits	Fields	Descriptions
31:0	CARL[31:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter.

Counter auto reload register (TIMERx_CAR) (x=2)

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-filed specifies the auto reload value of the counter.

Channel 0 capture/compare value register (TIMERx_CH0CV) (x=1)

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
CH0VAL[31:16]															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:0	CH0VAL[31:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value

corresponding to the last capture event. And this bit-filed is read-only.
When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 0 capture/compare value register (TIMERx_CH0CV) (x=2)

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 or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV) (x=1)

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
CH1VAL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

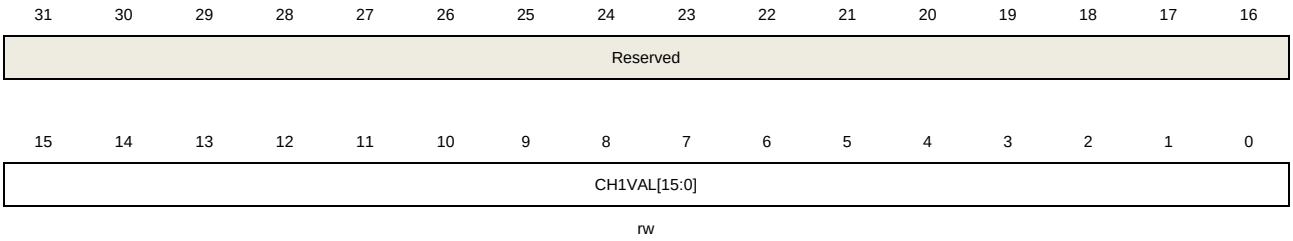
Bits	Fields	Descriptions
31:0	CH1VAL[31:0]	Capture or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value

corresponding to the last capture event. And this bit-filed is read-only.
When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV) (x=2)

Address offset: 0x38
Reset value: 0x0000 0000

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

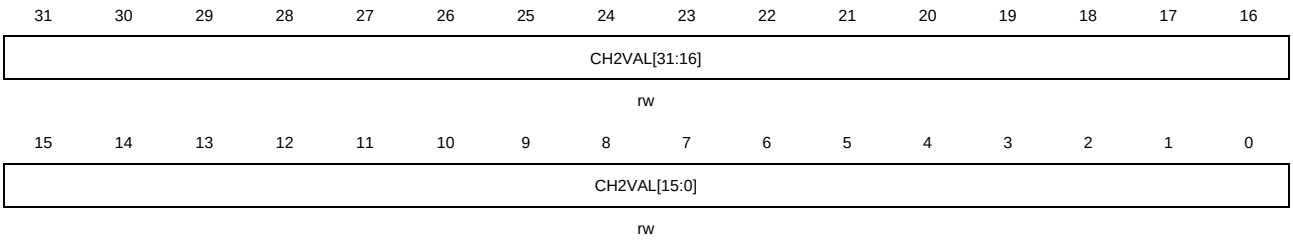


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH1VAL[15:0]	Capture or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV) (x=1)

Address offset: 0x3C
Reset value: 0x0000 0000

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



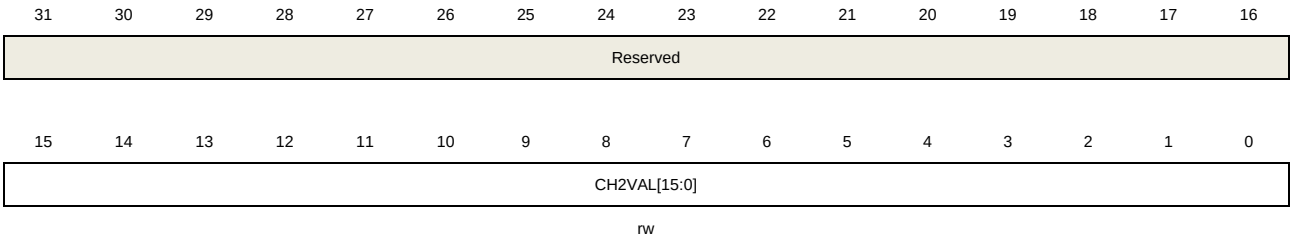
Bits	Fields	Descriptions
31:0	CH2VAL[31:0]	Capture or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value

corresponding to the last capture event. And this bit-filed is read-only.
When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV) (x=2)

Address offset: 0x3C
Reset value: 0x0000 0000

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

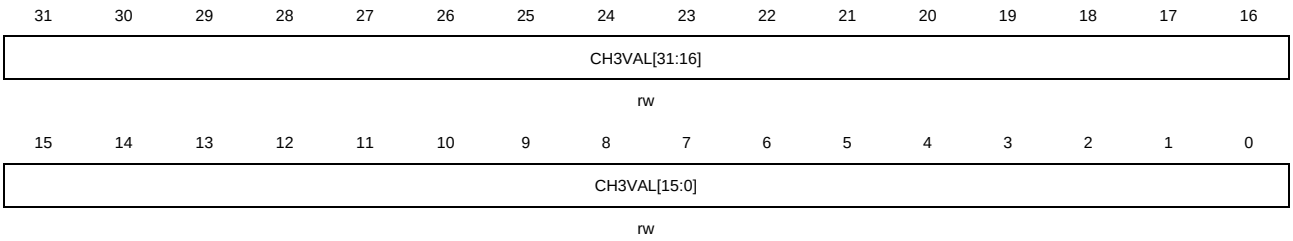


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH2VAL[15:0]	Capture or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV) (x=1)

Address offset: 0x40
Reset value: 0x0000 0000

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



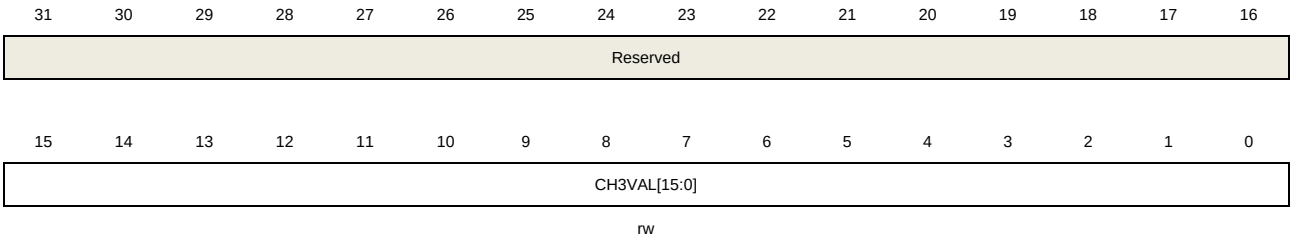
Bits	Fields	Descriptions
31:0	CH3VAL[31:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value

corresponding to the last capture event. And this bit-filed is read-only.
When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV) (x=2)

Address offset: 0x40
Reset value: 0x0000 0000

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

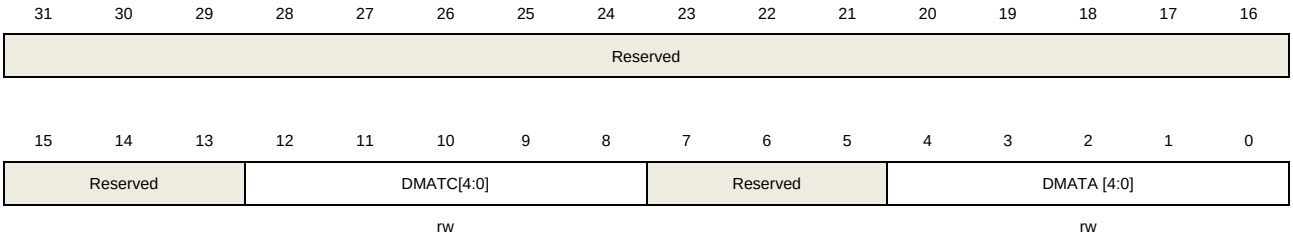


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH3VAL[15:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0x48
Reset value: 0x0000 0000

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



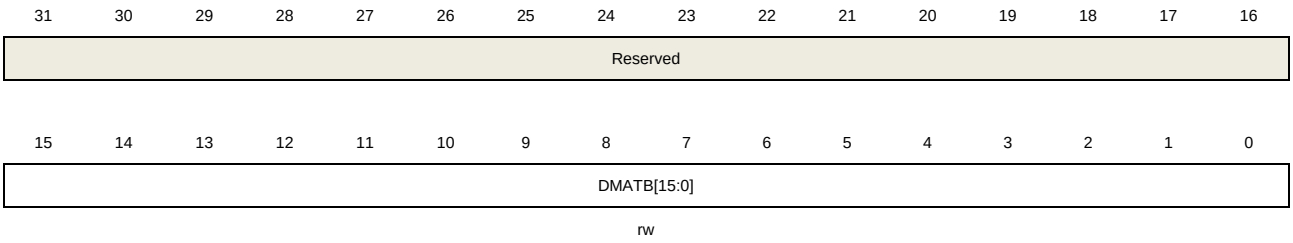
Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count

		This filed is defined the number of DMA will access(R/W) the register of TIMERx_DMATB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed 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.

DMA transfer buffer register (TIMERx_DMATB)

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
15:0	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.

15.3. General level2 timer (TIMERx, x=13)

15.3.1. Overview

The general level2 timer module (TIMER 13) is a one-channel timer that supports input capture, output compare. They can generate PWM signals to control motor or be used for power management applications. The general level2 time reference is a 16-bit counter that can be used as an unsigned counter.

In addition, the general level2 timers can be programmed and be used to count or time external events that drive other timers.

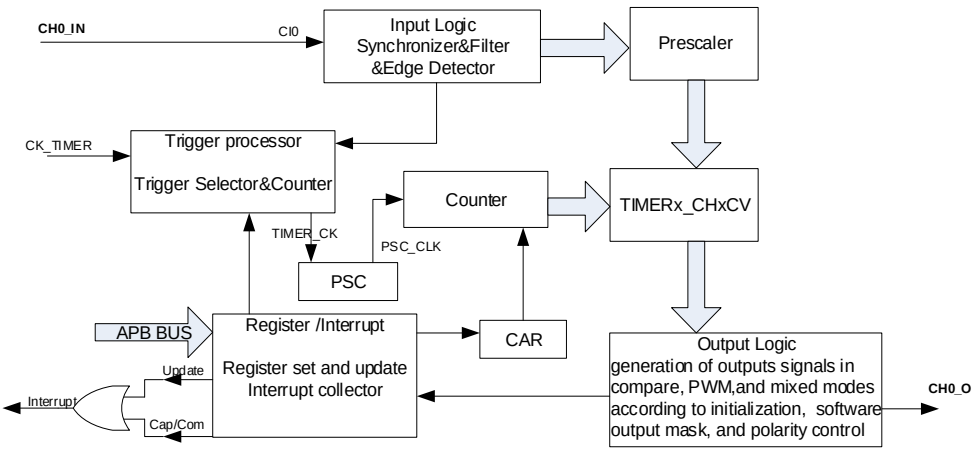
15.3.2. Characteristics

- Total channel num: 1.
- Counter width: 16bit.
- Source of count clock: internal clock.
- Counter mode: count up only.
- Programmable prescaler: 16 bit. Factor can be changed on the go.
- Each channel is user-configurable:
  Input capture mode, output compare mode, programmable and PWM mode.
- Auto-reload function.
- Interrupt output on: update, compare/capture event.

15.3.3. Block diagram

Figure 15-44. General level2 timer block diagram provides details on the internal configuration of the general level2 timer.

Figure 15-44. General level2 timer block diagram



15.3.4. Function overview

Clock source configuration

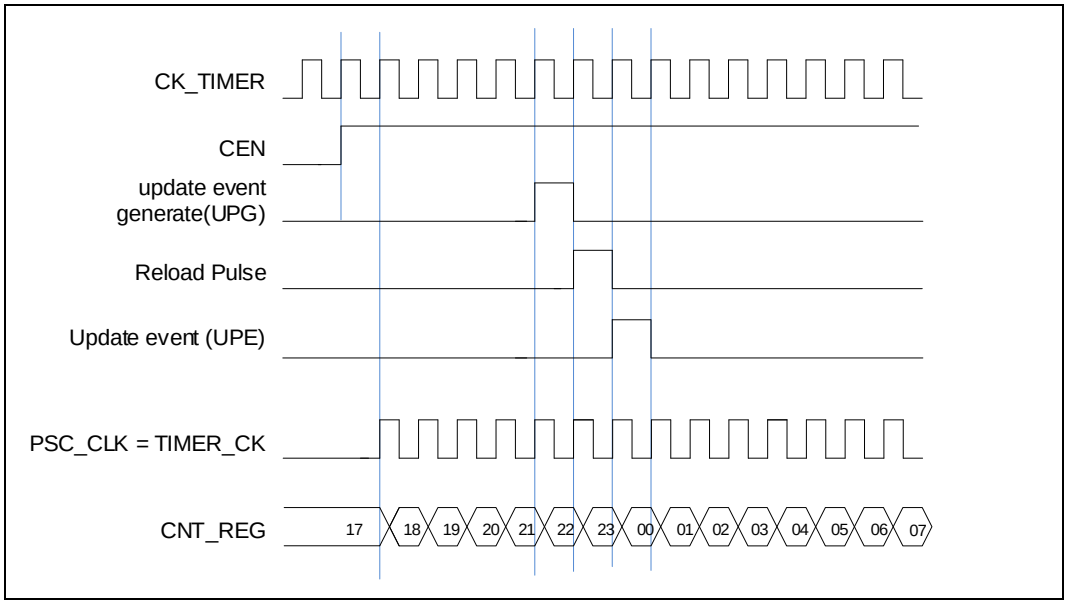
The general level2 TIMER can only being clocked by the CK_TIMER.

- Internal timer clock CK_TIMER which is from module RCU

The general level2 TIMER has only one clock source which is the internal 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.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

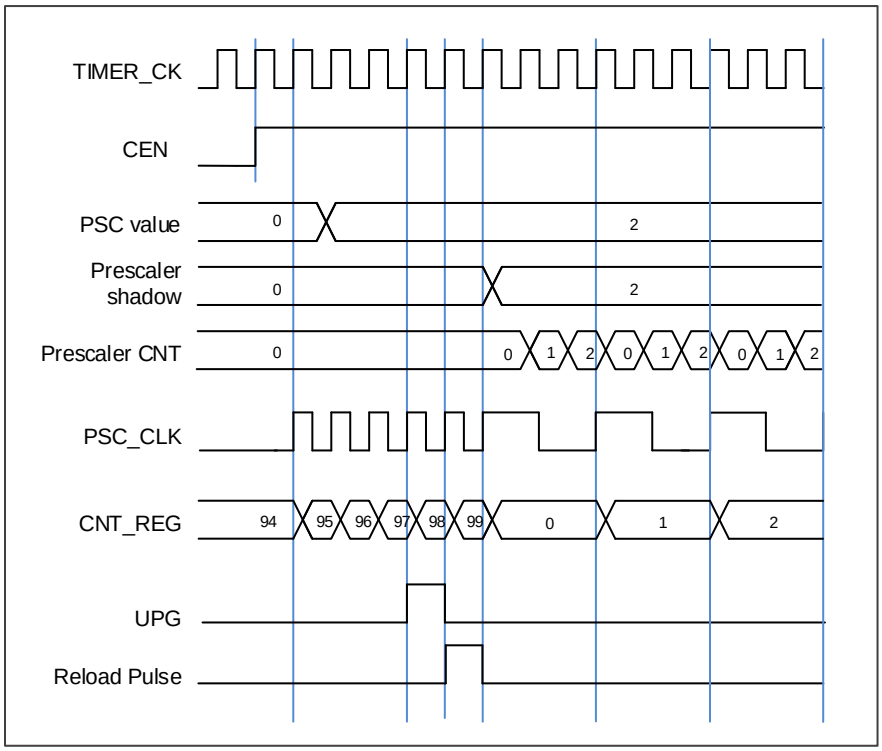
Figure 15-45. Timing chart of internal clock divided by 1



Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale 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 15-46. 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 (counter 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 15-47. Timing chart of up counting mode, PSC=0/2

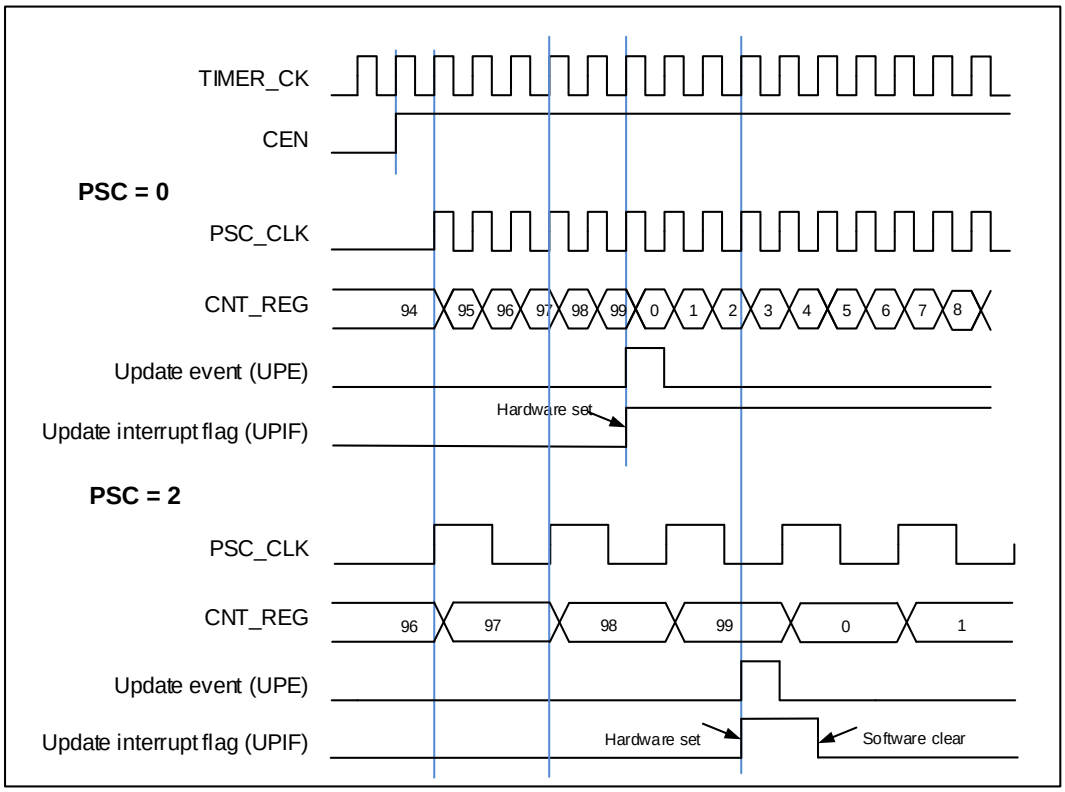
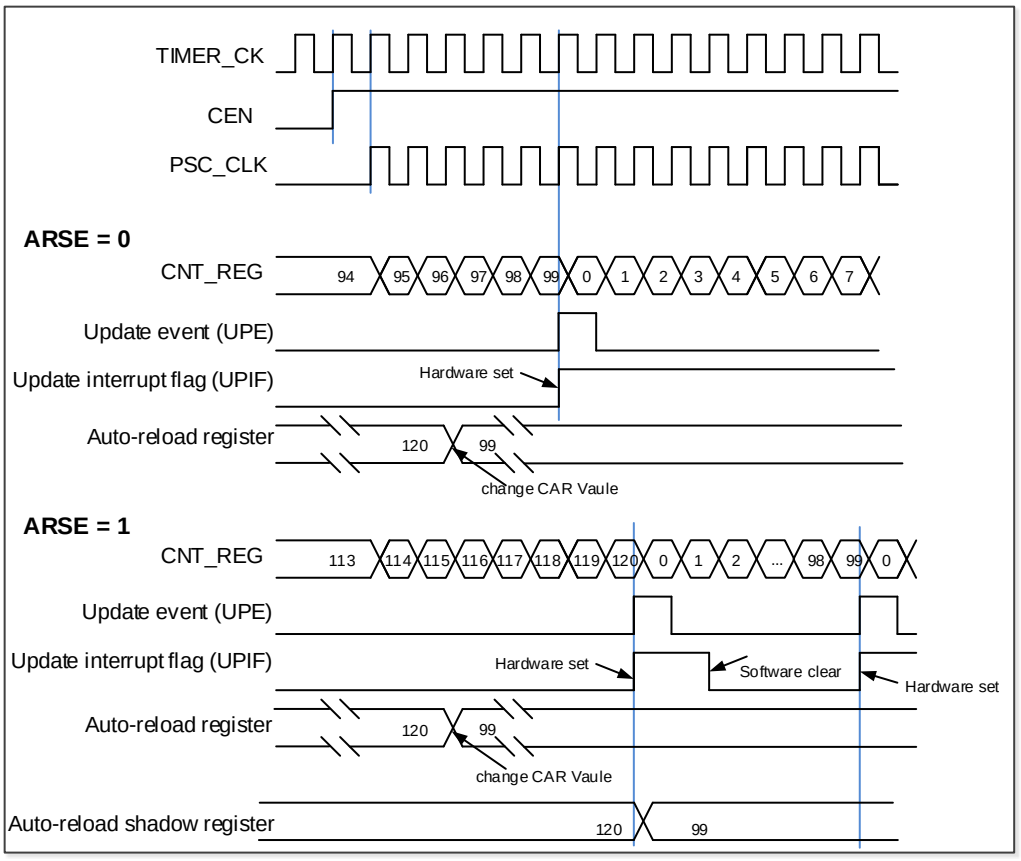


Figure 15-48. Timing chart of up counting mode, change TIMERx_CAR on the go



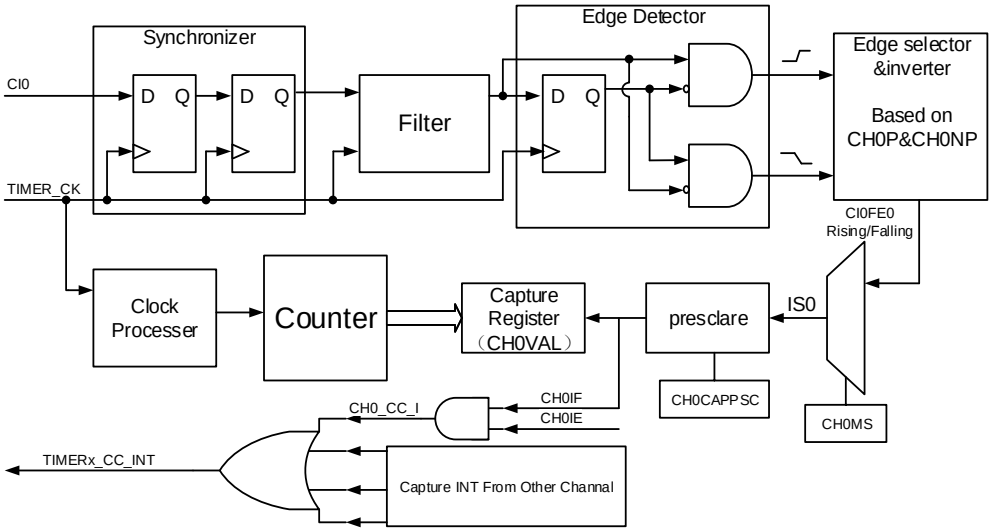
Input capture and output compare channels

The general level2 timer has one independent channel 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 enabled by CHxIE = 1.

Figure 15-49. Channel input capture principle



First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and fall edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So the process can be divided to 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, falling edge or both edges (rising and 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 in TIMERx_DMAINTEN.

Direct generation: An interrupt is generated by setting CHxG directly.

The channel input capture function can be also used for pulse period 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. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period.

- **Channel output compare function**

Figure 15-50. Channel output compare principle

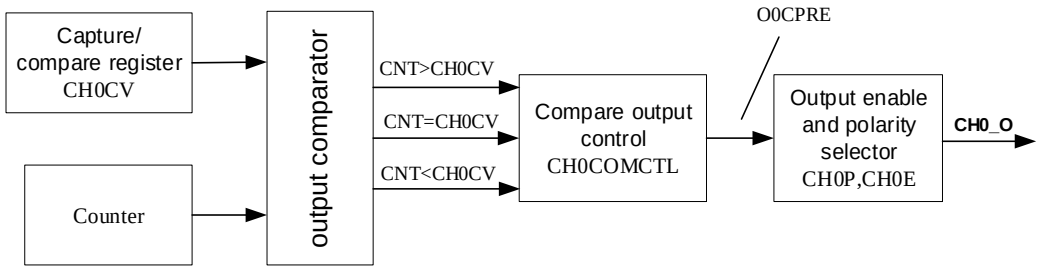


Figure 15-50. Channel output compare principle shows the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O and the OxCPRE signal is described as blew: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE signal, 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), CHxE=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 CHxVAL 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 CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 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.

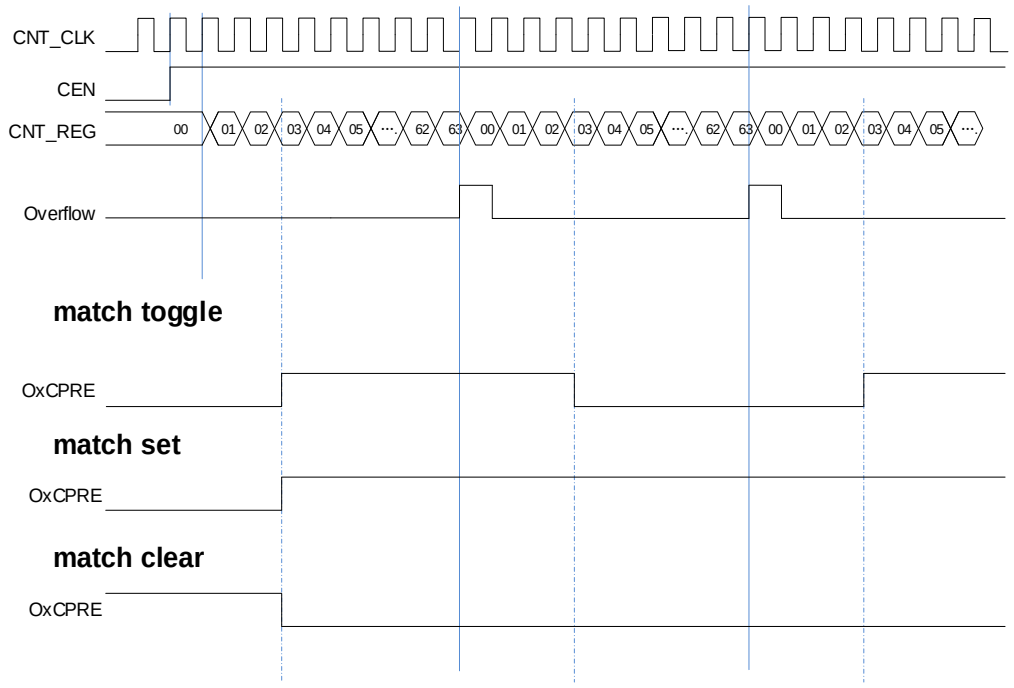
Step3: Interrupt/DMA-request enables configuration by CHxIE.

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 15-51. Output-compare under three modes show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 15-51. Output-compare under three modes



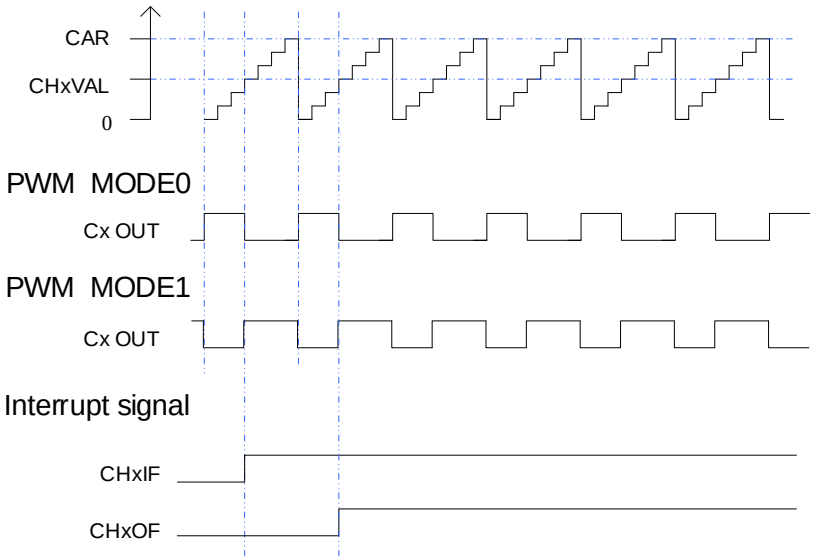
Output PWM function

In the output PWM function (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

The period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. ***Figure 15-52. PWM mode timechart*** shows the PWM output mode and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).
And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3'b110).

Figure 15-52. PWM mode timechart



Channel output prepare signal

As is shown in ***Figure 15-50. Channel output compare principle*** 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.

Timer debug mode

When the Cortex®-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

15.3.5. TIMERx registers(x=13)

TIMER13 base address: 0x4000 2000

Control register 0 (TIMERx_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						CKDIV[1:0]		ARSE	Reserved				UPS	UPDIS	CEN
						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
6:3	Reserved	Must be kept at reset value.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate update interrupts or DMA requests: 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, pause mode and quadrature decoder mode.

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	31
Reserved														CH0IE	UPIE
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
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						CH0OF	Reserved.							CH0IF	UPIF
						rc_w0								rc_w0	rc_w0

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
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
8:2	Reserved	Must be kept at reset value.
1	CH0IF	Channel 0 ‘s capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When 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 on an update event 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														CH0G	UPG
														w	w

Bits	Fields	Descriptions

31:2	Reserved	Must be kept at reset value.
1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	This bit can be set by software, and cleared by hardware automatically. 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

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved.								Reserved	CH0COMCTL[2:0]			CH0COMSEN	Reserved	CH0MS[1:0]	
								CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
									rw			rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. 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 mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(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 disable 1: Channel 0 output compare shadow enable 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
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 mode 01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0 10: Reserved 11: Reserved

Input capture mode:

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
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 clear.

00: Prescaler disable, 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 2 (TIMERx_CHCTL2)

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..												CH0NP	Reserved	CH0P	CH0EN
												rw		rw	rw

Bits	Fields	Descriptions

31:4	Reserved	Must be kept at reset value
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	Reserved	Must be kept at reset value
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 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in input mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in output 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-filed indicates the current counter value. Writing to this bit-filed 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-filed 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															

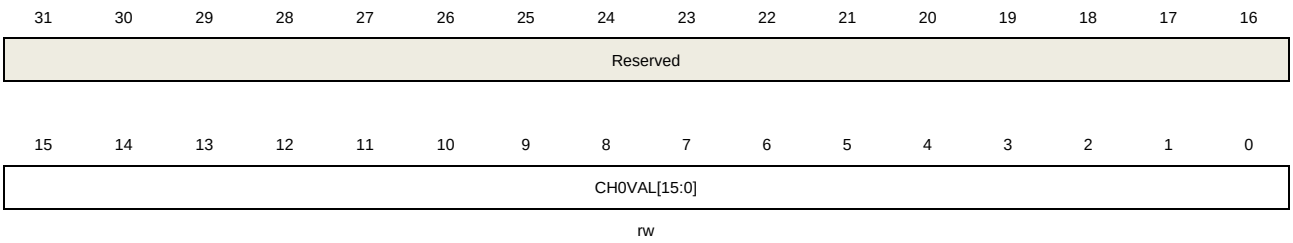
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter.

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).

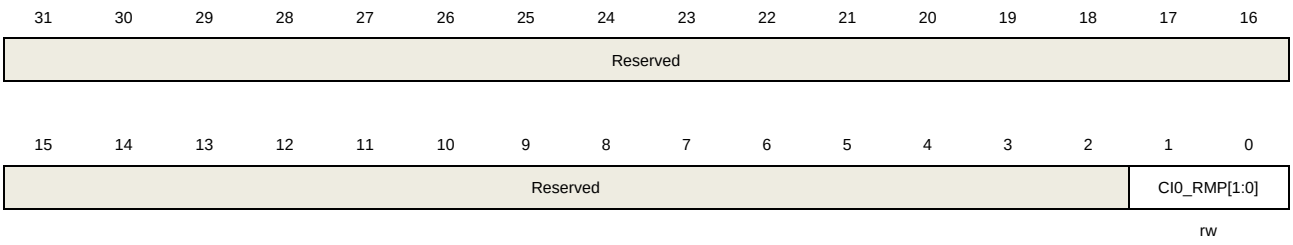


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel input remap register(TIMERx_IRMP)

Address offset: 0x50
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
1:0	CI0_RMP[1:0]	Channel 0 input remap 00: Channel 0 input is connected to GPIO(TIMER13_CH0) 01: Channel 0 input is connected to the RTCCLK 10: Channel 0 input is connected to HXTAL/32 clock 11: Channel 0 input is connected to CKOUTSEL, which is controlled by RCU_CFG0.

15.4. General level3 timer (TIMERx, x=14)

15.4.1. Overview

The general level3 timer module (TIMER14) is a two-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 general level3 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level3timers 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 issuitable for motor control applications.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counters incrementing in unison.

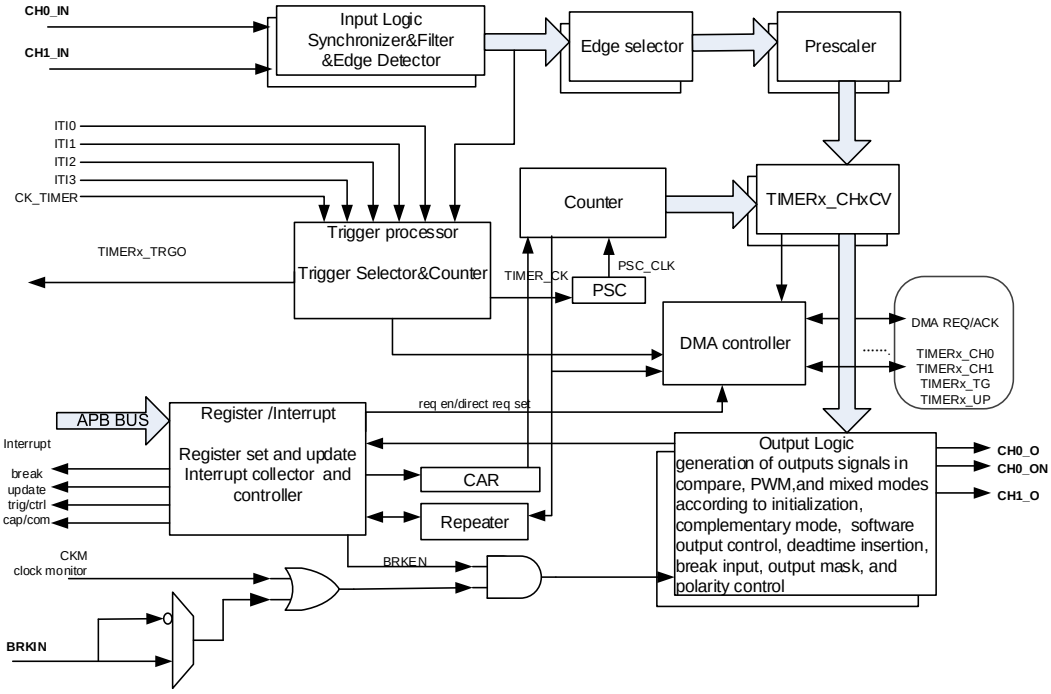
15.4.2. Characteristics

- Total channel num: 2.
- Counter width: 16 bit.
- Source of counter clock is selectable: internal clock, internal trigger, external input.
- Counter modes: count up only.
- Programmable prescaler: 16 bit.The factor can be changed on the go.
- Each channel is user-configurable:
  input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request on: update, trigger event, compare/capture event, and break input.
- Daisy chaining of timer modules allows a single timer to initiate multiple timers.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer master-slave management.

15.4.3. Block diagram

Figure 15-53. General level3 timer block diagram provides details of the internal configuration of the general level3 timer.

Figure 15-53. General level3 timer block diagram



15.4.4. Function overview

Clock source configuration

The general level3 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]).

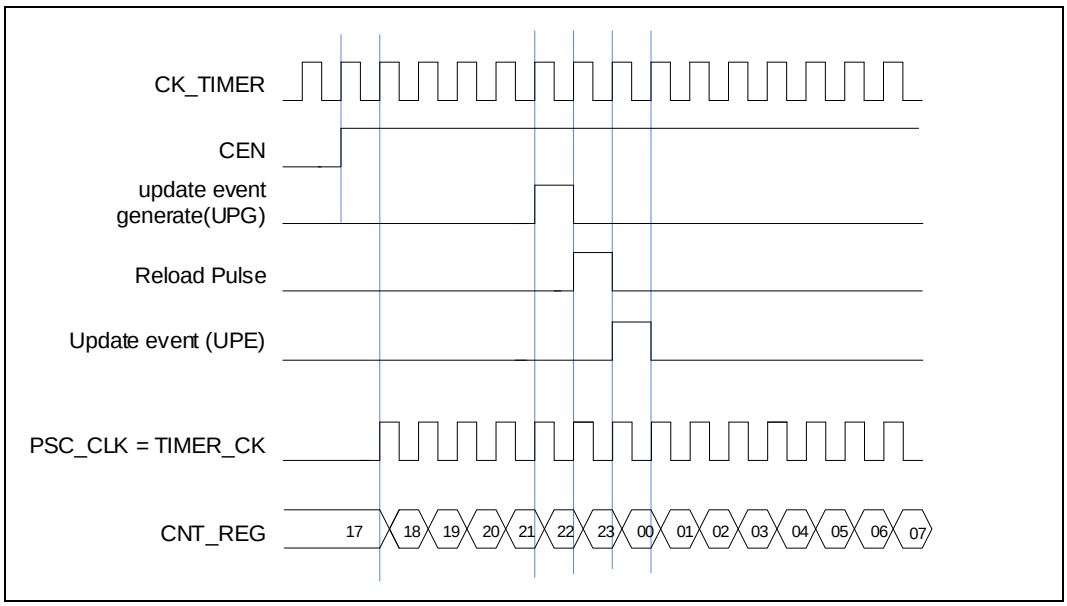
- SMC [2:0] == 3'b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

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.

If the SMC [2:0] in the TIMERx_SMCFG register are setting to an available value 0x7, the prescaler is clocked by other clock sources selected by the TRGS [2:0] in the TIMERx_SMCFG register, details as follows. 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 15-54. 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

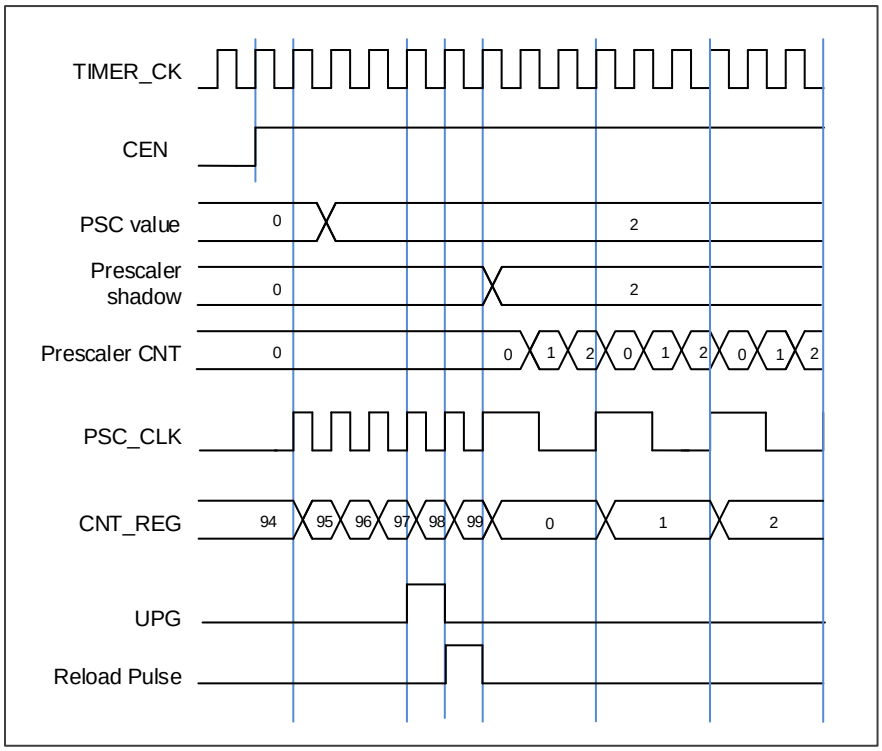
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting SMC [2:0] to 0x7 and the TRGS [2:0] to 0x4, 0x5 or 0x6.

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.

Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale 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 15-55. 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 generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter, counter 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 15-56. Timing chart of up counting mode, PSC=0/2

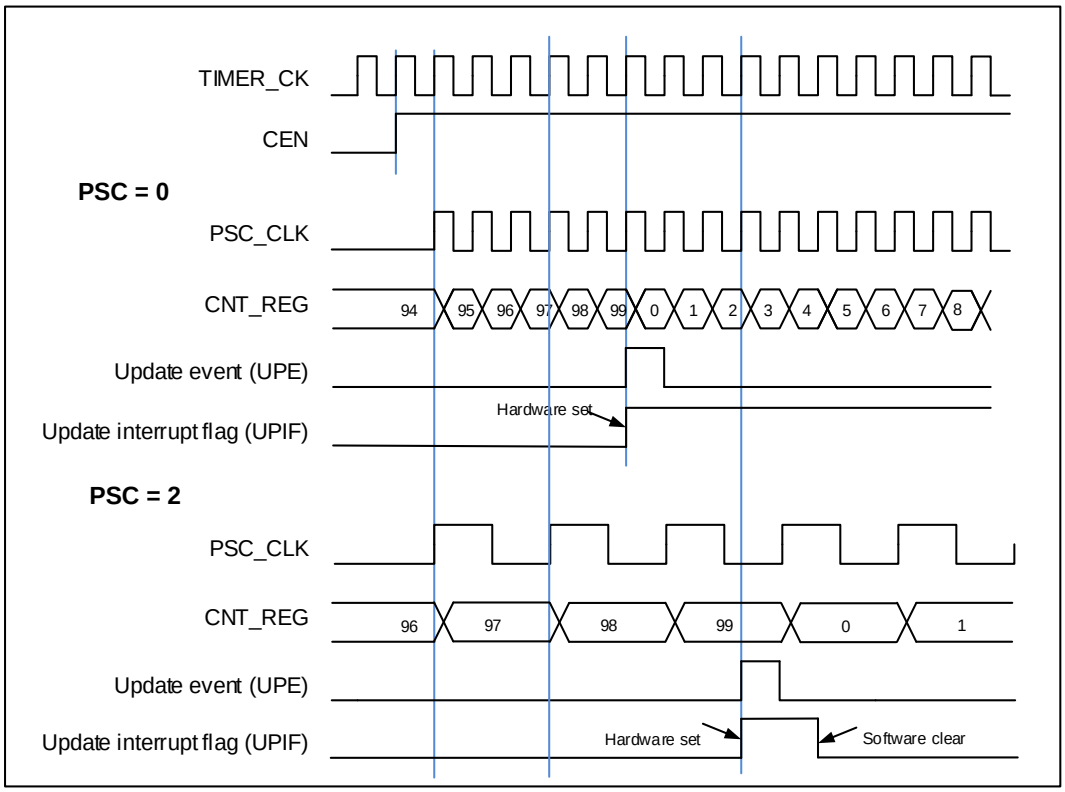
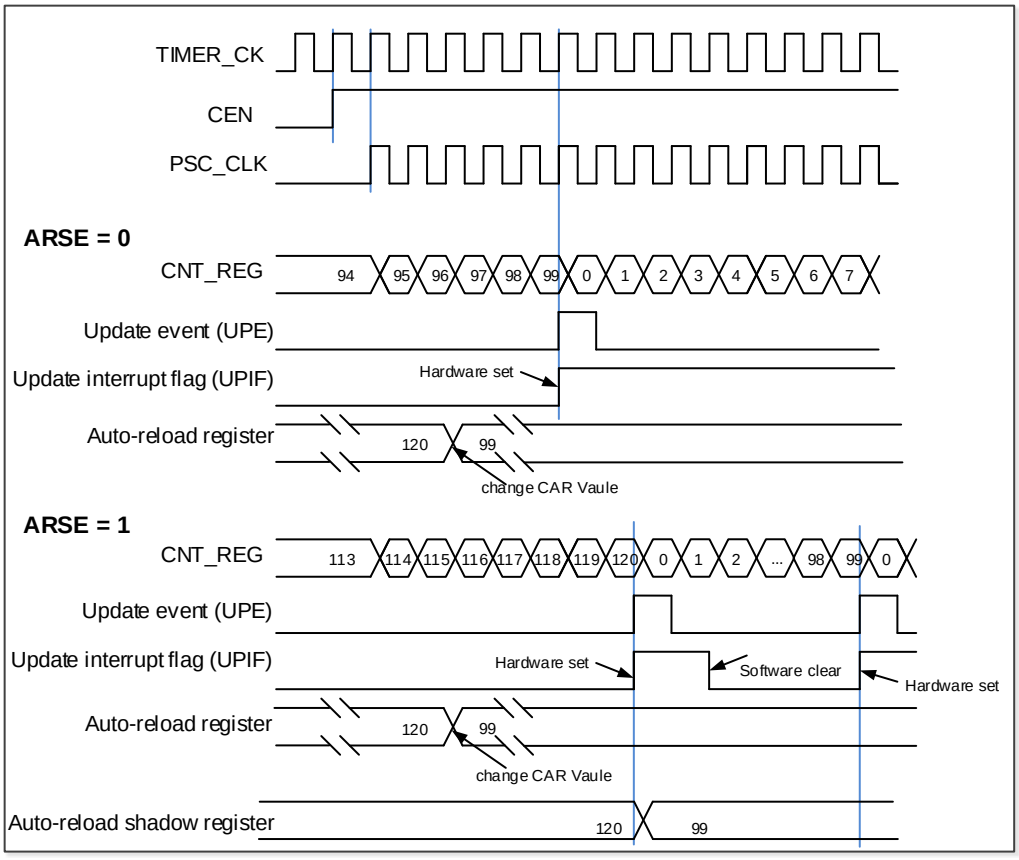


Figure 15-57. Timing chart of up counting mode, change TIMERx_CAR on the go

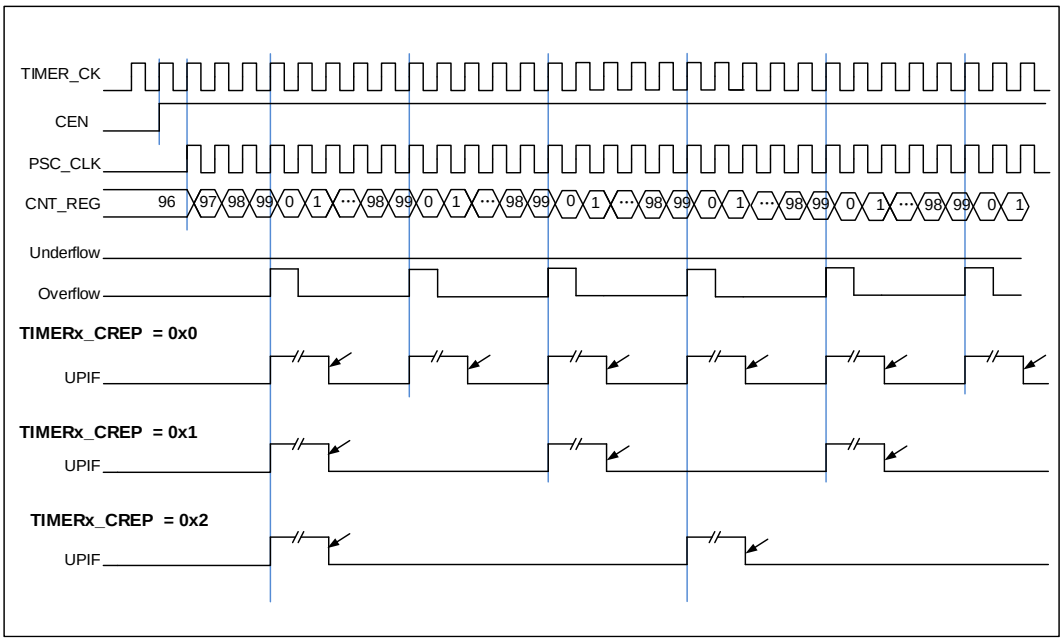


Update event (from overflow/underflow) rate configuration

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 in up-counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generator an update event.

Figure 15-58. Repetition counter timing chart of up counting mode



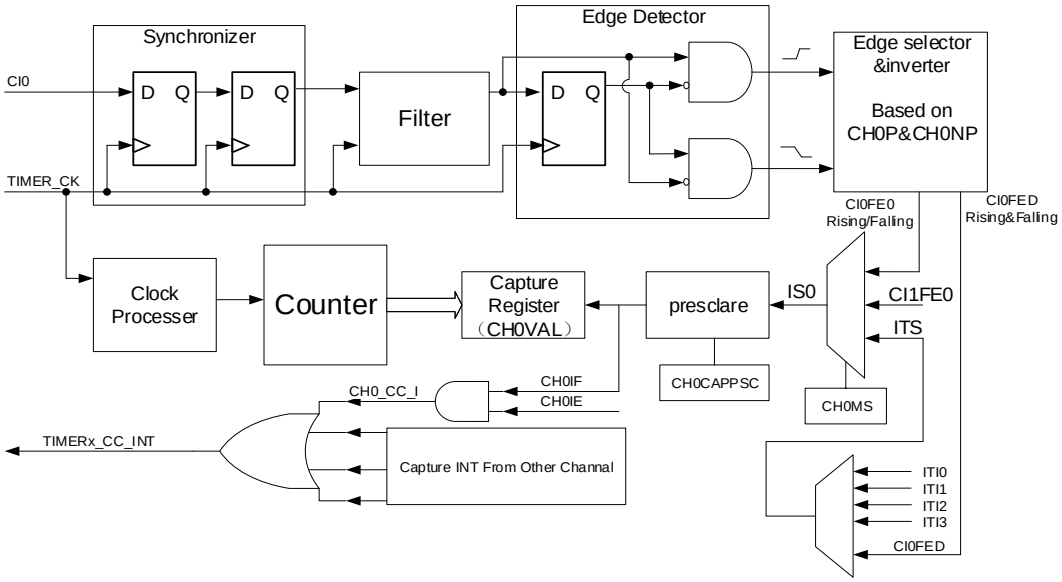
Input capture and output compare channels

The general level3 timer has two 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 enabled by CHxIE = 1.

Figure 15-59. Channel input capture principle



Channels' input signals (CIx) is the TIMERx_CHx signal. First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and falling edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So the process can be divided to several steps as below:

Step1: Filter configuration. (CHxCAPFLT in TIMERx_CHCTL0)

Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge selection. (CHxP/CHxNP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel 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 enable; you can got the interrupt and DMA request.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2)

Result: when you wanted input signal is got, TIMERx_CHxCV will be set by counter's value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt and DMA request will be asserted based on the configuration of CHxIE and CHxDEN in TIMERx_DMAINTEN

Direct generation: if you want to generate a DMA request or Interrupt, you can set CHxG by software 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 connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

- **Channel output compare function**

Figure 15-60. Channel output compare principle (with complementary output, x=0)

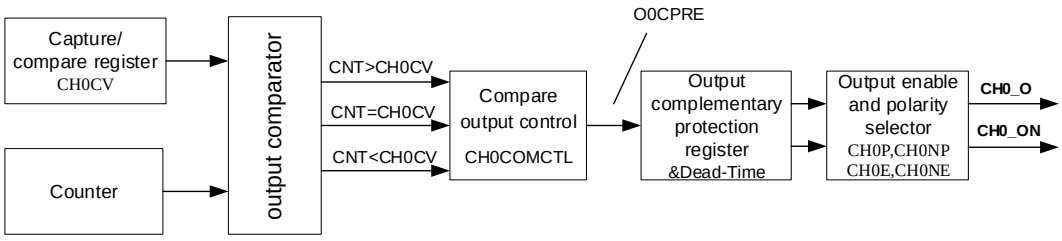


Figure 15-61. Channel output compare principle (CH1_O)

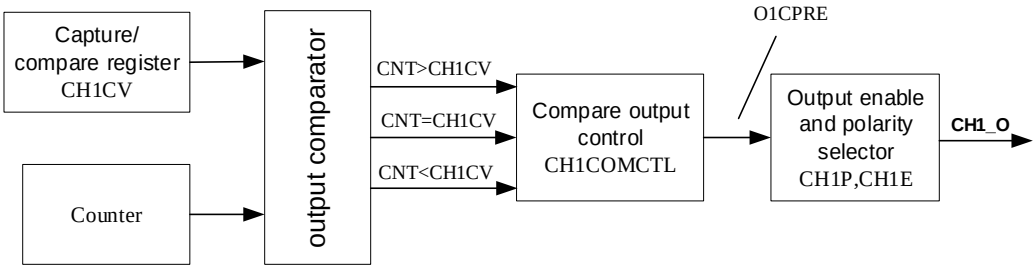


Figure 15-60. Channel output compare principle (with complementary output, x=0) and *Figure 15-61. Channel output compare principle (CH1_O)* show the logic circuit of output compare mode. The relationship between the channel output signal CHx_O/CHx_ON and the OxCPRE signal is described as blew: The active level of O0CPRE is high, the output level of CH0_O/CH0_ON depends on OxCPRE signal, CHxP/CHxNP bit and CH0E/CH0NE bit (please refer to the TIMERx_CHCTL2 register for more details). For examples, configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxE=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.

Configure CHxNP=0 (the active level of CHx_ON is low, contrary to OxCPRE), CHxNE=1 (the

output of CHx_ON is enabled):

If the output of OxCPRE is active(high) level, the output of CHx_O is active(low) level;

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(high) level.

When CH0_O and CH0_ON are output at the same time, the specific outputs of CH0_O and CH0_ON are related to the relevant bits (ROS, IOS, POE and DTCFG bits) in the TIMERx_CCHP register.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the CHxVAL 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 CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be assert, if CHxDEN =1.

So the process can be divided to 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 high polarity by CHxP/CHxNP
* Enable the output by CHxEN

Step3: Interrupt/DMA-request enables configuration by CHxIE/ CHxDEN

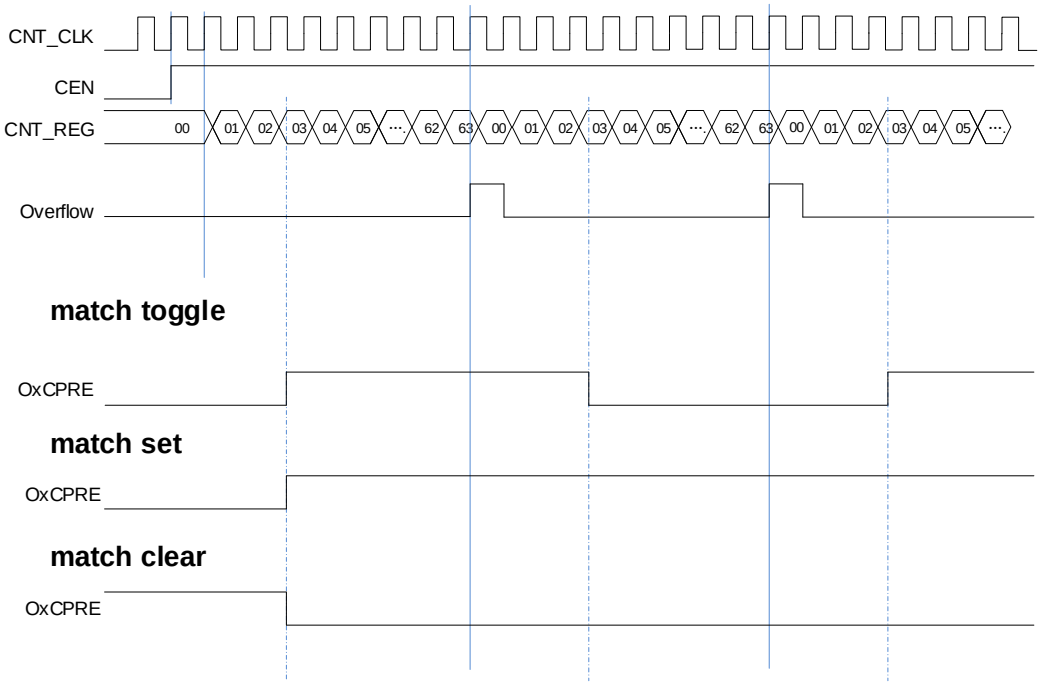
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV

About the CHxVAL; you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

Figure 15-62. Output-compare under three modes show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 15-62. Output-compare under three modes



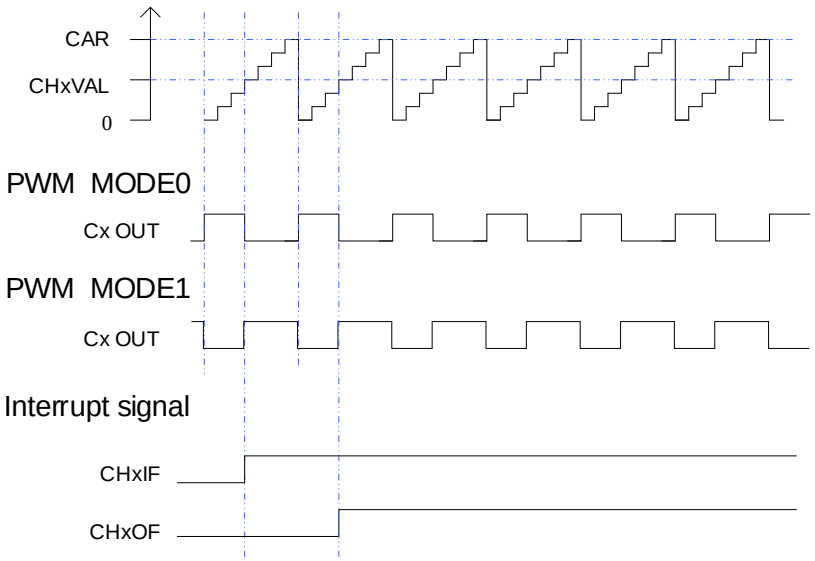
Output PWM function

In the output PWM function (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

The period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. ***Figure 15-63. PWM mode timechart*** shows the PWM output mode and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).
And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3'b110).

Figure 15-63. PWM mode timechart



Channel output prepare signal

When the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL filed. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting 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. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

Outputs complementary

Function of complementary is for a pair of CHx_O and CHx_ON. Those two output signals cannot be active at the same time. The TIMERx has 2 channels, but only the first channel have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register and the POEN, ROS, IOS, ISOx and ISOxN bits in the TIMERx_CCHP and TIMERx_CTL1 registers. The outputs polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 15-7. Complementary outputs controlled by parameters

<table>
<tr><th colspan="5">Complementary Parameters</th><th colspan="2">Output Status</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON output disable(1).</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">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)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">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.</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="6">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O output disable.</td><td>CHx_ON =OxCPRE⊕(4)CHxNP
CHx_ON output enable.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON = LOW
CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON =(!OxCPRE)(5)⊕CHxNP.
CHx_ON output enable.</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O output “off-state”.</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O output “off-state”</td><td>CHx_ON =OxCPRE⊕CHxNP
CHx_ON output enable</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON =(!OxCPRE)⊕CHxNP
CHx_ON output enable.</td></tr>
</table>

Note:

(2) 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.

Dead time insertion

The dead time insertion is enabled when both CHxEN and CHxNEN are 1'b1, and set POEN is also necessary. The field named DTCFG defines the dead time delay that can be used for channel 0. The detail about the delay time, refer to the register TIMERx_CCHP.

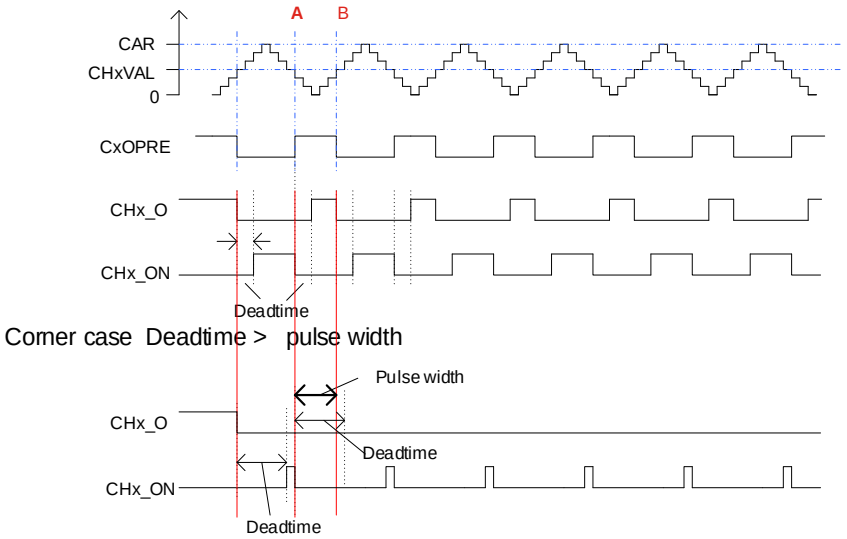
The dead time delay insertion ensures that no two complementary signals drive the active state at the same time.

When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the ***Figure 15-64. Complementary output with dead-time insertion.*** CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at once. Similarly, At point B when counter match (counter = CHxVAL) occurs again, OxCPRE is cleared, CHx_O signal will be cleared at once, while CHx_ON signal remains at the low value 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 CHx_O duty cycle, then the CHx_O signal is always the inactive value. (as show in the***Figure 15-64. Complementary output with dead-time insertion.)***
- The dead time delay is greater than or equal to the CHx_ON duty cycle, then the CHx_ON signal is always the inactive value.

Figure 15-64. Complementary output with dead-time insertion.



Break function

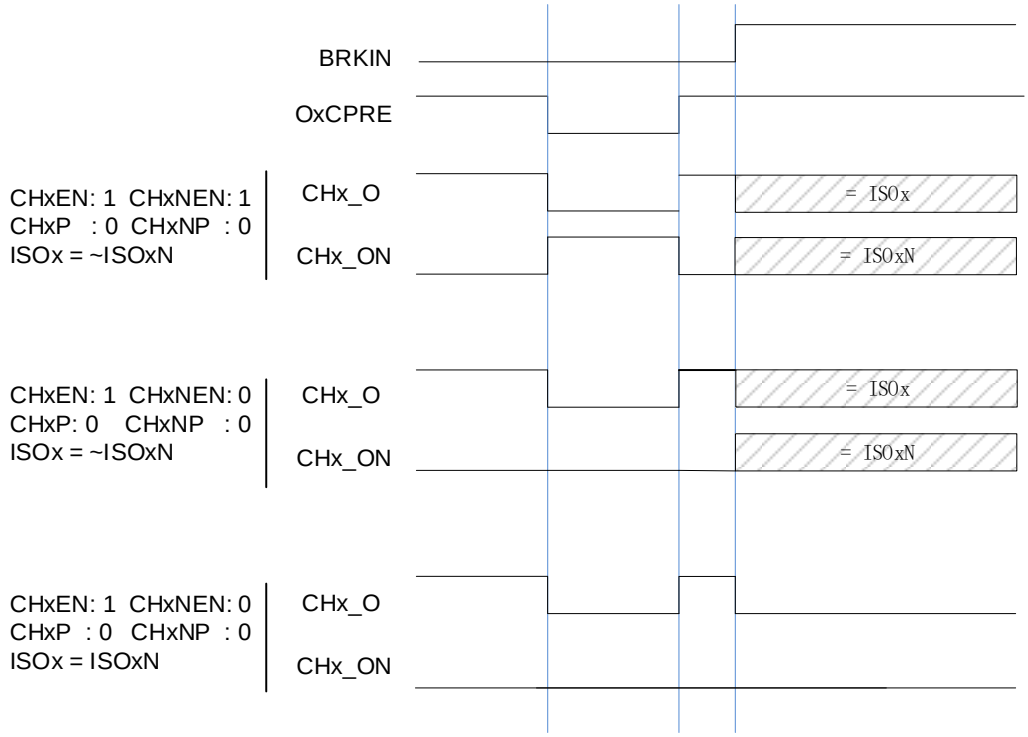
In this function, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot be set both to active level when break occurs. The break sources are input break pin and HXTAL stuck event by Clock Monitor (CKM) in RCU. The break function enabled by

setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is setting by the BRKP bit in TIMERx_CCHP.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order 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 is set. If BRKIE is 1, an interrupt generated.

Figure 15-65. Output behavior in response to a break(The break high active)

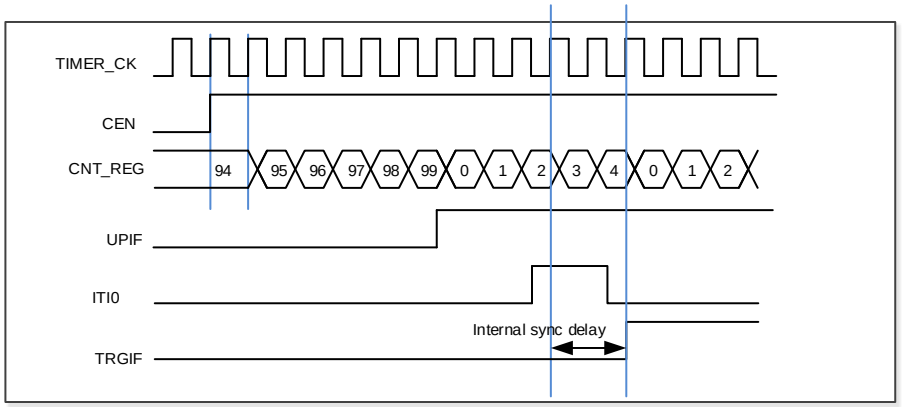
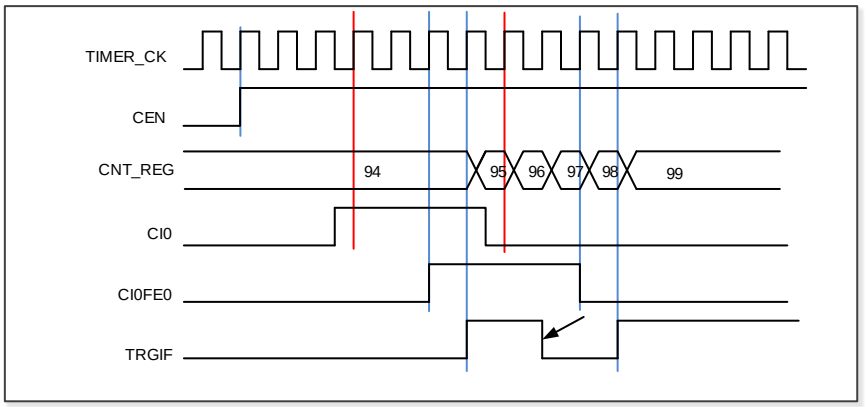


Master-slave management

The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode which is selected by the SMC[2:0] in the TIMERx_SMCFG register. The trigger input of these modes can be selected by the TRGS[2:0] in the TIMERx_SMCFG register.

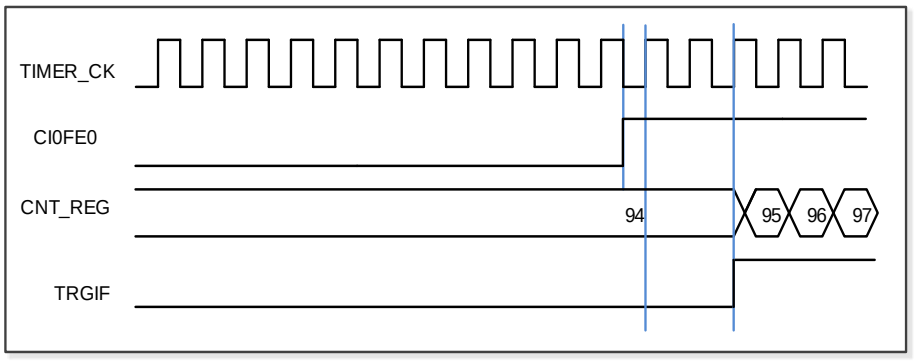
Table 15-8. Slave mode example table

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	SMC[2:0] 3'b100 (restart mode)	TRGS[2:0] 000: ITI0 001: ITI1	If you choose the CI0FE0 or CI1FE1, configure the CHxP and CHxNP for the	For the ITIx no filter and prescaler can be used. For the CIx, configure Filter

<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>3'b101 (pause mode)
3'b110 (event mode)</td><td>010: ITI2
011: ITI3
100: CI0F_ED
101: CI0FE0
110: CI1FE1
111: Reserved</td><td>polarity selection and inversion.</td><td>by CHxCAPFLT, no prescaler can be used.</td></tr>
<tr><td rowspan="2">Exam1</td><td>Restart mode
The counter can be clear and restart when a rising trigger input.</td><td>TRGS[2:0]=3’b000
ITI0 is the selection.</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 15-66. Restart mode

</td></tr>
<tr><td rowspan="2">Exam2</td><td>Pause mode
The counter can be paused when the trigger input is low.</td><td>TRGS[2:0]=3’b101
CI0FE0 is the selection.</td><td>TI0S=0.(Non-xor) [CH0NP==0, CH0P==0] no inverted. Capture will be sensitive to the rising edge only.</td><td>Filter is bypass in this example.</td></tr>
<tr><td colspan="4">Figure 15-67. Pause mode

</td></tr>
<tr><td>Exam3</td><td>Event mode
The counter will start to count</td><td>TRGS[2:0]=3’b101
CI0FE0 is the</td><td>TI0S=0(Non-xor) [CH0NP==0, CH0P==0] no inverted.</td><td>Filter is bypass in this example.</td></tr>
</table>

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	when a rising trigger input.	selection.		

Figure 15-68. Event mode



Single pulse mode

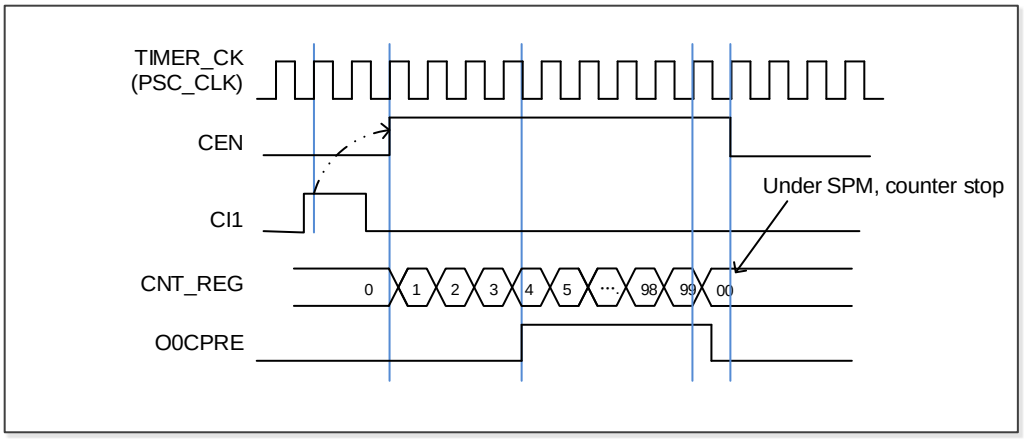
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the trigger signals edge or by setting the CEN bit to 1 using software. Setting the CEN bit to 1 or a trigger from the trigger signals 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 using software, the counter will be stopped and its value held.

In the single pulse mode, the trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0 register. 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. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 15-69. Single pulse mode TIMERx CHxCV = 4 TIMERx CAR=99 shows an example.

Figure 15-69. Single pulse mode TIMERx_CHxCV = 4 TIMERx_CAR=99



Timers interconnection

Refer to ***Advanced timer (TIMERx, x=0)***.

Table 15-9. TIMERx(x=14) interconnection

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER14	TIMER1	TIMER2	Reserved	Reserved

Timer DMA mode

Timer's DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Of course, you have to enable a DMA request which will be asserted by some internal event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB 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), then the timer's DMA request is finished. 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, DMATA+0xc at the next 3 accesses to TIMERx_DMATB. In one word, one time DMA internal interrupt event assert, DMATC+1 times request will be send by TIMERx.

If one more time DMA request event coming, TIMERx will repeat the process as above.

Timer debug mode

When the Cortex®-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

15.4.5. TIMERx registers(x=14)

TIMER14 base address: 0x4001 4000

Control register 0 (TIMERx_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						CKDIV[1:0]		ARSE	Reserved			SPM	UPS	UPDIS	CEN
						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
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. 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 update interrupts or DMA requests: 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, pause mode and quadrature 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					ISO1	ISO0N	ISO0	Reserved	MMC[2:0]			DMAS	CCUC	Reserved	CCSE
					rw	rw	rw		rw			rw	rw		rw

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of channel 0 complementary output 0: When POEN bit is reset, CH0_ON is set low. 1: When POEN bit is reset, CH0_ON 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 CH0_ON is implemented. This bit

		can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
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 in master mode to slave timers for synchronization function. 000: When a counter reset event occurs, a TGRO trigger signal is output. The counter resert source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a conter start event occurs, a TGRO 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 TGRO trigger signal is output. The update source depends on UPDIS bit and UPS bit. 011: When a capture or compare pulse event occurs in channal0, a TGRO trigger signal is output. 100: When a compare event occurs, a TGRO trigger signal is output. The compare source is from O0CPRE. 101: When a compare event occurs, a TGRO trigger signal is output. The compare source is from O1CPRE. 110: When a compare event occurs, a TGRO trigger signal is output. The compare source is from O2CPRE. 111: When a compare event occurs, a TGRO 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 enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below: 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by 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, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event

coming.

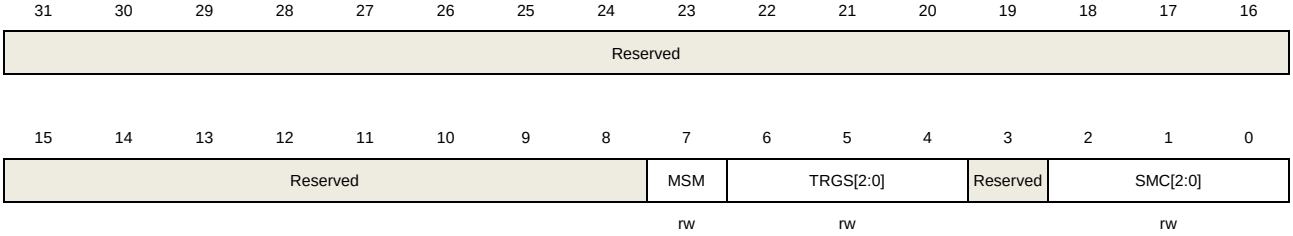
When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	MSM	Master-slave mode This bit can be used to synchronize 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 disable 1: Master-slave mode enable
6:4	TRGS[2:0]	Trigger selection This bit-field specifies which signal is selected as the trigger input, which is used to synchronize the counter. 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0 edge flag (CI0F_ED) 101: Channel 0 input filtered output (CI0FE0) 110: Channel 1 input filtered output (CI1FE1) 111: 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 mode. The slave mode is disabled; The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high. 001: Reserved 010: Reserved 011: Reserved

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.
Because CI0F_ED outputs 1 pulse for each transition on CI0F, and the pause mode checks the level of the trigger signal, when CI0F_ED is selected as the trigger input, the pause mode must not be used.

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			CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	Reserved		CH1IE	CH0IE	UPIE
	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
13:11	Reserved	Must be kept at reset value
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
4:3	Reserved	Must be kept at reset value
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					CH1OF	CH0OF	Reserved	BRKIF	TRGIF	CMTIF	Reserved.		CH1IF	CH0IF	UPIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
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 channel's commutation event occurs, and cleared by software 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4:3	Reserved	Must be kept at reset value
2	CH1IF	Channel 1 's capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 's capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When 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 on an update event 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								BRKG	TRGG	CMTG	Reserved		CH1G	CH0G	UPG
								w	w	w			w	w	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	BRKG	Break event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is 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 is 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, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel's c/c control update event
4:3	Reserved	Must be kept at reset value
2	CH1G	Channel 1's capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	Update event generation This bit can be set by software, and cleared by hardware automatically. When this

bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) 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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		Reserved	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	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
31:15	Reserved	Must be kept at reset value
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	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
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 mode 01: Channel 1 is programmed as input mode, IS1 is connected to CI1FE1 10: Channel 1 is programmed as input mode, IS1 is connected to CI0FE1 11: Channel 1 is programmed as input mode, IS1 is connected to ITS. **Note:** When CH1MS[1:0]=11, it is necessary to select an internal trigger input through TRGS bits in TIMERx_SMCFG register.
7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control

This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV.
010: Clear the channel output. 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 mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise.
111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise.
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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(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 disable
1: Channel 0 output compare shadow enable
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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.

2 CH0COMFEN Channel 0 output compare fast enable
When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison.
0: Channel 0 output quickly compare disable.
1: Channel 0 output quickly compare enable.

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 mode
01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0
10: Channel 0 is programmed as input mode, IS0 is connected to CI1FE0
11: Channel 0 is programmed as input mode, IS0 is connected to ITS
Note: When CH0MS[1:0]=11, it is necessary to select an internal trigger input 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:

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 clear. 00: Prescaler disable, 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 2 (TIMERx_CHCTL2)

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								CH1NP	Reserved	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
								rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	CH1NP	Channel 1 complementary output 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 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 complementary output high level is active level 1: Channel 0 complementary output low level is active level When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is

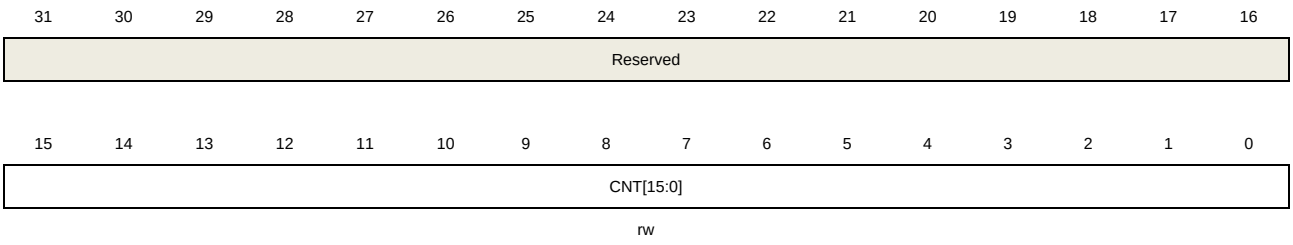
		11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled 1: Channel 0 complementary output enabled
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed 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).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CNT[15:0]	This bit-filed indicates the current counter value. Writing to this bit-filed 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-filed 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															

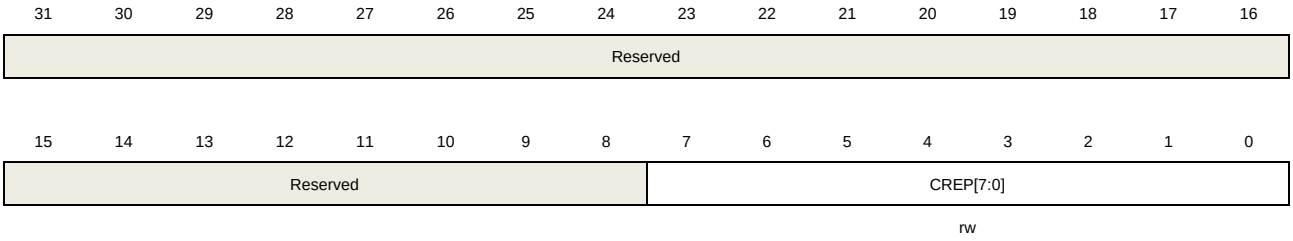
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter.

Counter repetition register (TIMERx_CREP)

Address offset: 0x30
Reset value: 0x0000 0000

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

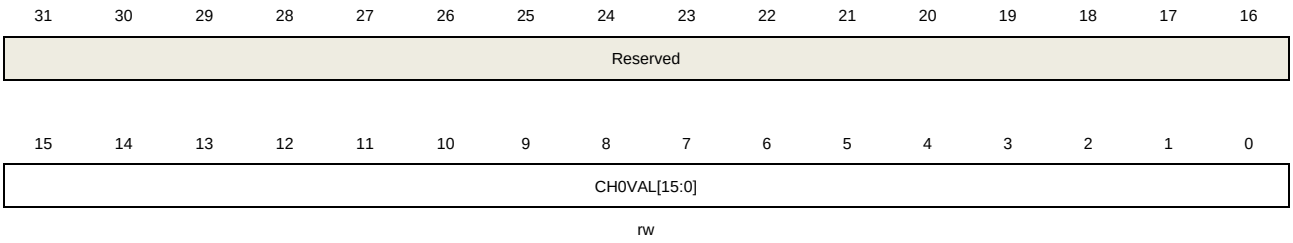


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP[7:0]	Counter repetition value This bit-filed specifies the update event generation rate. Each time the repetition counter counting down to zero, an update event is generated. The update rate of the shadow registers is also affected by this bit-filed 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).



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 (asynchronous). When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN 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: The 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-filed in TIMERx_CCHP register is 00.
13	BRKP	Break polarity This bit specifies the polarity of the BRKIN input signal. 0: BRKIN input active low 1: BRKIN input active high
12	BRKEN	Break enable This bit can be set to enable the BRKIN 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-filed 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. 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. 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-filed specifies the write protection property of registers.
00: protect disable. No write protection.
01: PROT mode 0.The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected.
10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected.
11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN bits in TIMERx_CHCTL0 registers (if the related channel is configured in output) are writing protected.
This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.

7:0 DTCFG[7:0] Dead time configure

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

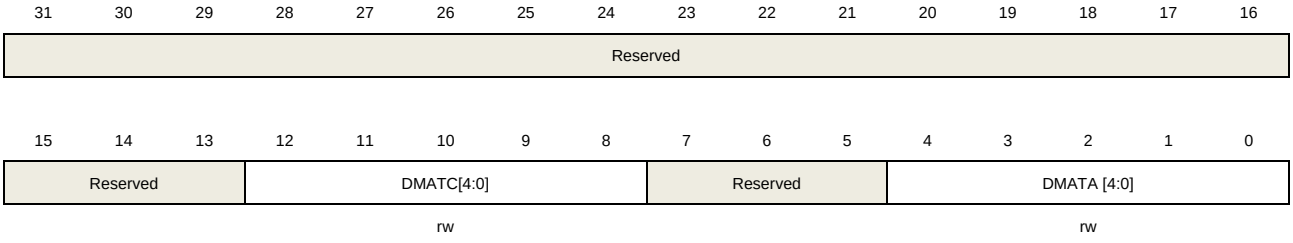
Note:

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.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0x48
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed is defined the number of DMA will access(R/W) the register of

		TIMERx_DMATB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed 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.

DMA transfer buffer register (TIMERx_DMATB)

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
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.

15.5. General level4 timer (TIMERx, x=15,16)

15.5.1. Overview

The general level4 timer module (TIMER15,TIMER16) is a one-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 general level4 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level4 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 issuitable for motor control applications.

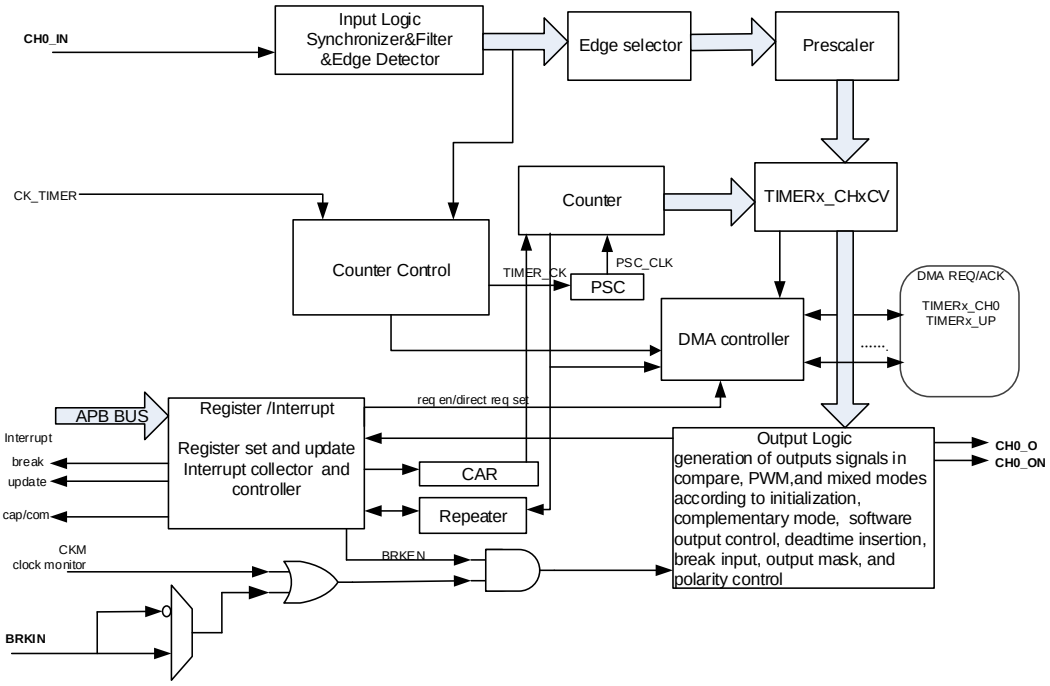
15.5.2. Characteristics

- Total channel num: 1.
- Counter width: 16 bit.
- Source of counter clock: internal clock.
- Counter modes: count up only.
- Programmable prescaler: 16 bit.The factor can be changed on the go.
- Each channel is user-configurable:
  input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request on: update, compare/capture event, and break input.

15.5.3. Block diagram

Figure 15-70. General level4 timer block diagram provides details of the internal configuration of the general level4 timer.

Figure 15-70. General level4 timer block diagram



15.5.4. Function overview

Clock source configuration

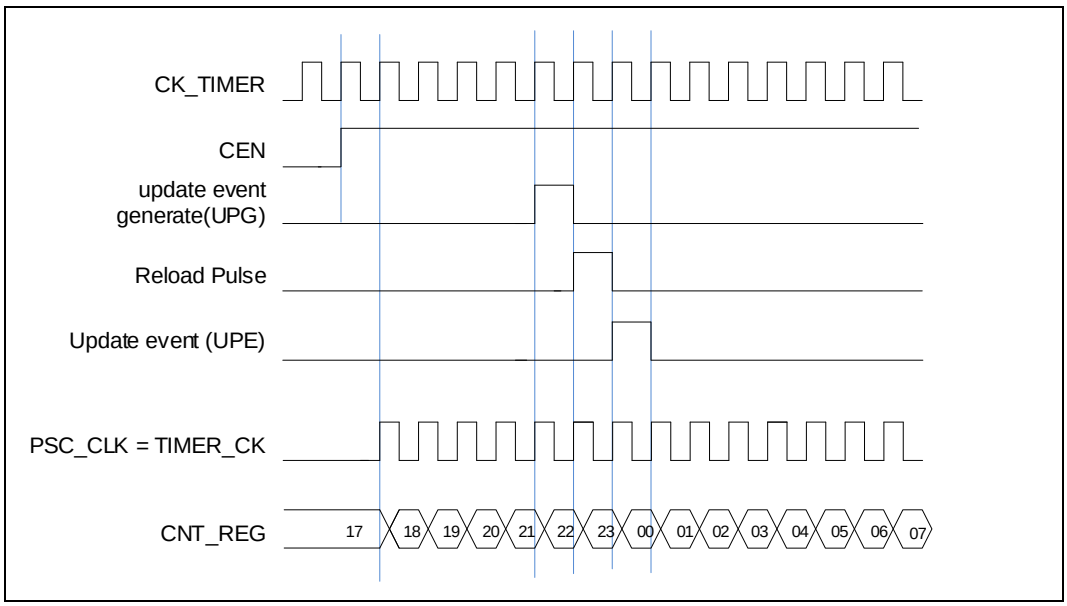
The general level4 TIMER can only being clocked by the CK_TIMER.

- Internal timer clock CK_TIMER which is from module RCU

The general level4 TIMER has only one clock source which is the internal 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.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

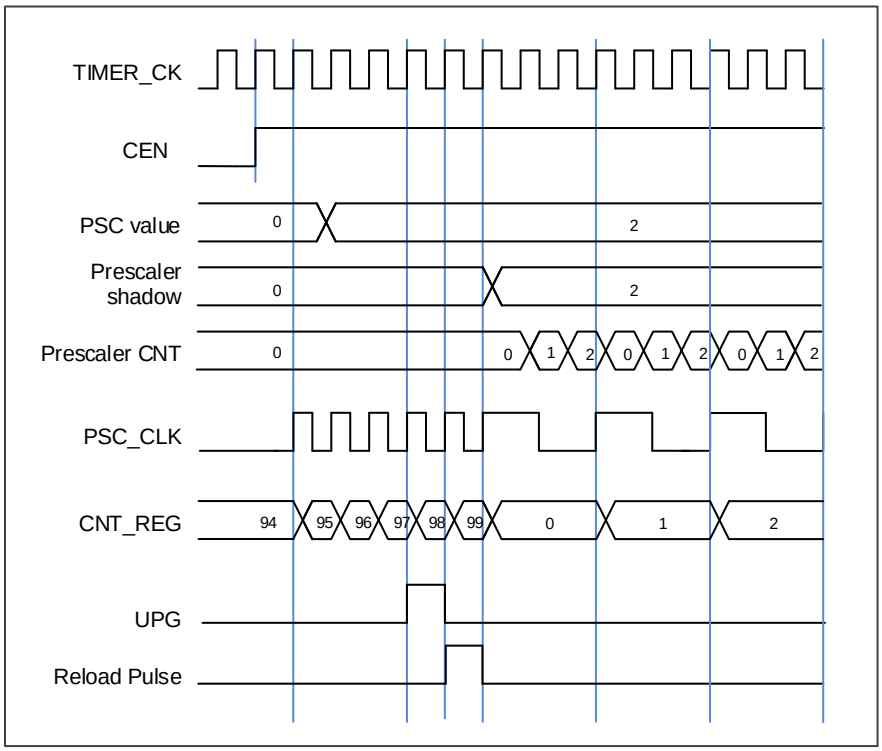
Figure 15-71. Timing chart of internal clock divided by 1



Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale 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 15-72. 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 generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter, counter 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 15-73. Timing chart of up counting mode, PSC=0/2

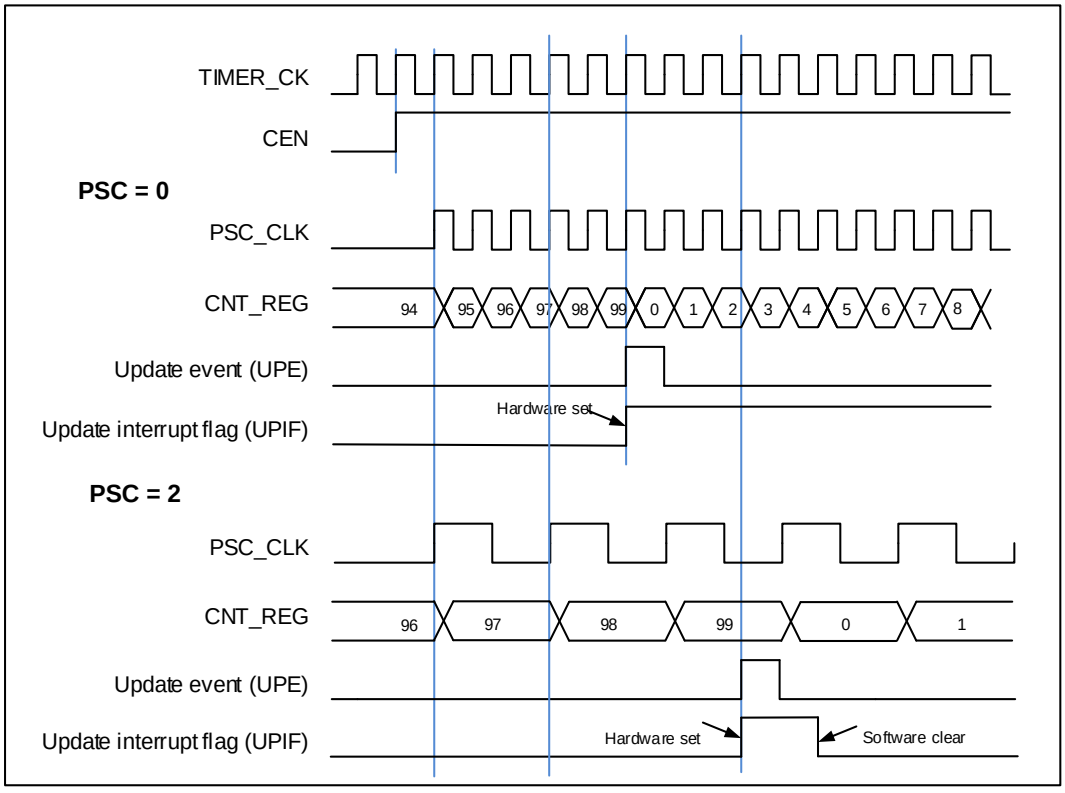
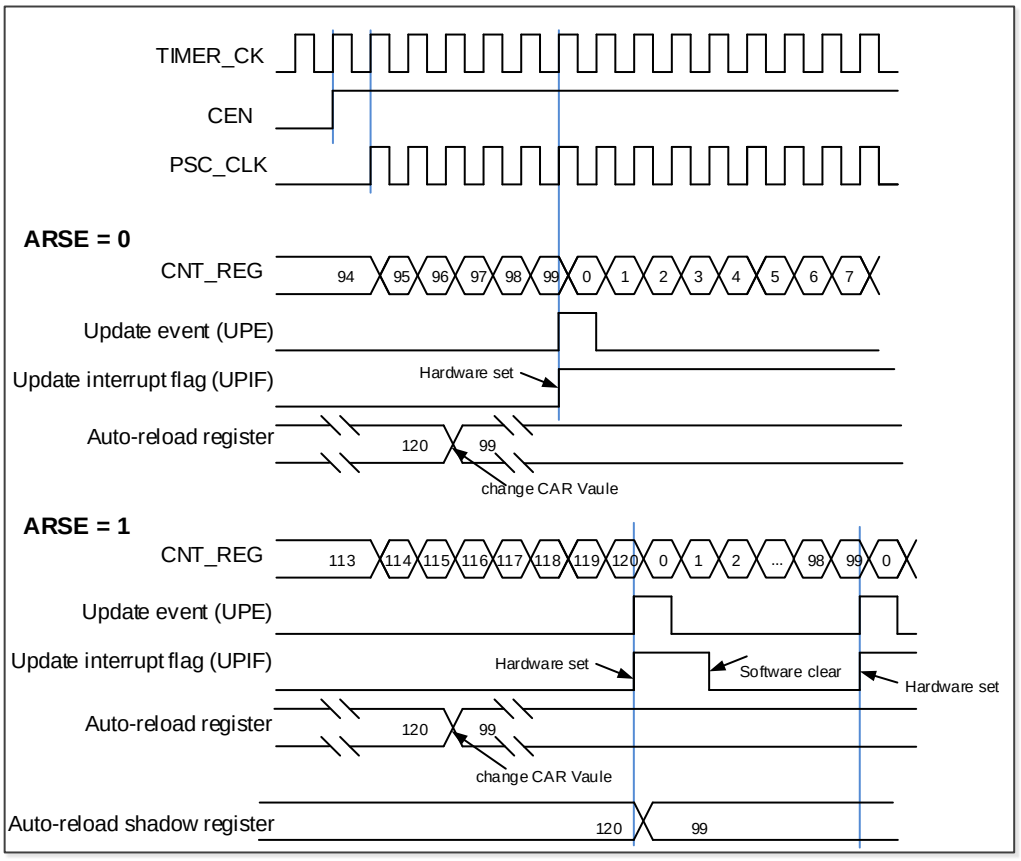


Figure 15-74. Timing chart of up counting mode, change TIMERx_CAR on the go

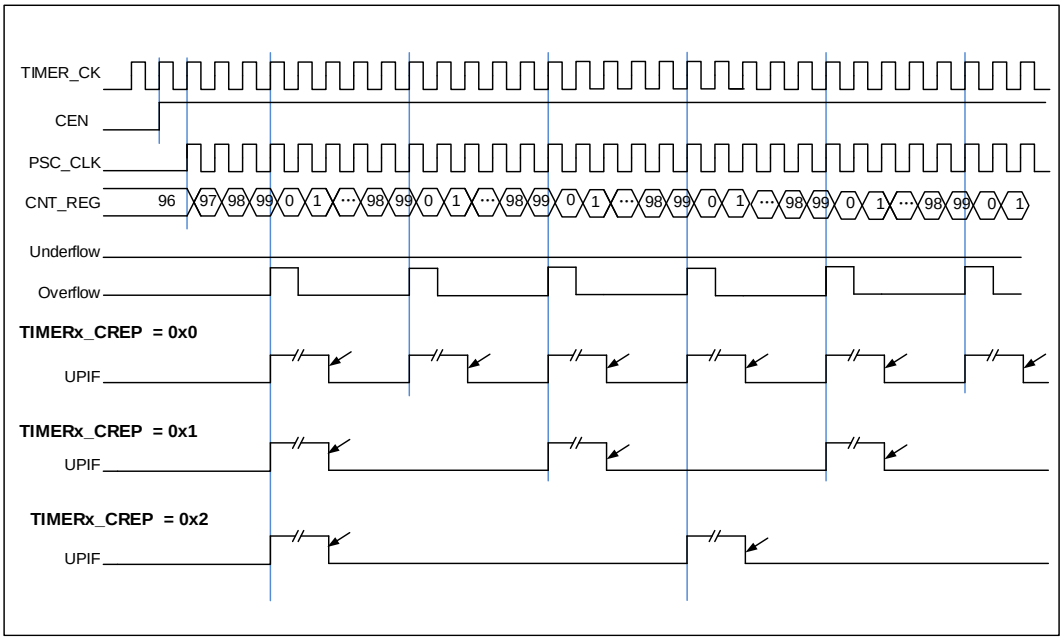


Update event (from overflow/underflow) rate configuration

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 in up counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generator an update event.

Figure 15-75. Repetition counter timing chart of up counting mode



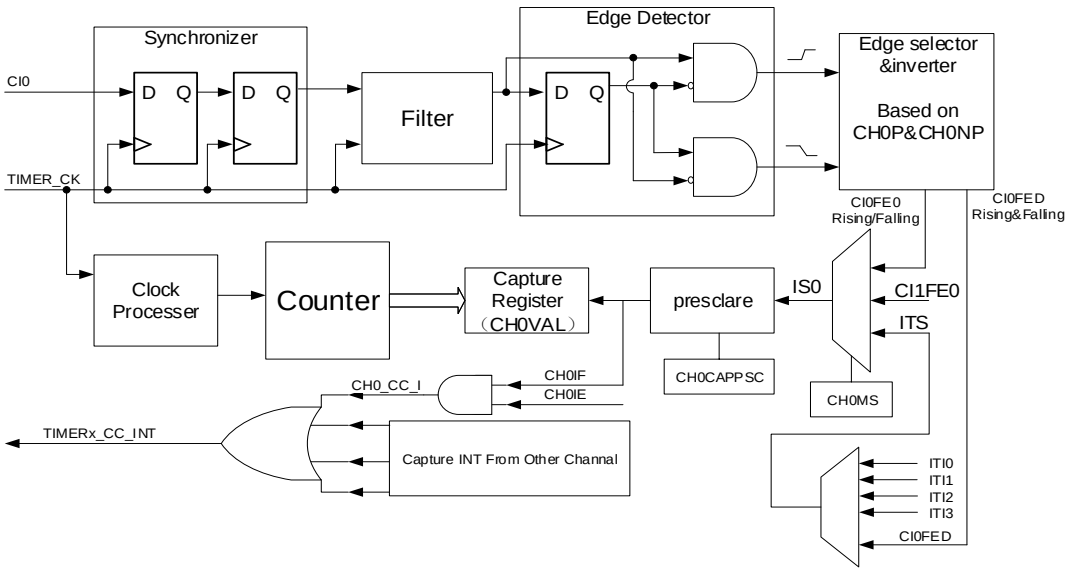
Input capture and output compare channels

The general level4 timer has one 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 enabled by CHxIE = 1.

Figure 15-76. Channel input capture principle



Channels' input signals (CIx) is the TIMERx_CHx signal. First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and falling edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So the process can be divided to several steps as below:

Step1: Filter configuration. (CHxCAPFLT in TIMERx_CHCTL0)

Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge selection. (CHxP/CHxNP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel 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 enable; you can got the interrupt and DMA request.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2)

Result: when you wanted input signal is got, TIMERx_CHxCV will be set by counter's value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt and DMA request will be asserted based on the configuration of CHxIE and CHxDEN in TIMERx_DMAINTEN

Direct generation: if you want to generate a DMA request or Interrupt, you can set CHxG by software directly.

- **Channel output compare function**

The channel input capture function, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the CHxVAL 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 CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be assert, if CHxDEN =1.

So the process can be divided to 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 high polarity by CHxP/CHxNP
* Enable the output by CHxEN

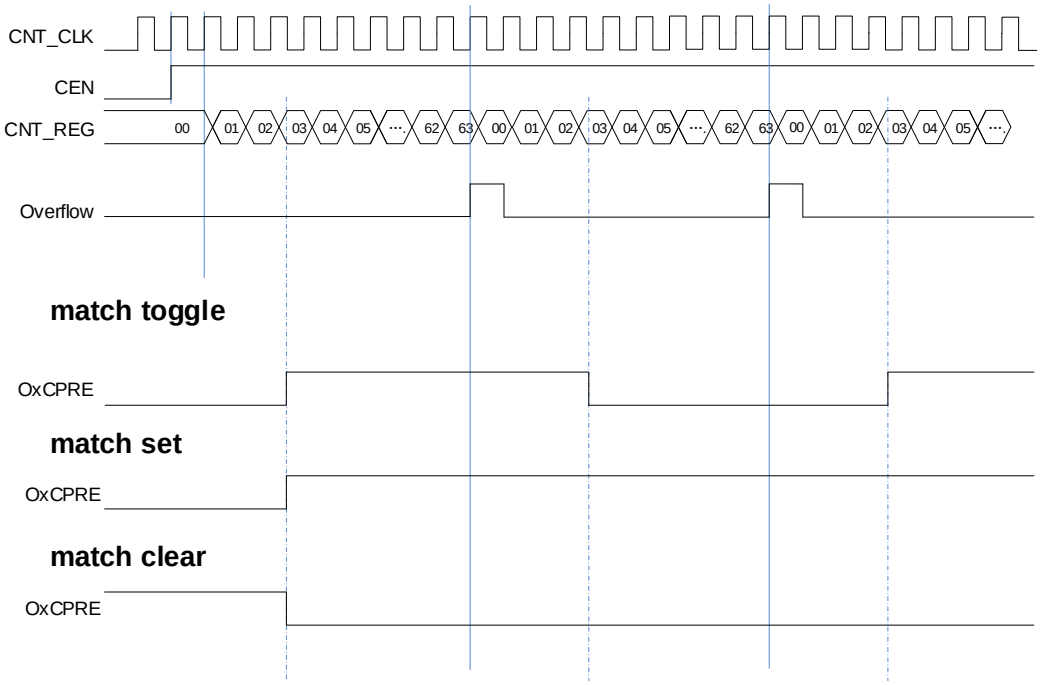
Step3: Interrupt/DMA-request enables configuration by CHxIE/CHxDEN

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV
About the CHxVAL; you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

Figure 15-77. Output-compare under three modes show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 15-77. Output-compare under three modes



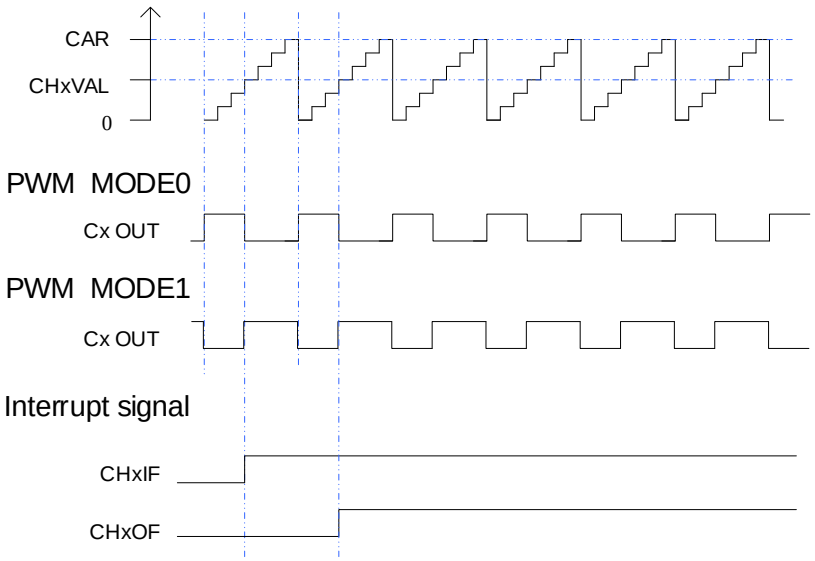
Output PWM function

In the output PWM function (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

The period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. ***Figure 15-78. PWM mode timechart*** shows the PWM output mode and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).
And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3'b110).

Figure 15-78. PWM mode timechart



Channel output prepare signal

When the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL filed. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting 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. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

Channel output complementary PWM

Function of complementary is for a pair of CHx_O and CHx_ON. Those two output signals cannot be active at the same time. The TIMERx has only 1 channel have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register and the POEN, ROS, IOS, ISOx and ISOxN bits in the TIMERx_CCHP and TIMERx_CTL1 registers. The outputs polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 15-10. Complementary outputs controlled by parameters

<table>
<tr><th colspan="5">Complementary Parameters</th><th colspan="2">Output Status</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON output disable(1).</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">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)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">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.</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="6">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O output disable.</td><td>CHx_ON =OxCPRE⊕(4)CHxNP
CHx_ON output enable.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON = LOW
CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON =(!OxCPRE)(5)⊕CHxNP.
CHx_ON output enable.</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O output “off-state”.</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O output “off-state”</td><td>CHx_ON =OxCPRE⊕CHxNP
CHx_ON output enable</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON =(!OxCPRE)⊕CHxNP
CHx_ON output enable.</td></tr>
</table>

Note:

(3) 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 both CHxEN and CHxNEN are 1'b1, and set POEN is also necessary. The field named DTCFG defines the dead time delay that can be used for channel 1. The detail about the delay time, refer to the register TIMERx_CCHP.

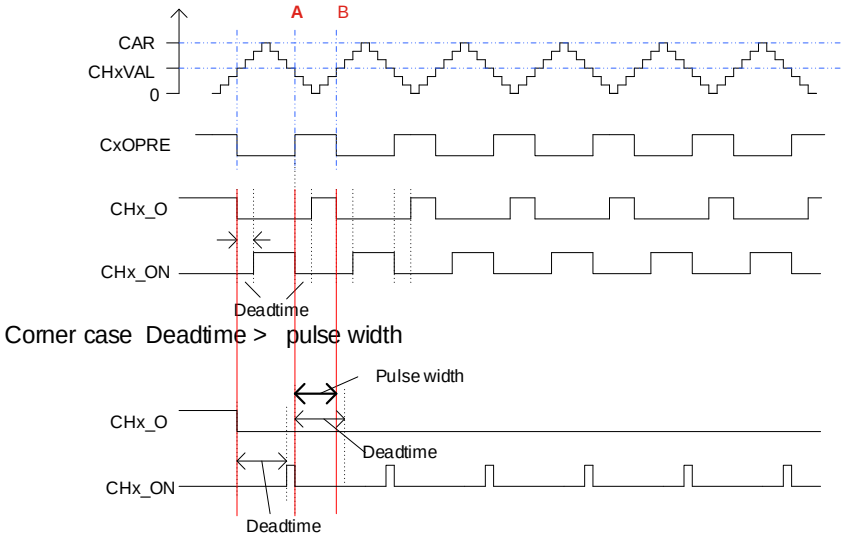
The dead time delay insertion ensures that no two complementary signals drive the active state at the same time.

When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the ***Figure 15-79. Channel output complementary PWM with dead-time insertion*** CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at once. Similarly, At point B when counter match (counter = CHxVAL) occurs again, OxCPRE is cleared, CHx_O signal will be cleared at once, while CHx_ON signal remains at the low value 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 CHx_O duty cycle, then the CHx_O signal is always the inactive value. (as show in the ***Figure 15-79. Channel output complementary PWM with dead-time insertion***)
- The dead time delay is greater than or equal to the CHx_ON duty cycle, then the CHx_ON signal is always the inactive value.

Figure 15-79. Channel output complementary PWM with dead-time insertion



Break mode

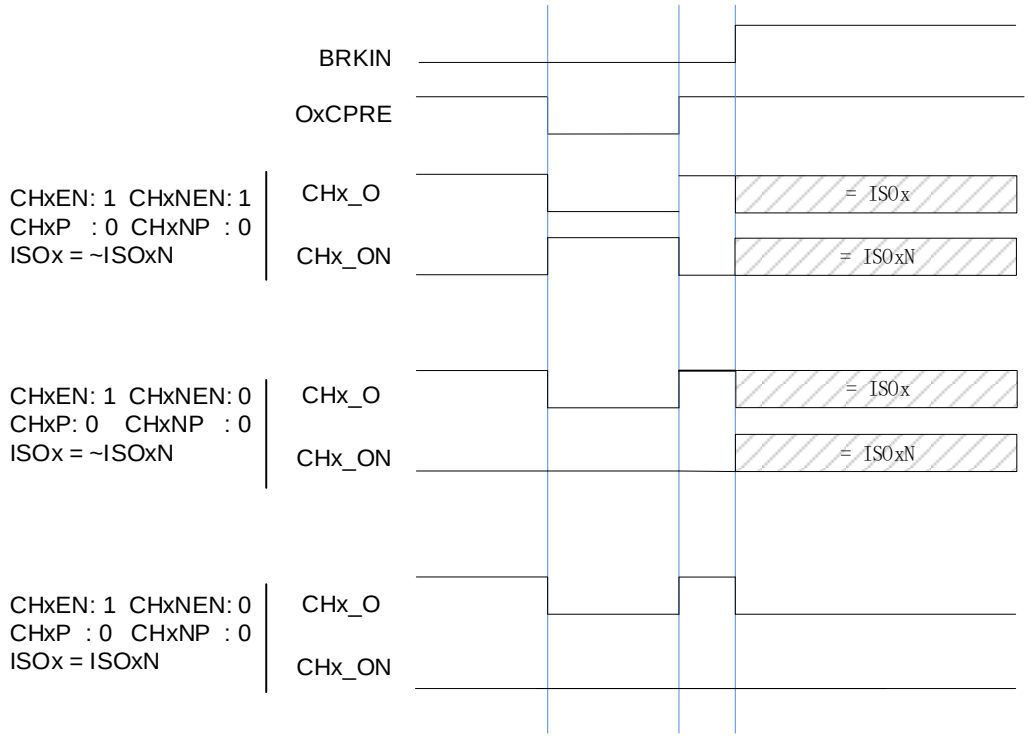
In this mode, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot be set both to active level when break occurs. The break sources are input break pin and

HXTAL stuck event by Clock Monitor (CKM) in RCU. The break function enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is setting by the BRKP bit in TIMERx_CCHP.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order 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 is set. If BRKIE is 1, an interrupt generated.

Figure 15-80. Output behavior in response to a break(The break high active)



Single pulse mode

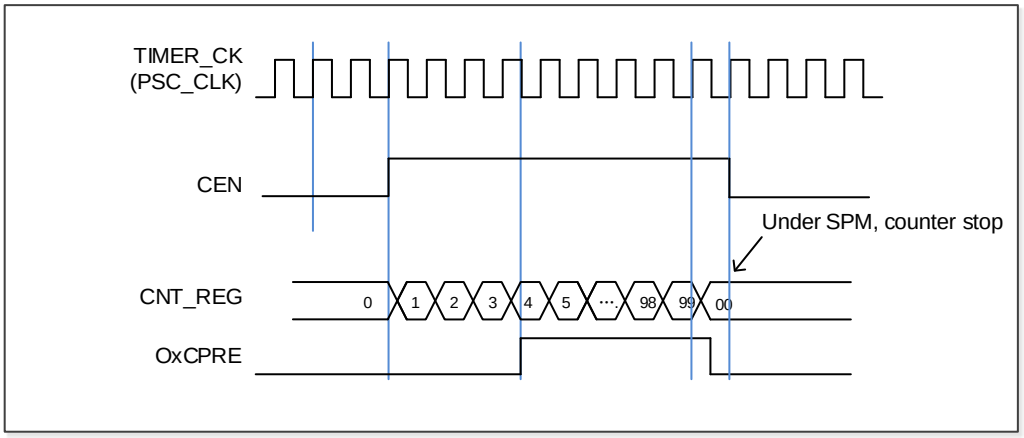
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

Once the timer is set to 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. Setting the CEN bit to 1 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 using software, the counter will be stopped and its value held.

In the single pulse mode, the trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0 register. 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. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 15-81. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 shows an example.

Figure 15-81. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timer DMA mode

Timer's DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Of course, you have to enable a DMA request which will be asserted by some internal event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB 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), then the timer's DMA request is finished. 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, DMATA+0xc at the next 3 accesses to TIMERx_DMATB. In one word, one time DMA internal interrupt event assert, DMATC+1 times request will be send by TIMERx.

If one more time DMA request event coming, TIMERx will repeat the process as above.

Timer debug mode

When the Cortex®-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

15.5.5. TIMERx registers(x=15,16)

TIMER15 base address: 0x4001 4400
TIMER16 base address: 0x4001 4800

Control register 0 (TIMERx_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						CKDIV[1:0]		ARSE	Reserved			SPM	UPS	UPDIS	CEN
						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
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. 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 update interrupts or DMA requests: 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, pause mode and quadrature 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						ISO0N	ISO0	Reserved				DMAS	CCUC	Reserved	CCSE
						rw	rw					rw	rw		rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9	ISO0N	Idle state of channel 0 complementary output 0: When POEN bit is reset, CH0_ON is set low. 1: When POEN bit is reset, CH0_ON 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 CH0_ON is implemented. This bit

		can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
7:4	Reserved	Must be kept at reset value
3	DMAS	DMA request source selection 0: DMA request of channel x is sent when capture/compare event occurs. 1: DMA request of channel x is sent when update event occurs.
2	CCUC	Commutation control shadow register update control When the commutation control shadow enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below: 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by 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, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event coming. When a channel does not have a complementary output, this bit has no effect.

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	31
Reserved						CH0DEN	UPDEN	BRKIE	Reserved	CMTIE	Reserved			CH0IE	UPIE
						rw	rw	rw		rw				rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
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	Reserved	Must be kept at reset value
5	CMTIE	Commutation interrupt enable 0: disabled 1: enabled
4:2	Reserved	Must be kept at reset value
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).



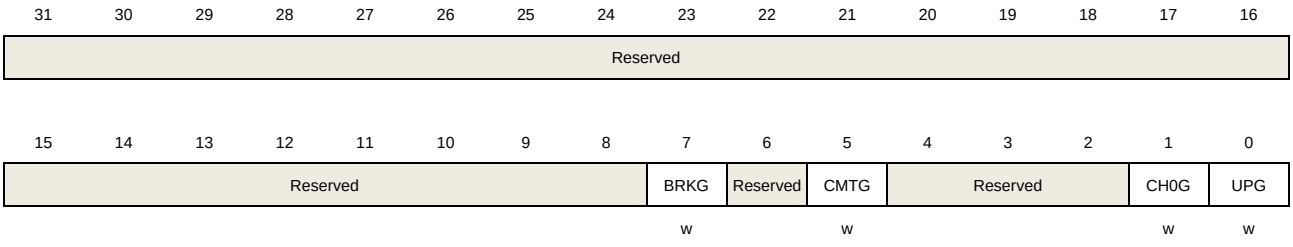
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt 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	Reserved	Must be kept at reset value
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when channel's commutation event occurs, and cleared by software 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4:2	Reserved	Must be kept at reset value
1	CH0IF	Channel 0 's capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When 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 on an update event 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
31:8	Reserved	Must be kept at reset value

7	BRKG	Break event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is set, related interrupt or DMA transfer can occur if enabled. 0: No generate a break event 1: Generate a break event
6	Reserved	Must be kept at reset value
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, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel's c/c control update event
4:2	Reserved	Must be kept at reset value
1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	Update event generation This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) 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
Reserved															

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

Reserved	Reserved	CH0COMCTL[2:0]	CH0COMSEN	Reserved	CH0MS[1:0]
	CH0CAPFLT[3:0]		CH0CAPPSC[1:0]		
		rw	rw		rw

Output compare mode:

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP 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. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. 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 mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(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 disable 1: Channel 0 output compare shadow enable 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-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
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 mode 01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0 10: Reserved 11: Reserved

Input capture mode:

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
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 clear. 00: Prescaler disable, 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 2 (TIMERx_CHCTL2)

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												CH0NP	CH0NEN	CH0P	CH0EN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 complementary output high level is active level 1: Channel 0 complementary output low level is active level When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled 1: Channel 0 complementary output enabled
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 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or

		trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed 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-filed indicates the current counter value. Writing to this bit-filed 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-filed 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-filed specifies the auto reload value of the counter.

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-filed specifies the update event generation rate. Each time the repetition counter counting down to zero, an update event is generated. The update rate of the shadow registers is also affected by this bit-filed 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 or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates 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 (asynchronous). When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN 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: The 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-filed in TIMERx_CCHP register is 00.
13	BRKP	Break polarity This bit specifies the polarity of the BRKIN input signal. 0: BRKIN input active low 1: BRKIN input active high
12	BRKEN	Break enable This bit can be set to enable the BRKIN 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-filed 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. 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.

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-filed specifies the write protection property of registers.

00: protect disable. No write protection.

01: PROT mode 0.The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected.

10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected.

11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN bits in TIMERx_CHCTL0 registers (if the related channel is configured in output) are writing protected.

This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.

7:0 DTCFG[7:0] Dead time configure

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.

DMA configuration register (TIMERx_DMACFG)

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
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]				

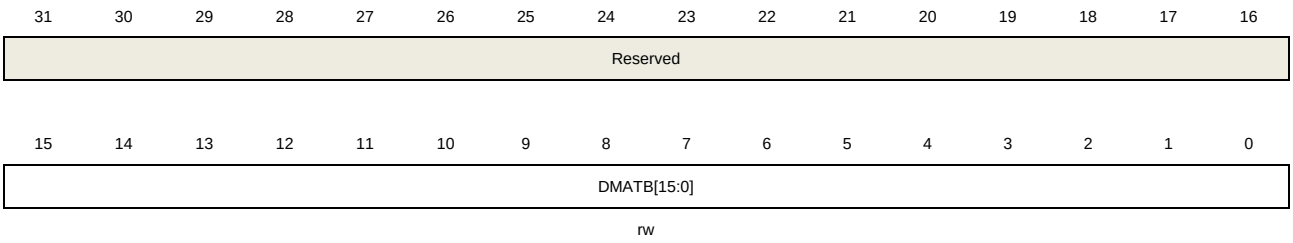
rw rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed is defined the number of DMA will access(R/W) the register of TIMERx_DMATB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed 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.

DMA transfer buffer register (TIMERx_DMATB)

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
15:0	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.

15.6. Basic timer (TIMERx, x=5)

The basic timer is only available on GD32F150 series.

15.6.1. Overview

The basic timer module (TIMER5) reference is a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate DMA request and TRGO to DAC.

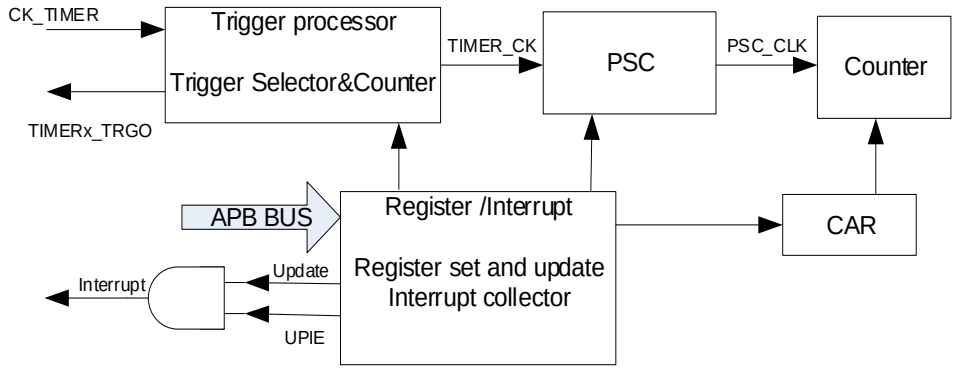
15.6.2. Characteristics

- Counter width: 16bit.
- Source of count clock is internal clock only.
- Counter modes: only count up.
- Programmable prescaler: 16 bit. Factor can be changed on the go.
- Auto-reload function.
- Interrupt output or DMA request on update event.

15.6.3. Block diagram

Figure 15-82. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 15-82. Basic timer block diagram



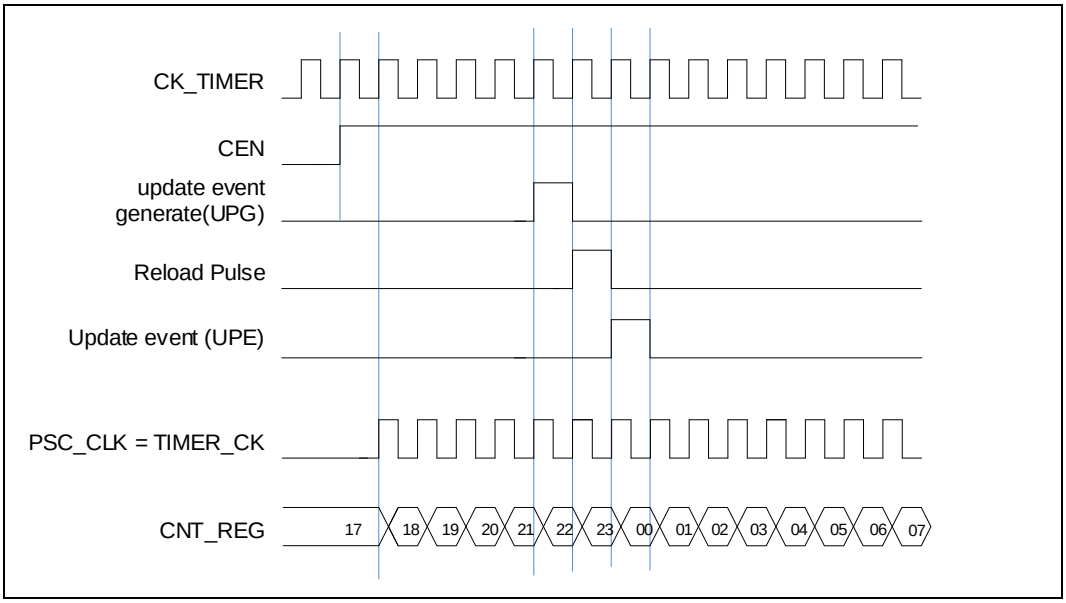
15.6.4. Function overview

Clock source configuration

The basic TIMER can only being 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.

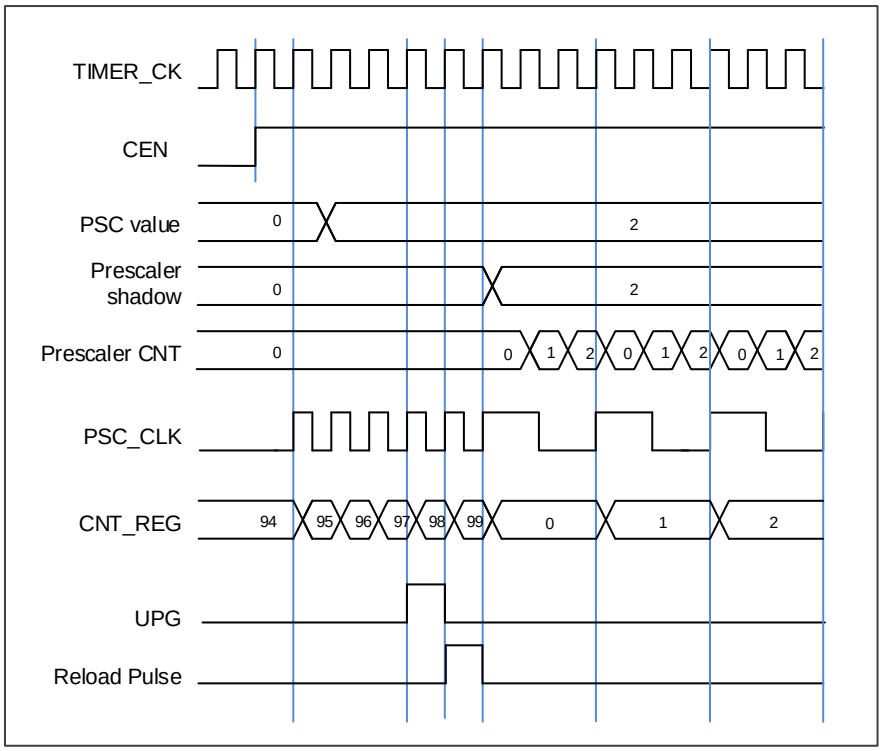
Figure 15-83. Timing chart of internal clock divided by 1



Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale 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 15-84. 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 (counter 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 15-85. Timing chart of up counting mode, PSC=0/2

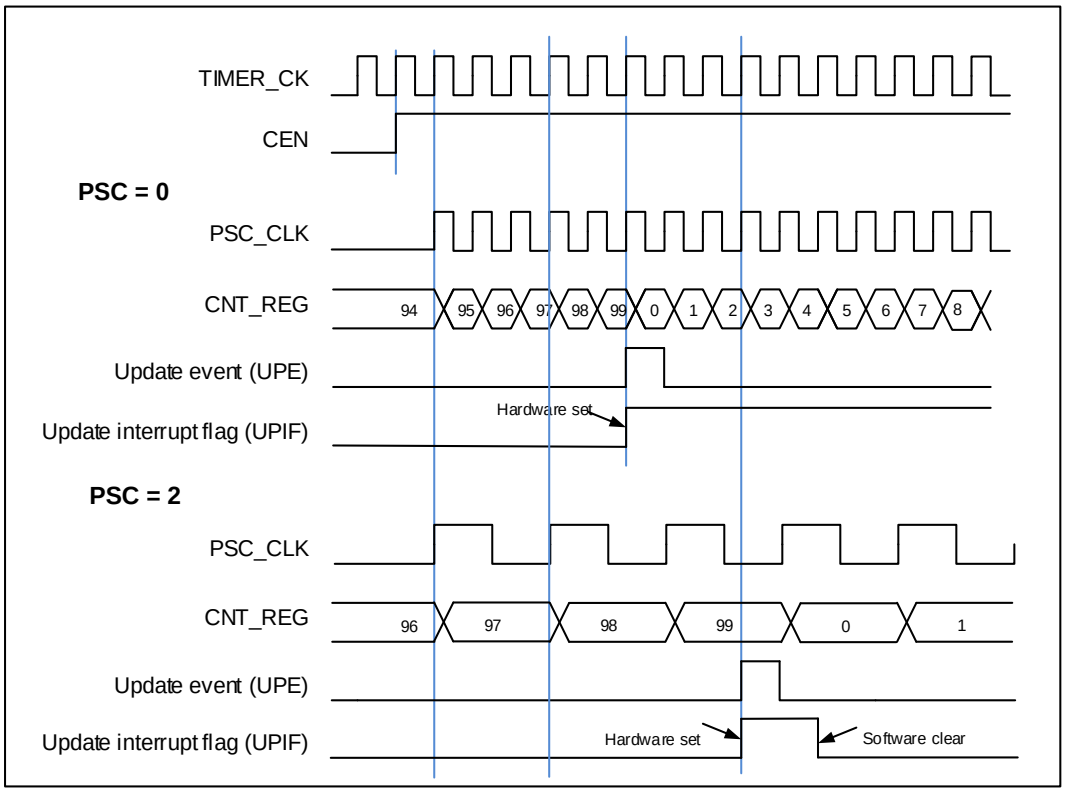
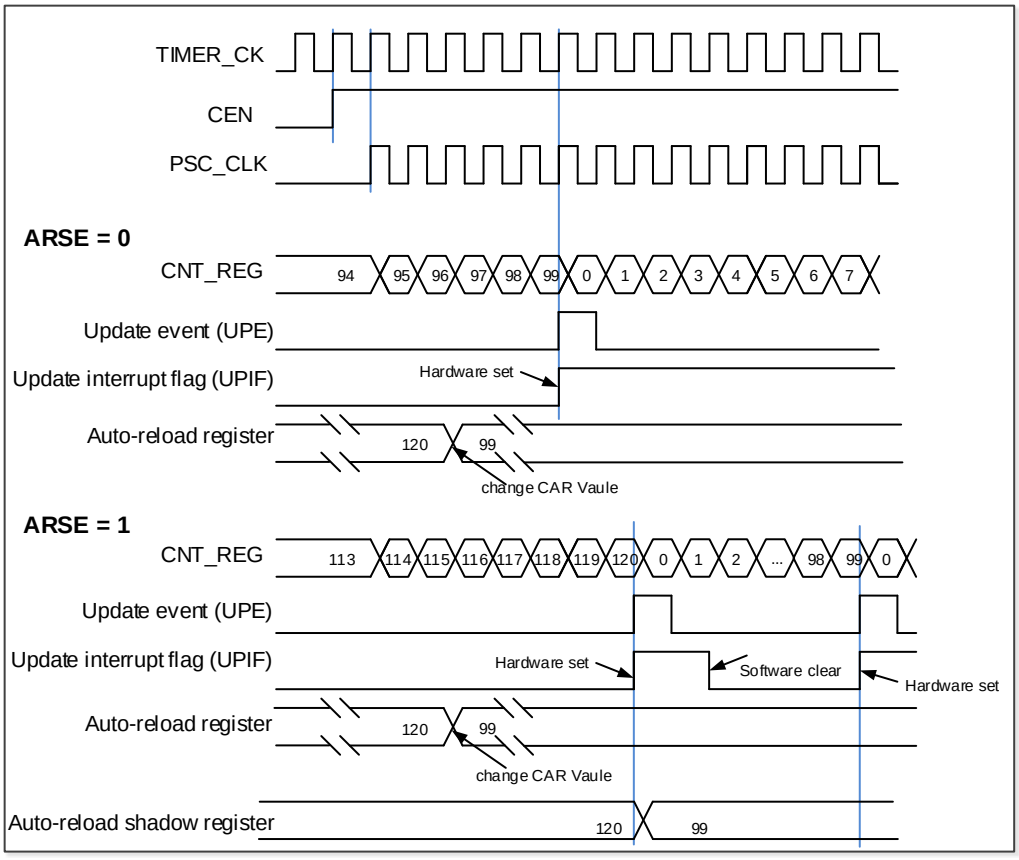


Figure 15-86. Timing chart of up counting mode, change TIMERx_CAR on the go



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®-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

15.6.5. TIMERx registers(x=5)

TIMER5 base address: 0x4000 1000

Control register 0 (TIMERx_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								ARSE	Reserved			SPM	UPS	UPDIS	CEN
								rw				rw	rw	rw	rw

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 disable. The counter continues after update event. 1: Single pulse mode enable. 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 update interrupts or DMA requests: 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, pause mode and quadrature 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 in master mode to slave timers for synchronization function. 000: When a counter reset event occurs, a TRGO trigger signal is output. The counter resert source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a conter 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.
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).

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							UPDEN	Reserved							UPIE
							rw								rw

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).

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

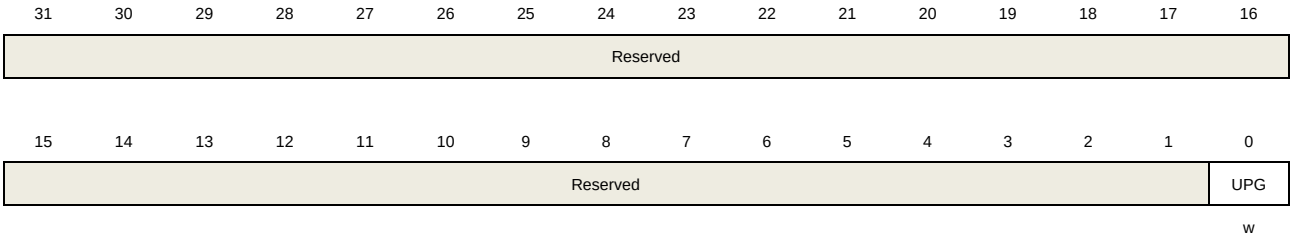
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UPIF
															rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPIF	Update interrupt flag This bit is set by hardware on an update event 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).

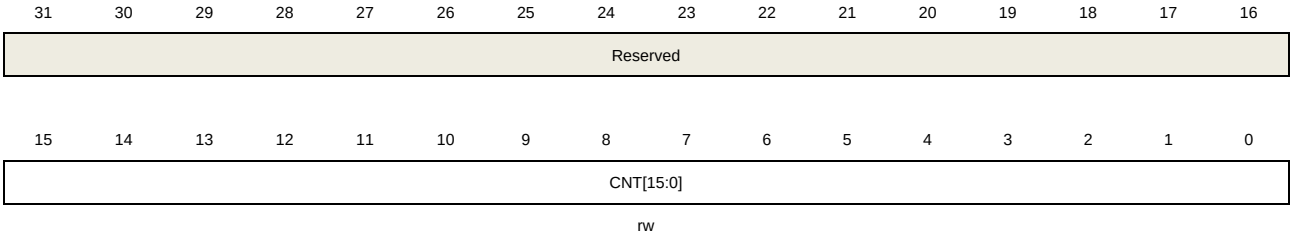


Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPG	This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

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-filed indicates the current counter value. Writing to this bit-filed 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-filed 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-filed specifies the auto reload value of the counter.

16. Infrared ray port (IFRP)

16.1. Overview

Infrared ray port (IFRP) is used to control infrared light LED, and send out infrared data to implement infrared ray remote control.

There is no register in this module, which is controlled by TIMER15 and TIMER16. You can improve the module's output to high current capacity by set the GPIO pin to Fast Mode.

16.2. Characteristics

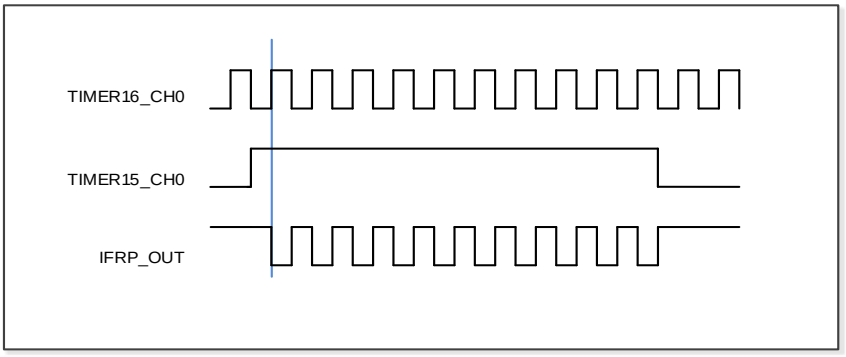
- The IFRP output signal is decided by TIMER15_CH0 and TIMER16_CH0
- To get correct infrared ray signal, TIMER15 should generate low frequence modulation envelope signal, and TIMER16 should generate high frequence carrier signal
- The IFRP output (PB9) can provide high current to control LED interface by setting PB9_HCCE in SYSCFG_CFG0

16.3. Function overview

IFRP is a module which is able to integrate the output of TIMER15 and TIMER16 to generate an infrared ray signal.

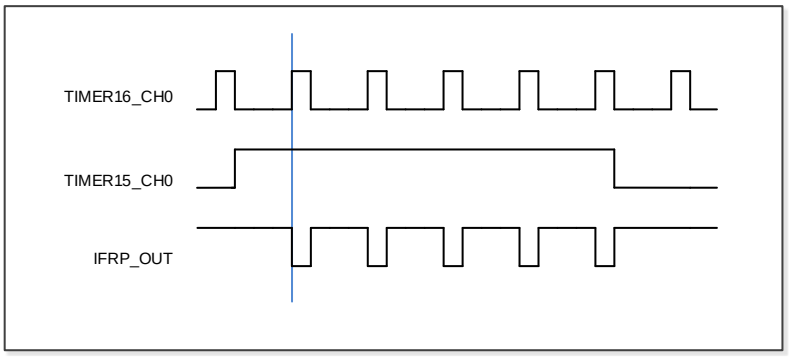
1. The TIMER15's CH0 is programed to generate the low frequence PWM signal which is the modulation evalope signal. The TIMER16's CH0 is programed to generate the high frquence PWM signal which is the carrier signal. And the channel need to be enabled before generating these signals.
2. Program the GPIO remap regisger and enable the pin.
3. If you want to get the high current capacity of output, remapping IFRP_OUT to PB9 and setting the PB9 to Fast Mode by the register in SYS_CFG module are required.

Figure 16-1. IFRP output timechart 1



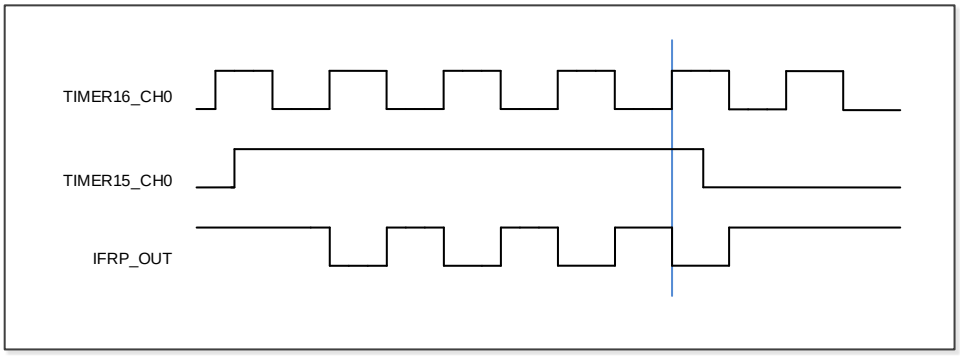
Note: IFRP_OUT has one APB clock delay from TIMER16_CH0.

Figure 16-2. IFRP output timechart 2



Note: Carrier (TIMER15_CH0)'s duty cycle can be changed, and IFRP_OUT has inverted relationship with TIMER16_CH0 when TIMER15_CH0 is high.

Figure 16-3. IFRP output timechart 3



Note: IFRP_OUT will keep the integrity of TIMER16_CH0, even if evelope signal (TIMER15_CH0) is no active.

17. Universal synchronous asynchronous receiver transmitter (USART)

17.1. 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 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 data bits and the TX/RX pins can be configured flexibly.

ALL USARTs support DMA function for high-speed data communication.

17.2. Characteristics

- NRZ standard format (Mark/Space)
- Asynchronous, full duplex communication
- Half duplex single wire communications
- Dual clock domain
  - Asynchronous pclk and USART clock
  - Baud rate programming independent from the PCLK reprogramming
- Programmable baud-rate generator allowing speed up to 9 MBits/s when the clock frequency is 72 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
  - 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 or address 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), 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 is fully implemented, USART1 is only partially implemented with the following features not supported:

- Smartcard mode
- IrDA SIR ENDEC block
- LIN mode
- Dual clock domain and wakeup from deep-sleep mode
- Receiver timeout interrupt
- Modbus communication

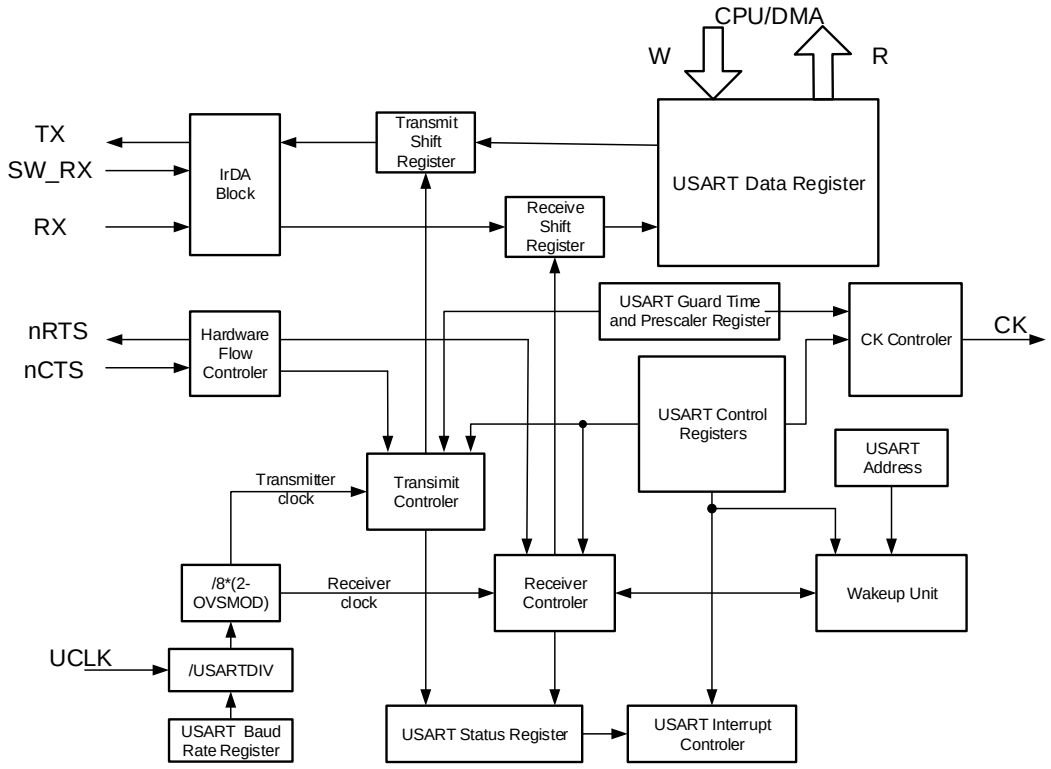
17.3. Function overview

The interface is externally connected to another device by the main pins listed as following.

Table 17-1. USART important pins description

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
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

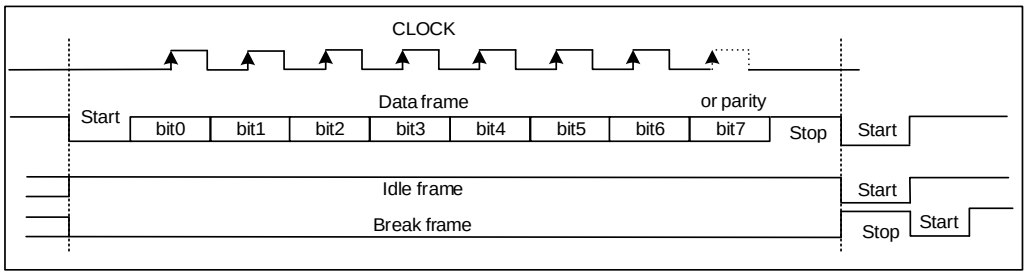
Figure 17-1. USART module block diagram



17.3.1. 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 17-2. 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 17-2. Stop bits configuration

STB[1:0]	stop bit length (bit)	usage description
00	1	default value
01	Reserved	Reserved
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.

A break frame is configured 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.

17.3.2. Baud rate generation

The baud-rate divisor is a 16-bit number consisting 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 to the UCLK:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (17\text{-}1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (17\text{-}2)$$

The choice of the USART clock (UCLK) is done through the Clock Control system (see the Reset and clock unit (RCU) section). The clock source must be chosen before enabling the USART (by setting the UEN bit).

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.

17.3.3. 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 be output through the CK pin.

In case of transmission corruption, the TEN bit should not be disabled when transmission is ongoing.

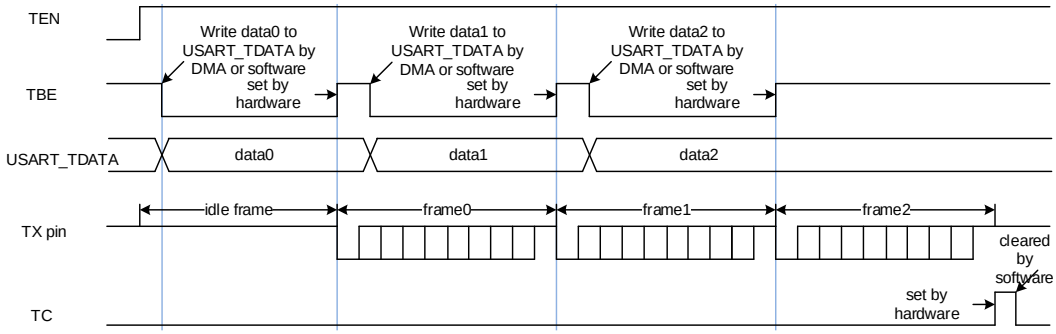
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit of the USART_STAT register is asserted. The TBE bit is cleared by a writing to the USART_TDATA register and will be set by hardware after the data is put into the transmit shift register. If a data 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 is generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

Refer to the following procedure for the USART transmission:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the stop bits length in USART_CTL1.
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. Wait until TC=1 to finish.

Figure 17-3. USART transmit procedure



It is necessary to wait for the TC bit asserted before disabling the USART or entering the power saving mode. The TC bit can be cleared by writing 1 to 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.

17.3.4. 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 are 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 bits of the received are stored in the USART_STAT register.

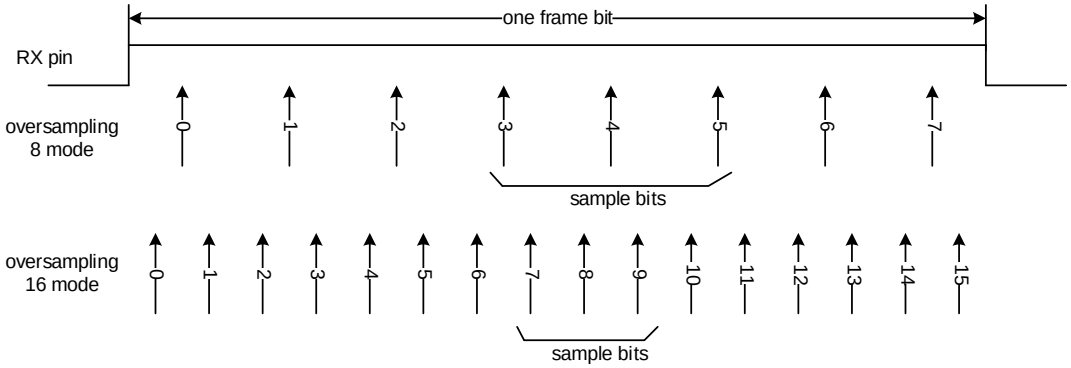
The software can get the received data by reading the USART_RDATA register directly, or through DMA. The RBNE status is cleared by a read operation on the USART_RDATA register, whatever it is performed by software directly, or through DMA.

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 are 0, the frame bit is confirmed as a 0, else 1. If the three samples of any bit of a frame are not the same, whatever it is a data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame. An interrupt is generated, If the receive DMA is enabled

and 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 17-4. 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 receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set.

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 receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set, or if the RBNEIE is set.

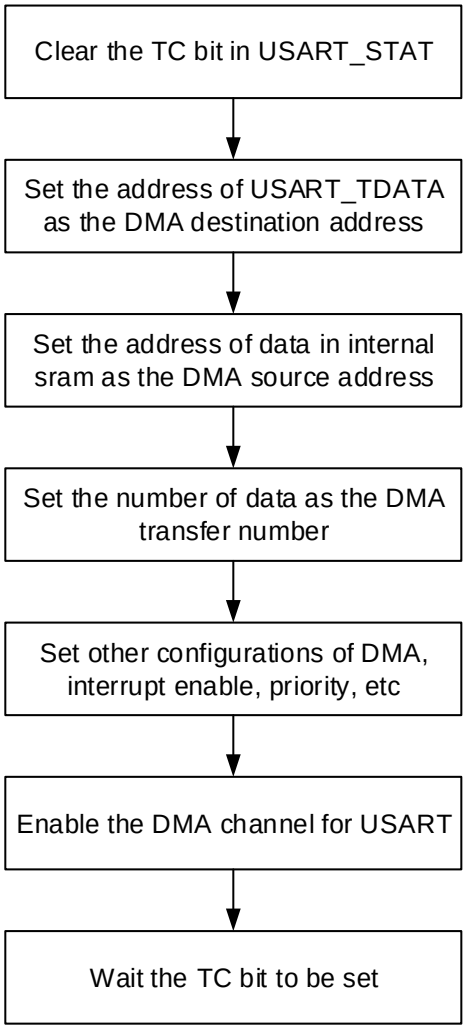
If a noise error (NERR), parity error (PERR), frame error (FERR) or overrun error (ORERR) occurs during reception, NERR, PERR, FERR or ORERR will be set at the same time with RBNE. If the receive DMA is not enabled, when the RBNE interrupt occurs, software need to check whether there is a noise error, parity error, frame error or overrun error.

17.3.5. 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 is shown in ***Figure 17-5. Configuration step when using DMA for USART transmission***

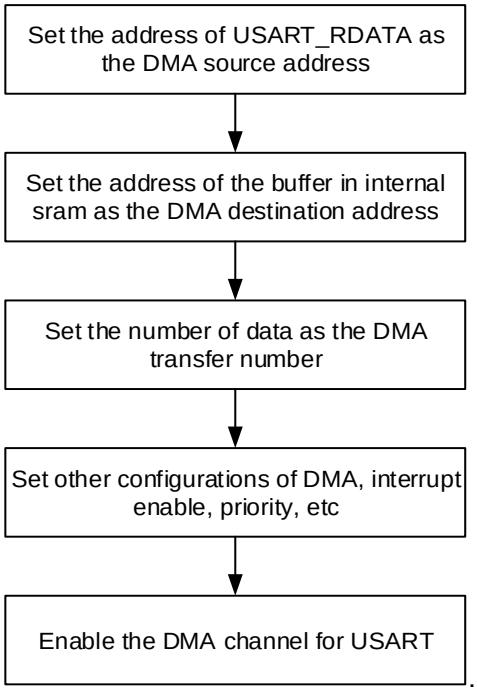
Figure 17-5. Configuration step when using DMA for USART transmission



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 step is shown in ***Figure 17-6. 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 17-6. Configuration step when using DMA for USART reception

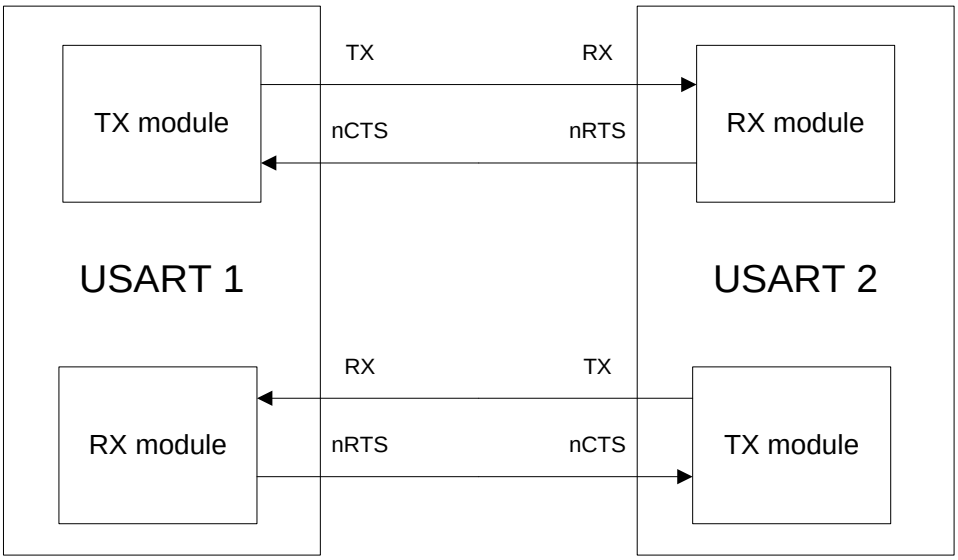


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

17.3.6. 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 17-7. Hardware flow control between two USARTs



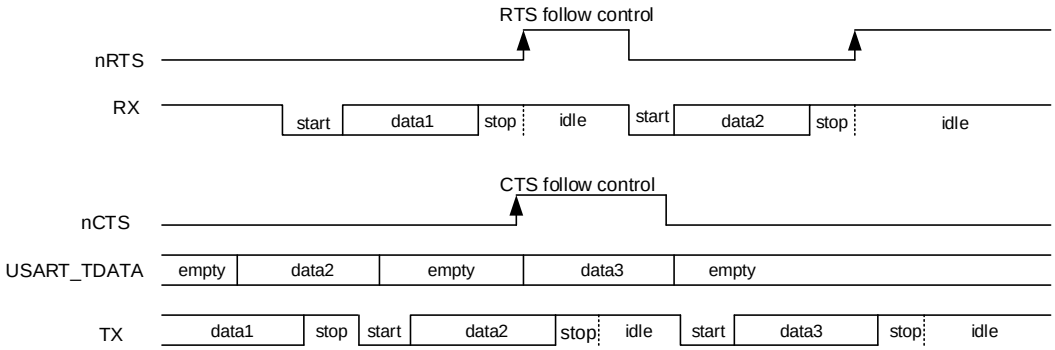
RTS flow control

USART receiver can receive data only when the nRTS signal is low, and the signal does not go high until the data frame reception is finished. The next reception occurs when the nRTS signal goes low again. The signal keeps high when the receive register is full.

CTS flow control

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 17-8. 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.

17.3.7. 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 two methods: idle frame method and address match method.

The idle frame wake up method is selected by default. When an idle frame is detected on the RX pin, the hardware clears the RWU bit and exits the mute mode. When wake up at an idle

frame, the IDLEF bit in USART_STAT is not set.

When the WM bit of in USART_CTL0 register is set, the MSB bit of a frame is detected as the address flag. 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 bits in the USART_CTL1 register, the hardware clears the RWU bit and exits the mute mode. The RBNE bit is set for the frame that wakes up the USART. The status bits are available in the USART_STAT register. If the LSB 4/7 bits of an address frame defers from the ADDR 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 flag. If the ADDM bit is set and the receive frame is a 7bit data, the LSB 6 bits will be compared with ADDR[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[7:0].

17.3.8. LIN mode

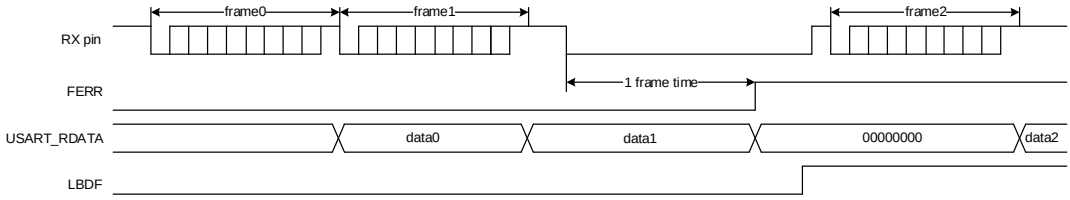
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be reset in LIN mode.

The LIN transmission procedure is almost the same as the normal transmission procedure. 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 from 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 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 in USART_STAT is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

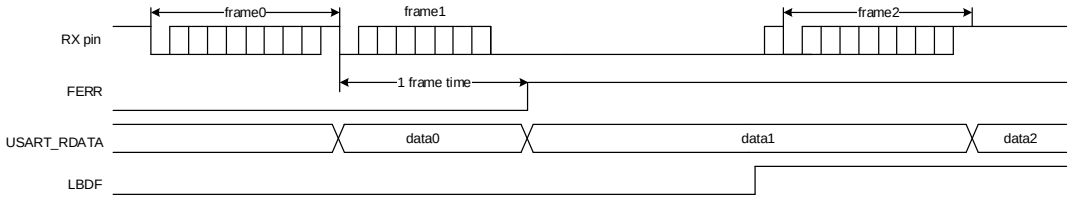
As shown in ***Figure 17-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all ‘0’ frame, with an asserted FERR status.

Figure 17-9. Break frame occurs during idle state



As shown in ***Figure 17-10. 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 17-10. Break frame occurs during a frame



17.3.9. 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 reset in synchronous mode. The CK pin is the synchronous USART transmitter clock output, and can be only activated when the TEN bit is enabled. No clock pulse will be sent to the CK pin during the start bit and stop bit transmission. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the LSB (address index) 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 mode idle state.

These 3 bits (CPL, CPH, and CLEN) should not be changed while the transmitter or the receiver is enabled

The clock is synchronized with the data transmitted. The receiver in synchronous mode samples the data on the transmitter clock without any oversampling.

Figure 17-11. Example of USART in synchronous mode

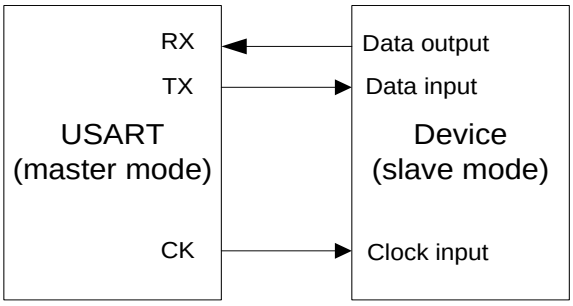
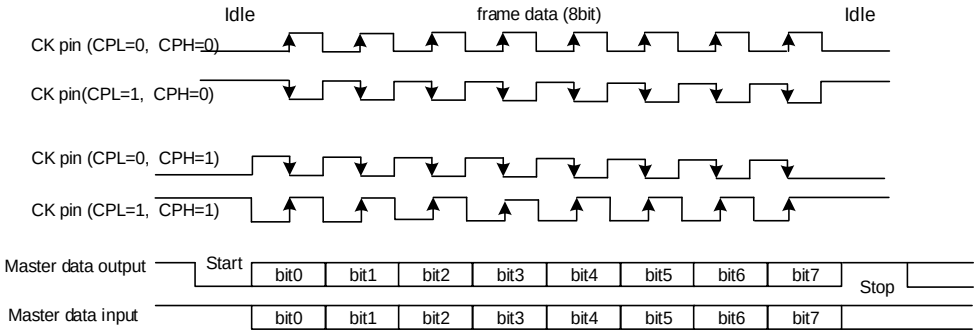


Figure 17-12. 8-bit format USART synchronous waveform (CLEN=1)

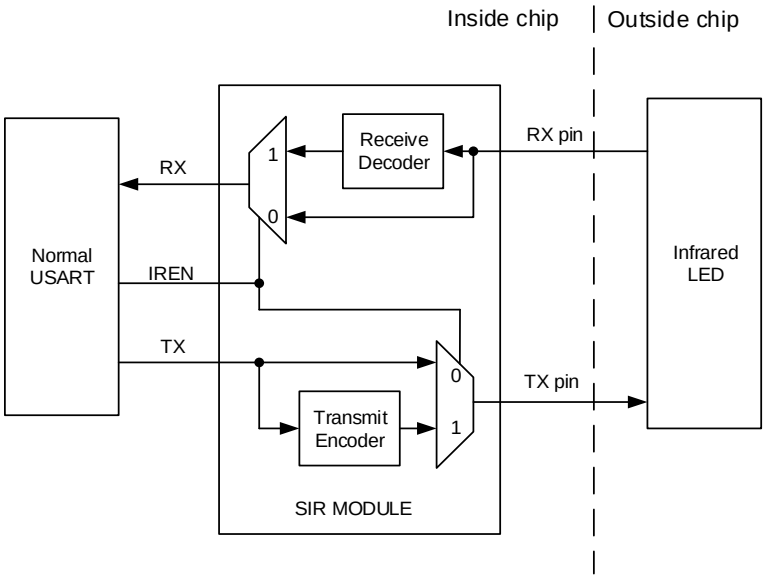


17.3.10. IrDA SIR ENDEC mode

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 reset in IrDA mode.

In IrDA mode, the USART transmission data frame is modulated in the SIR transmit quadrature decoder 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 quadrature decoder.

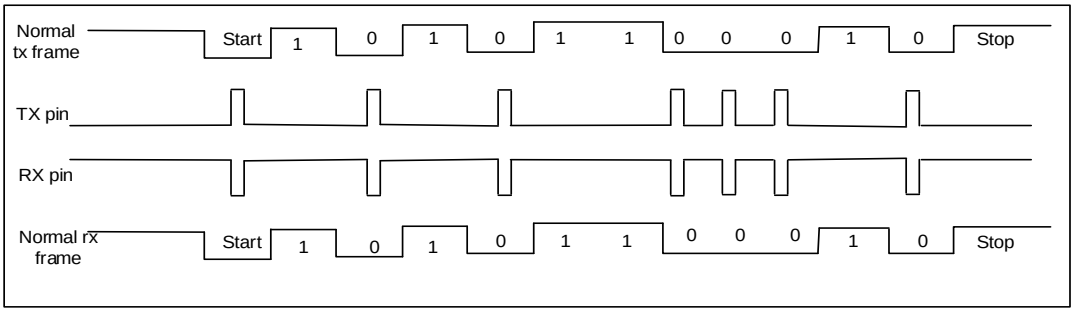
Figure 17-13. IrDA SIR ENDEC module



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 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 17-14. IrDA data modulation



The SIR submodule 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 divide ratio is configured by the PSC[7:0] bits in USART_GP register, USART_BAUD register needs to be configured as 16*PSC[7:0]. The pulse width on the TX pin is 3 cycles of this low speed clock. The receiver decoder works in the same manner as the normal IrDA mode.

17.3.11. 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 reset 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 the TEN bit is set, the data in the data register will be sent.

17.3.12. Smartcard (ISO7816-3) mode

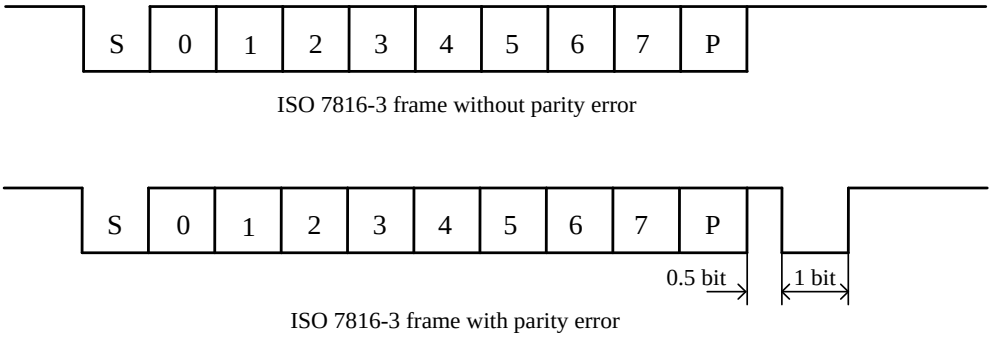
The smartcard mode is an asynchronous mode, which is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be reset in smartcard mode.

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 and drives a bidirectional line that is also driven by the smartcard.

Figure 17-15. ISO7816-3 frame format



T=0 mode

Comparing to the time in normal operation, the transmission time from transmit shift register to the TX pin is delayed half baud clock, and the TC flag assertion time delayed a certain value wrote in the guard time register. The USART can automatically re-send data according to the protocol by SCRTNUM times. At the end of reception of the last repeated character the TC bit is set without gardtime immediately. The USART will stop transmitting and signal the error as a framing error if it continues receiving the NACK after the programmed number of retries. The TXFCMD bit in the USART_CMD register can be used to clear the TBE bit.

During USART reception, the TX line is pulled low for a baud clock after finishing receiving the frame if a parity error is detected. This signal is the ‘NACK’ signal to smartcard. Then a frame error occurred 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 signals the error as a parity error if the received character is still erroneous after the maximum number of retries specified in the SCARNUM bit field.

The ‘NACK’ signal will be sent to the USART if the NKEN bit in USART_CTL2 is set. And the USART will not take the ‘NACK’ signal as the start bit.

The idle frame and break frame do not apply for the smartcard mode.

T=1 mode (block mode)

In T=1 (block) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the parity error transmission.

When requesting a read from the smartcard, the 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 received byte.

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 all the characters received. The length of the block, which must be programmed to the BL field in the USART_RT register, is communicated by 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.

17.3.13. 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.

17.3.14. Wakeup from deep-sleep mode

The USART is able to wake up the MCU from deep-sleep mode by the standard RBNE interrupt or the WUM interrupt.

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 soure 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.

17.3.15. USART interrupts

The USART interrupt events and flags are listed in the table below.

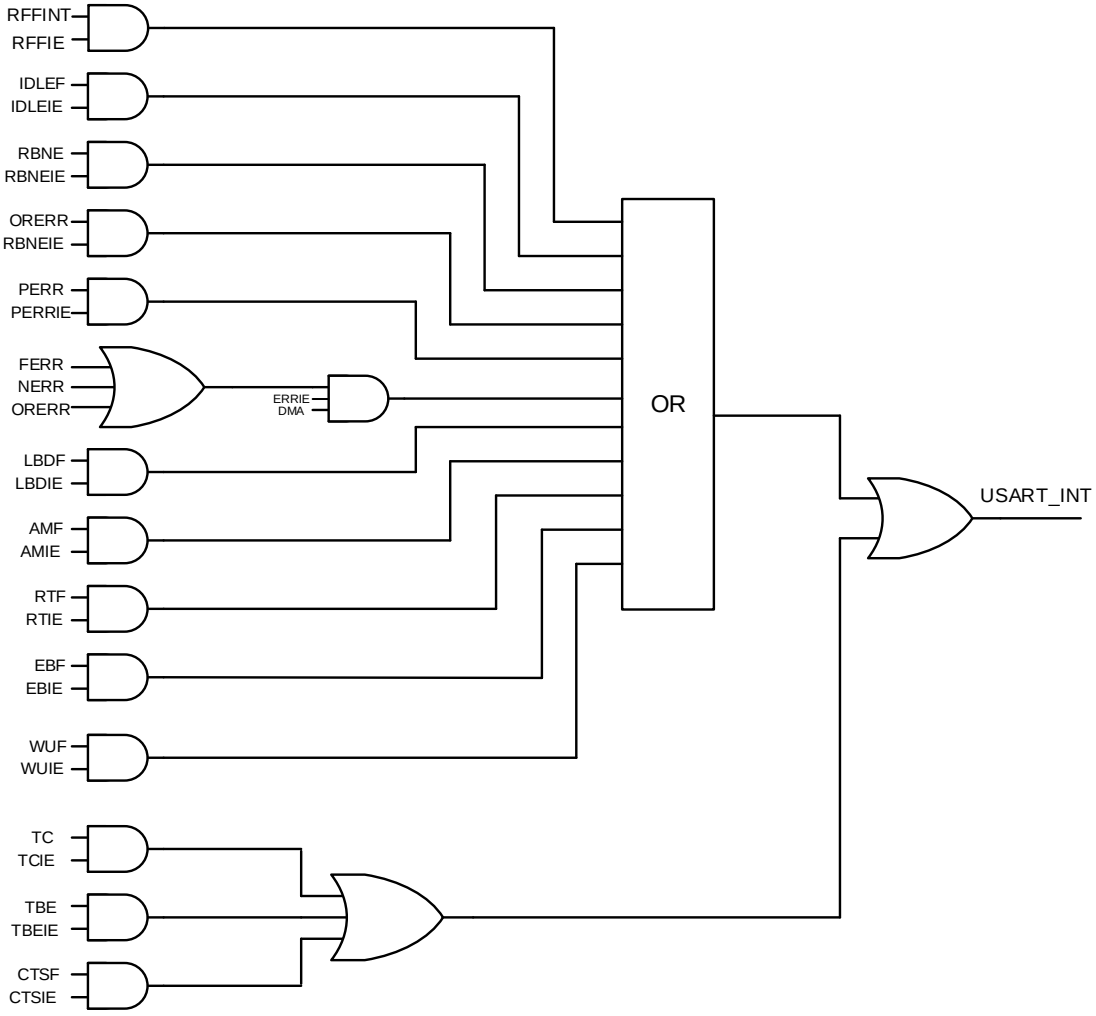
Table 17-3. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty	TBE	TBEIE
CTS flag	CTSF	CTSIE
Transmission complete	TC	TCIE
Received data ready to be read	RBNE	RBNEIE
Overrun error detected	ORERR	
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) in DMA reception	NERR or ORERR or FERR	ERRIE
Character match	AMF	AMIE
Receiver timeout error	RTF	RTIE

Interrupt event	Event flag	Enable Control bit
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 17-16. USART interrupt mapping diagram



17.4. Register definition

USART0 base address: 0x4001 3800
USART1 base address: 0x4000 4400

17.4.1. 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				EBIE	RTIE	DEA[4:0]					DED[4:0]				
				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.
27	EBIE	End of Block interrupt enable. 0: End of Block interrupt is disabled. 1: End of Block interrupt is enabled. This bit is reserved in USART1.
26	RTIE	Receiver timeout interrupt enable. 0: Receiver timeout interrupt is disabled. 1: Receiver timeout interrupt is enabled. This bit is reserved in USART1.
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	WM	Wakeup method in mute mode. 0: Idle Line 1: Address Mark 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. 0: Transmission complete interrupt is disabled. 1: An interrupt will occur whenever the TC bit is set in USART_STAT.
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. This bit is reserved in USART1.
0	UEN	USART enable 0: USART prescaler and outputs disabled. 1: USART prescaler and outputs enabled.

17.4.2. 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[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[7:0]	Address of the USART terminal. These bits give the address of the USART terminal. In multiprocessor communication during mute mode or deep-sleep mode, this is used for wakeup with address match detection. The received frame, the MSB of which is equal to 1, will be compared to these bits. When 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. This bit is reserved in USART1.
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). This bit is reserved in USART1.
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: 0.5 Stop bits 10: 2 Stop bits 11: 1.5 Stop bits 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).

		This bit is reserved in USART1.
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. This bit is reserved in USART1.
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). This bit is reserved in USART1.
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.

17.4.3. 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. This bit is reserved in USART1.
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 match, which is defined by ADDR and ADDM. 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). This bit is reserved in USART1.
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. This bit is reserved in USART1.
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 enabled, 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). This bit is reserved in USART1.
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). This bit is reserved in USART1.
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). This bit is reserved in USART1.
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.

17.4.4. 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).

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
BRR [15:4]												BRR[3:0]			
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.

17.4.5. 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).
This register is reserved in USART1.

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
GUAT[7:0]								PSC[7:0]							
rw								rw							

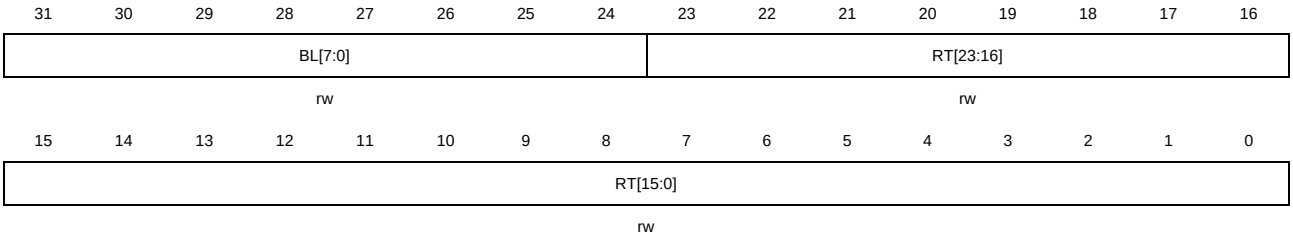
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	GUAT[7:0]	Guard time value in smartcard mode. This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock. In IrDA Low-power mode, the division factor is the prescaler 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).

17.4.6. Receiver timeout register (USART_RT)

Address offset: 0x14
Reset value: 0x0000_0000

This register has to be accessed by word (32-bit).
This bit is reserved in USART1.



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.

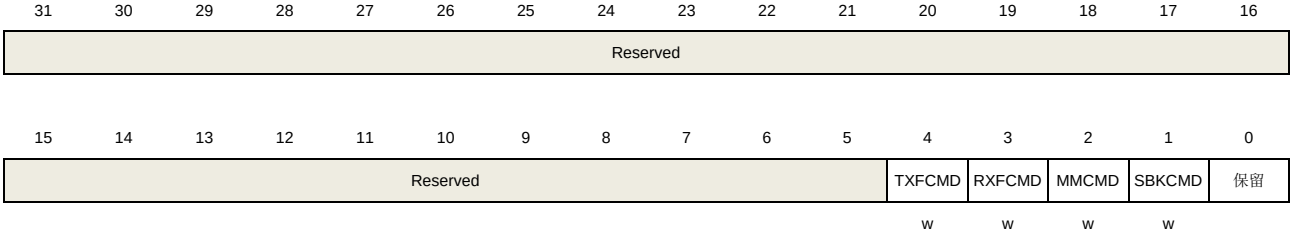
Note： When oversampling by 16, RT[23:0] cannot be configured as 0xFFFFFF.

17.4.7. Command register (USART_CMD)

Address offset: 0x18

Reset value: 0x0000_0000

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



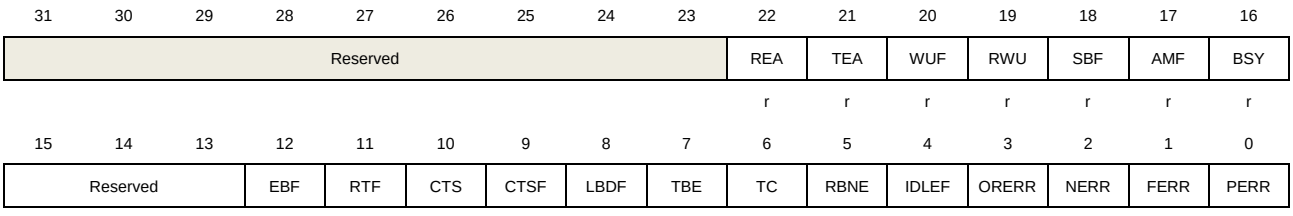
Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TXFCMD	Transmit data flush request Writing 1 to this bit sets the TBE flag, to discard the transmit data. This bit is reserved in USART1.
3	RXFCMD	Receive data flush command Writing 1 to this bit clears the RBNE flag to discard the 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.

17.4.8. Status register (USART_STAT)

Address offset: 0x1C

Reset value: 0x0000_00C0

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



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. This bit is reserved in USART1.
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. This bit is reserved in USART1.
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. This bit is reserved in USART1
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. This bit is reserved in USART1.

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.

17.4.9. 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. This bit is reserved in USART1.
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.

		This bit is reserved in USART1.
11	RTC	Receiver timeout clear. Writing 1 to this bit clears the RTF flag in the USART_STAT register. This bit is reserved in USART1.
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. This bit is reserved in USART1.
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.

17.4.10. Receive data register (USART_RDATA)

Offset: 0x24

Reset value: 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															

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

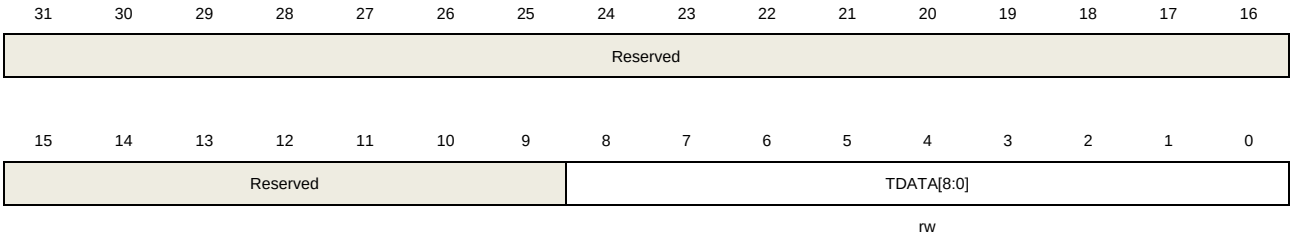
Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8:0	RDATA[8:0]	Receive data value The received data character is contained in these bits. The value read in the MSB (bit 7 or bit 8 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).

17.4.11. Transmit data register (USART_TDATA)

Offset: 0x28

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	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.

18. Inter-integrated circuit interface (I2C)

18.1. 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 and fast-mode 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.

18.2. 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.
- Multi-master capability.
- Supports standard-mode (up to 100 kHz) and fast-mode (up to 400 kHz).
- Configurable SCL stretching in slave mode.
- Supports DMA mode.
- SMBus 2.0 and PMBus compatible.
- 2 Interrupts: one for successful byte transmission and the other for error event.
- Optional PEC (Packet Error Checking) generation and check.

18.3. Function overview

Figure 18-1. I2C module block diagram below provides details of the internal configuration of the I2C interface.

Figure 18-1. I2C module block diagram

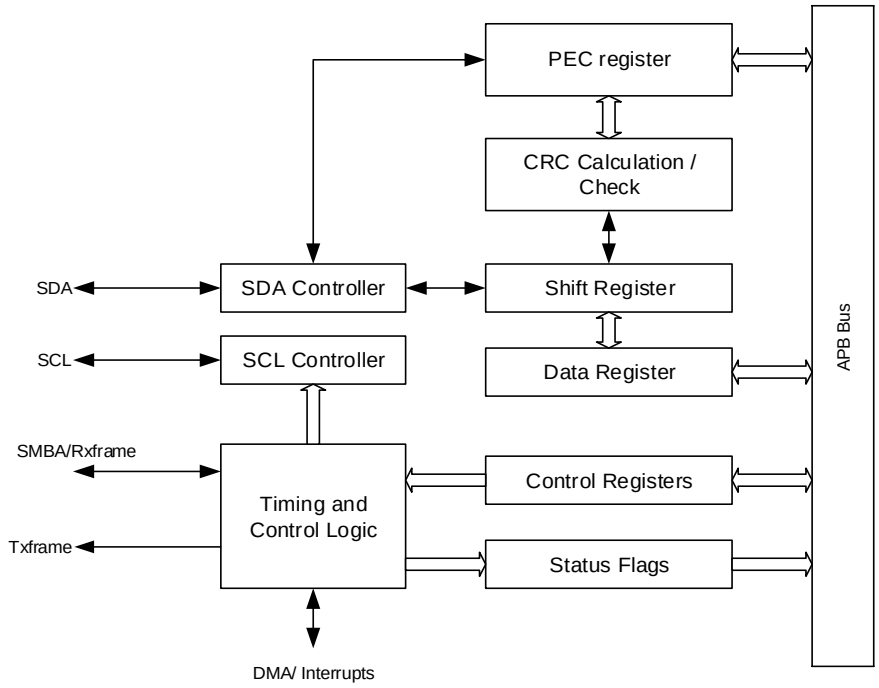


Table 18-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

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
Synchronization	Procedure to synchronize the clock signals of two or more devices
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

18.3.1. SDA and SCL lines

The I2C module has two external lines, the serial data SDA and serial clock SCL lines. The two wires carry information between the devices connected to the bus.

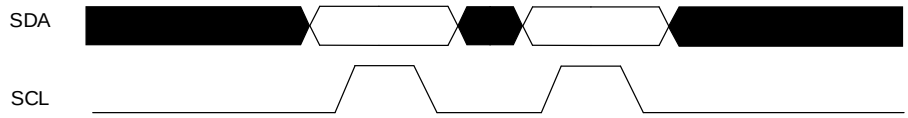
Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via current-source or pull-up resistor. When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collect to perform the wired-AND function. Data on the I2C-bus can be transferred at rates of up to 100 Kbit/s in the standard mode and up to 400 Kbit/s in the fast mode. Due to the variety of different technology

devices (CMOS, NMOS, bipolar) that can be connected to the I2C-bus, the voltage levels of the logical '0' (LOW) and '1' (HIGH) are not fixed and depend on the associated level of V_{DD}.

18.3.2. 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 SDA line can only change when the clock signal on the SCL line is LOW (see ***Figure 18-2. Data validation***). One clock pulse is generated for each data bit to be transferred.

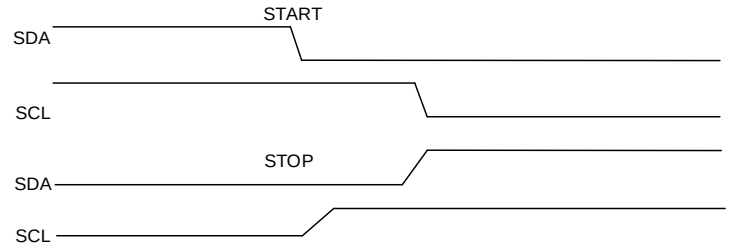
Figure 18-2. Data validation



18.3.3. START and STOP signal

All transmissions begin with a START and are terminated by a STOP (see ***Figure 18-3. 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 18-3. START and STOP signal

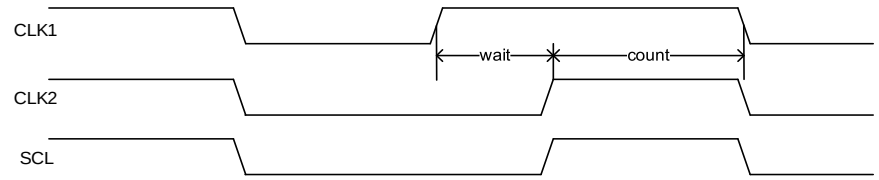


18.3.4. Clock synchronization

Two masters can begin transmitting on a free bus at the same time and there must be a method for deciding which master takes control of the bus and completes its transmission. This is done by clock synchronization and bus arbitration. In a single master system, clock synchronization and bus arbitration are unnecessary.

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line causes the masters concerned to start counting their LOW period, and once a master clock has gone LOW, it holds the SCL line in that state until the clock HIGH state is reached (see ***Figure 18-4. Clock synchronization***). However, if another clock is still within its LOW period, the LOW to HIGH transition of this clock may not change the state of the SCL line. The SCL line is therefore held LOW by the master with the longest LOW period. Masters with shorter LOW period enter a HIGH wait-state during this time.

Figure 18-4. Clock synchronization



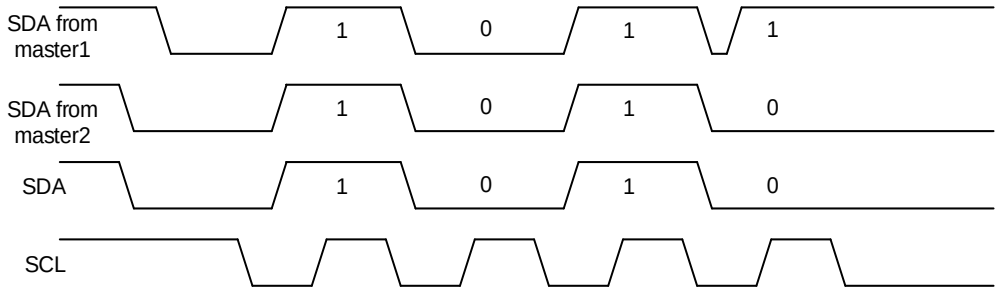
18.3.5. Arbitration

Arbitration, like synchronization, is part of the protocol where more than one master is used in the system. Slaves are not involved in the arbitration procedure.

A master may start a transfer only if the bus is free. Two masters may generate a START signal within the minimum hold time of the START signal which results in a valid START signal on the bus. Arbitration is then required to determine which master will complete its transmission.

Arbitration proceeds bit by bit. During every bit, while SCL is HIGH, each master checks whether the SDA level matches what it has been sent. This process may take many bits. Two masters can even complete an entire transmission without error, as long as the transmissions are identical. The first time a master tries to send a HIGH, but detects that the SDA level is LOW, then the master knows that it has lost the arbitration and turns off its SDA output driver. The other master goes on to complete its transmission.

Figure 18-5. SDA line arbitration



18.3.6. I2C communication flow

Each I2C device is recognized by a unique address (whether it is a microcontroller, LCD driver, memory or keyboard interface) and can be operated as either a transmitter or receiver, depending on the function of the device.

An I2C slave will continue to detect addresses after a START signal on I2C bus and compare the detected address with its slave address which is programmed by software. Once the two addresses match with each other, the I2C slave will send an ACK to the I2C bus and respond 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 responds to a General Call Address (0x00). The I2C block supports both 7-bit and 10-bit address modes.

An I2C master always initiates or ends a transfer using START or STOP signal and it's also

responsible for SCL clock generation.

Figure 18-6. I2C communication flow with 7-bit address

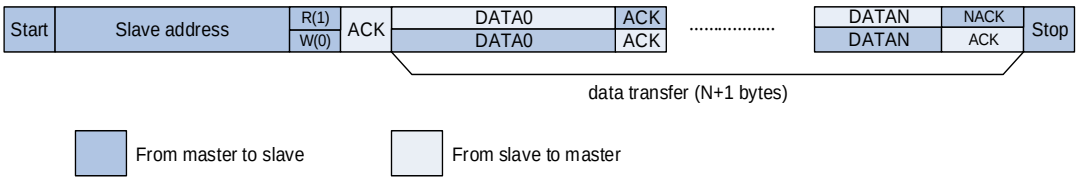


Figure 18-7. I2C communication flow with 10-bit address (Master Transmit)

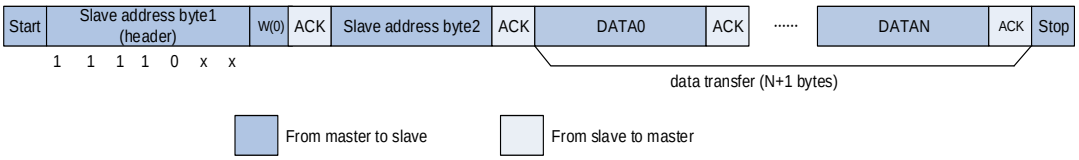
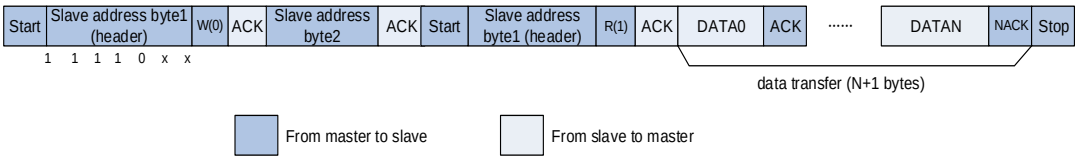


Figure 18-8. I2C communication flow with 10-bit address (Master Receive)



18.3.7. Programming model

An I2C device such as LCD driver may only be a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers. A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered as a slave.

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:

- Master Transmitter
- Master Receiver
- Slave Transmitter
- Slave Receiver

I2C block supports all of the four I2C modes. After system reset, it works in slave mode. After sending a START signal on I2C bus, it changes into master mode. The I2C changes back to slave mode after sending a STOP signal on I2C bus.

Programming model in slave transmitting mode

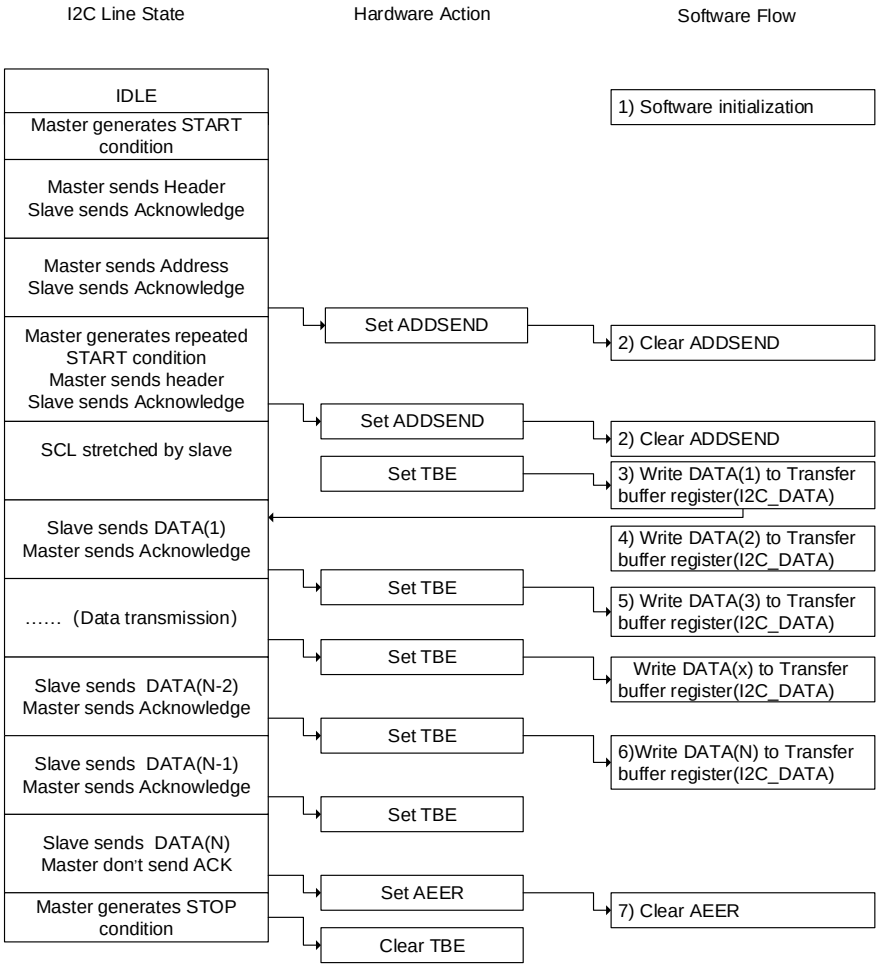
As is shown in ***Figure 18-9. Programming model for slave transmitting (10-bit address mode)***, the following software procedure should be followed if users wish to transmit data in slave transmitter mode:

1. First of all, enable I2C peripheral clock as well as configure clock related registers in

I2C_CTL1 to make sure correct I2C timing. After enabled and configured, I2C operates in its default slave state and waits for START signal followed by address on I2C bus.

2. After receiving a START signal followed by a matched address, either in 7-bit format or in 10-bit format, the I2C hardware sets the ADDSEND bit in I2C_STAT0 register, which should be monitored by software either by polling or interrupt. After that, software should read I2C_STAT0 and then I2C_STAT1 to clear ADDSEND bit. If 10-bit addressing format is selected, the I2C master should then send a repeated START signal followed by a header to the I2C bus. The slave sets ADDSEND bit again after it detects the repeated START signal and the following header. The ADDSEND bit must be cleared by software again by reading I2C_STAT0 and then I2C_STAT1.
3. Now I2C enters data transmission stage and hardware sets TBE bit because both the shift register and data register I2C_DATA are empty. Once TBE is set, software should write the first byte of data to I2C_DATA register, TBE is not cleared in this case because the byte written in I2C_DATA is moved to the internal shift register immediately. I2C begins to transmit data to I2C bus as soon as the shift register is not empty.
4. During the transmission of the first byte, software can write the second byte to I2C_DATA, and this time TBE is cleared because neither I2C_DATA nor shift register is empty.
5. After the transmission of the first byte, the TBE bit will be set, the software can write the third byte to the I2C_DATA register and TBE is cleared. After this, any time TBE is set, software can write a byte to I2C_DATA as long as there is still data to be transmitted.
6. During the transmission of the second last byte, software writes the last data to I2C_DATA to clear the TBE flag and doesn't care TBE anymore. So TBE will be set after the byte's transmission and not cleared until a STOP signal.
7. I2C master doesn't acknowledge to the last byte according to the I2C protocol, so after sending the last byte, I2C slave will wait for the STOP signal on I2C bus and sets AERR (Acknowledge Error) bit to notify software that the transmission completes. Software clears AERR bit by writing 0 to it.

Figure 18-9. Programming model for slave transmitting (10-bit address mode)



Programming model in slave receiving mode

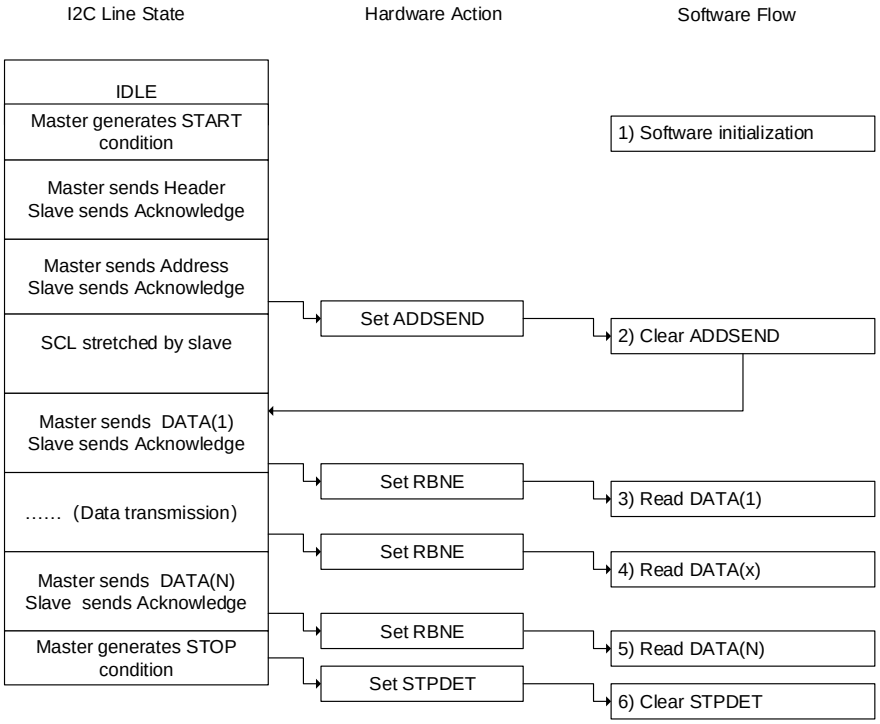
As is shown in ***Figure 18-10. Programming model for slave receiving (10-bit address mode)***, the following software procedure should be followed if users wish to receive data in slave receiver mode:

1. First of all, enable I2C peripheral clock as well as configure clock related registers in I2C_CTL1 to make sure correct I2C timing. After enabled and configured, I2C operates in its default slave state and waits for START signal followed by address on I2C bus.
2. After receiving a START signal followed by a matched 7-bit or 10-bit address, the I2C hardware sets the ADDSEND bit in I2C status register 0, which should be monitored by software either by polling or interrupt. After that software should read I2C_STAT0 and then I2C_STAT1 to clear ADDSEND bit. The I2C begins to receive data on I2C bus as soon as ADDSEND bit is cleared.
3. As soon as the first byte is received, RBNE is set by hardware. Software can now read the first byte from I2C_DATA and RBNE is cleared as well.
4. Any time RBNE is set, software can read a byte from I2C_DATA.

5. After the last byte is received, RBNE is set. Software reads the last byte.

6. STPDET bit is set when I2C detects a STOP signal on I2C bus and software reads I2C_STAT0 and then writes I2C_CTL0 to clear the STPDET bit.

Figure 18-10. Programming model for slave receiving (10-bit address mode)



Programming model in master transmitting mode

As is shown in ***Figure 18-11. Programming model for master transmitting (10-bit address mode)***, the following software procedure should be followed if users wish to make transaction in master transmitter mode:

1. First of all, enable I2C peripheral clock as well as configure clock related registers in I2C_CTL1 to make sure correct I2C timing. After enabled and configured, I2C operates in its default slave state and waits for START signal followed by address on I2C bus.

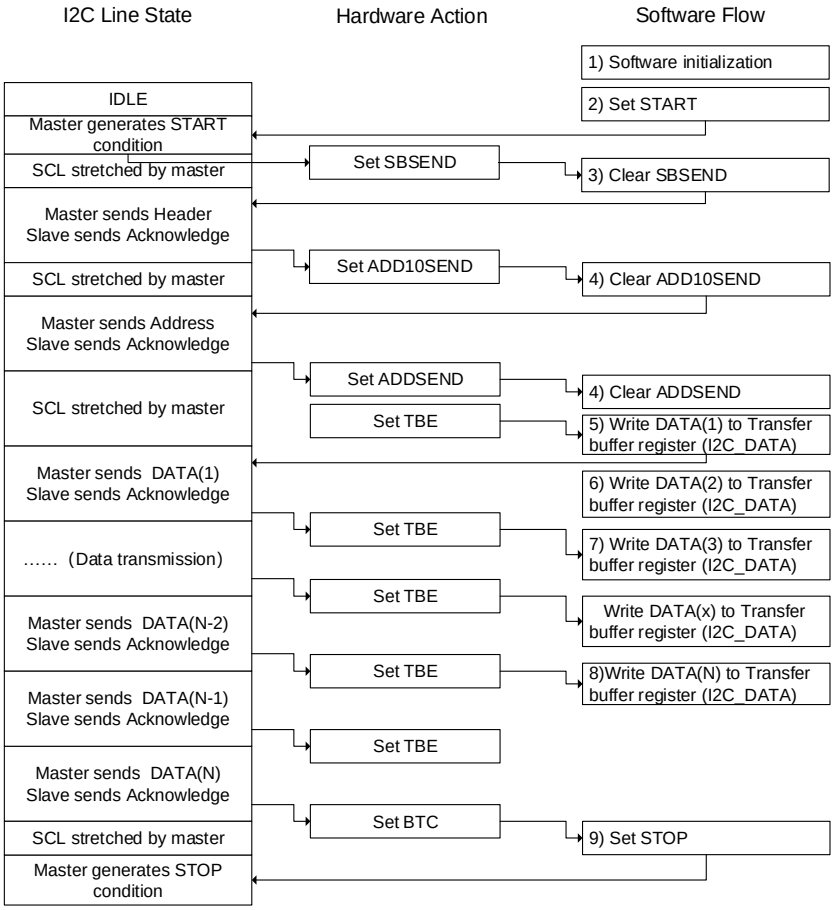
2. Software sets START bit requesting I2C to generate a START signal on I2C bus.

3. After sending a START signal, the I2C hardware sets the SBSEND bit in I2C_STAT0 register and enters master mode. Now software should clear the SBSEND bit by reading I2C_STAT0 and then writing a 7-bit address or header of a 10-bit address to I2C_DATA. I2C begins to send address or header to I2C bus as soon as SBSEND bit is cleared. If the address which has been sent is header of a 10-bit address, the hardware sets ADD10SEND bit after sending the header and software should clear the ADD10SEND bit by reading I2C_STAT0 and writing 10-bit lower address to I2C_DATA.

4. After the 7-bit or 10-bit address has been sent, the I2C hardware sets the ADDSEND bit and software should clear the ADDSEND bit by reading I2C_STAT0 and then I2C_STAT1.

5. Now I2C enters data transmission stage and hardware sets TBE bit because both the shift register and data register I2C_DATA are empty. Software now writes the first byte data to I2C_DATA register, but the TBE will not be cleared because the byte written in I2C_DATA is moved to internal shift register immediately. The I2C begins to transmit data to I2C bus as soon as the shift register is not empty.
6. During the transmission of the first byte, software can write the second byte to I2C_DATA, and this time TBE is cleared because neither I2C_DATA nor shift register is empty.
7. Any time TBE is set, software can write a byte to I2C_DATA as long as there is still data to be transmitted.
8. During the transmission of the second last byte, software writes the last data to I2C_DATA to clear the TBE flag and doesn't care TBE anymore. So TBE will be asserted after the transmission of the byte and not be cleared until a STOP signal.
9. After sending the last byte, I2C master sets BTC bit because both the shift register and I2C_DATA are empty. Software should set the STOP bit to generate a STOP signal, then the I2C clears both TBE and BTC flags.

Figure 18-11. Programming model for master transmitting (10-bit address mode)



Programming model in master receiving mode

In master receiving mode, a master is responsible for generating NACK for the last byte reception and then sending a STOP signal on I2C bus. So, special attention should be paid to ensure the correct ending of data reception. Two solutions for master receiving are provided here for applications: Solution A and B. Solution A requires the software's quick response to I2C events, while Solution B doesn't.

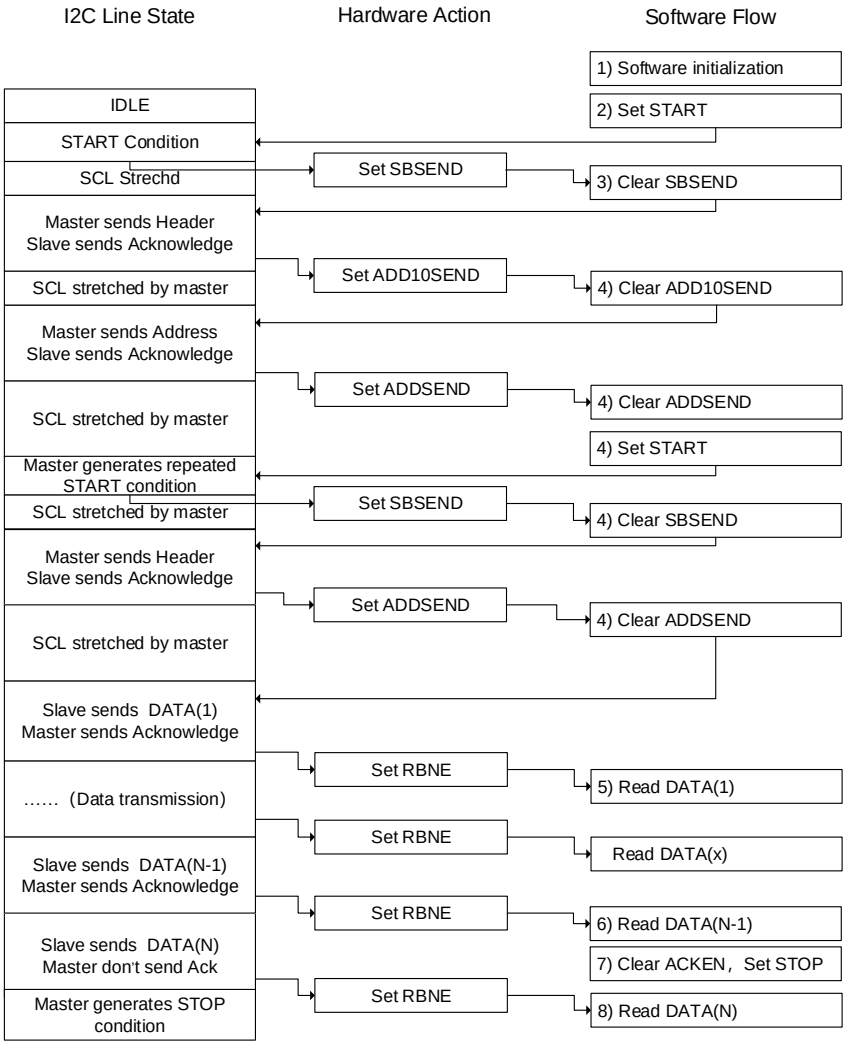
Solution A

1. First of all, enable I2C peripheral clock as well as configure clock related registers in I2C_CTL1 to make sure correct I2C timing. After enabled and configured, I2C operates in its default slave state and waits for START signal followed by address on I2C bus.
2. Software sets START bit requesting I2C to generate a START signal on I2C bus.
3. After sending a START signal, the I2C hardware sets the SBSEND bit in I2C_STAT0 register and enters master mode. Now software should clear the SBSEND bit by reading I2C_STAT0 and then writing a 7-bit address or header of a 10-bit address to I2C_DATA. I2C begins to send address or header to I2C bus as soon as SBSEND bit is cleared. If the address which has been sent is header of a 10-bit address, the hardware sets ADD10SEND bit after sending header and software should clear the ADD10SEND bit by reading I2C_STAT0 and writing 10-bit lower address to I2C_DATA.

4. After the 7-bit or 10-bit address has been sent, the I2C hardware sets the ADDSEND bit and software should clear the ADDSEND bit by reading I2C_STAT0 and then I2C_STAT1. If the address is in 10-bit format, software should then set START bit again to generate a repeated START signal on I2C bus and SBSEND is set after the repeated START is sent out. Software should clear the SBSEND bit by reading I2C_STAT0 and writing header to I2C_DATA. Then the header is sent out to I2C bus, and ADDSEND is set again. Software should again clear ADDSEND by reading I2C_STAT0 and then I2C_STAT1.
5. As soon as the first byte is received, RBNE is set by hardware. Software now can read the first byte from I2C_DATA and RBNE is cleared as well.
6. Any time RBNE is set, software can read a byte from I2C_DATA.
7. After the second last byte (N-1) is received, the software should clear ACKEN bit and set STOP bit. These actions should complete before the end of the last byte's receiving to ensure that NACK will be sent for the last byte.
8. After the last byte is received, RBNE is set. Software reads the last byte. Since ACKEN has been cleared in the previous step, I2C doesn't send ACK for the last byte and it generates a STOP signal after the transmission of the last byte.

The above steps require byte number N>1. If N=1, Step 7 should be performed after Step 4 and completed before the end of the single byte's receiving.

Figure 18-12. Programming model for master receiving using Solution A (10-bit address mode)



Solution B

1. First of all, enable I2C peripheral clock as well as configure clock related registers in I2C_CTL1 to make sure correct I2C timing. After enabled and configured, I2C operates in its default slave state and waits for START signal followed by address on I2C bus.
2. Software sets START bit requesting I2C to generate a START signal on I2C bus.
3. After sending a START signal, the I2C hardware sets the SBSEND bit in I2C_STAT0 register and enters master mode. Now software should clear the SBSEND bit by reading I2C_STAT0 and then writing a 7-bit address or header of a 10-bit address to I2C_DATA. I2C begins to send address or header to I2C bus as soon as SBSEND bit is cleared. If the address which has been sent is a header of 10-bit address, the hardware sets ADD10SEND bit after sending header and software should clear the ADD10SEND bit by reading I2C_STAT0 and writing 10-bit lower address to I2C_DATA.
4. After the 7-bit or 10-bit address has been sent, the I2C hardware sets the ADDSEND bit and software should clear the ADDSEND bit by reading I2C_STAT0 and then I2C_STAT1. If the address is in 10-bit format, software should then set START bit again to generate

a repeated START signal on I2C bus and SBSEND is set after the repeated START is sent out. Software should clear the SBSEND bit by reading I2C_STAT0 and writing header to I2C_DATA. Then the header is sent out to I2C bus, and ADDSEND is set again. Software should again clear ADDSEND by reading I2C_STAT0 and then I2C_STAT1.

5. As soon as the first byte is received, RBNE is set by hardware. Software now can read the first byte from I2C_DATA and RBNE is cleared as well.

6. Any time RBNE is set, software can read a byte from I2C_DATA until the master receives N-3 bytes.

As shown in ***Figure 18-13. Programming model for master receiving mode using solution B (10-bit address mode)***, the N-2 byte is not read out by software, so after the N-1 byte is received, both BTC and RBNE are asserted. The bus is stretched by master to prevent the reception of the last byte. Then software should clear ACKEN bit.

7. Software reads out N-2 byte, clearing BTC. After this, the N-1 byte is moved from shift register to I2C_DATA and bus is released and begins to receive the last byte. Master doesn't send an ACK for the last byte because ACKEN is already cleared.

8. After the last byte is received, both BTC and RBNE are set again, and SCL is stretched low. Software sets STOP bit and master sends out a STOP signal on bus.

9. Software reads the N-1 byte, clearing BTC. After this the last byte is moved from shift register to I2C_DATA.

10. Software reads the last byte, clearing RBNE.

The above steps require that byte number N>2. N=1 and N=2 are similar:

N=1

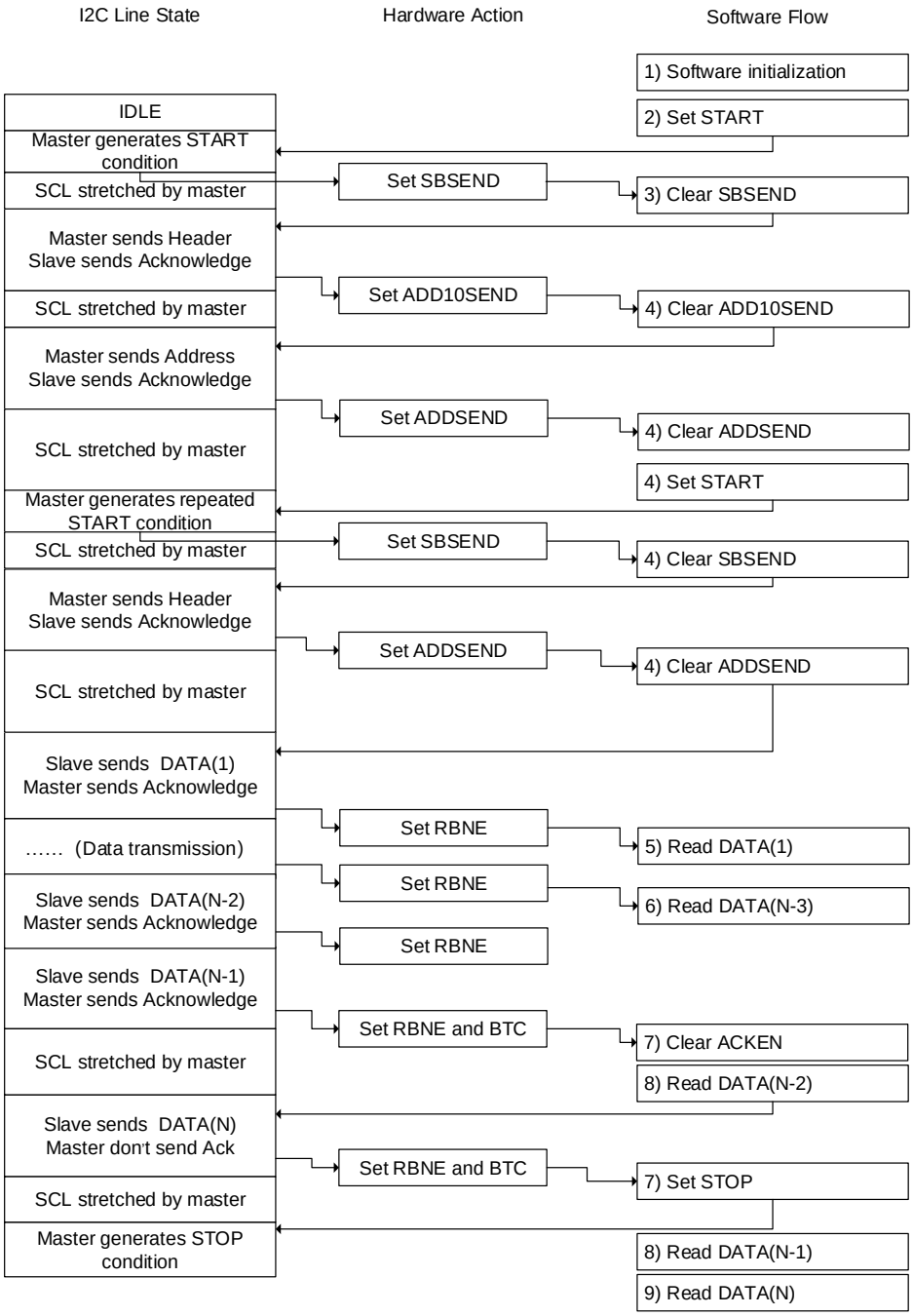
In Step4, software should reset ACKEN bit before clearing ADDSEND bit and set STOP bit after clearing ADDSEND bit. Step 5 is the last step when N=1.

N=2

In Step 2, software should set POAP bit before setting START bit. In Step 4, software should reset ACKEN bit before clearing ADDSEND bit. In Step 5, software should wait until BTC is set and then set STOP bit and read I2C_DATA twice.

Figure 18-13. Programming model for master receiving mode using solution B (10-bit

address mode)



18.3.8. SCL line stretching

The SCL line stretching function is designed to avoid overflow error in reception and underflow error in transmission. As is shown in Programming Model, when the TBE and BTC bits are set in transmitting mode, the transmitter stretches the SCL line low until the transfer buffer register is filled with the next data to be transmitted. When the RBNE and BTC bits are set in receiving mode, the receiver stretches the SCL line low until the data in the transfer buffer is read out.

When works in slave mode, the SCL line stretching function can be disabled by setting the

SS bit in the I2C_CTL0 register. If this bit is set, the software is required to be quick enough to serve the TBE, RBNE and BTC status, otherwise, overflow or underflow situation might occur.

18.3.9. Use DMA for data transfer

As is shown in Programming Model, each time TBE or RBNE is asserted, software should write or read a byte, this may cause CPU to be high overloaded. The DMA controller can be used to process TBE and RBNE flags: each time TBE or RBNE is asserted, DMA controller does a read or write operation automatically. It reduces the load on the CPU. See the DMA section for details on how to configure DMA.

The DMA request is enabled by the DMAON bit in the I2C_CTL1 register. This bit should be set after clearing the ADDSEND status. If the SCL line stretching function is disabled for a slave device, the DMAON bit should be set before the ADDSEND event.

Refer to the specification of the DMA controller for the configuration method of a DMA stream. The DMA controller must be configured and enabled before the I2C transfer. When the configured number of bytes have been transferred, the DMA controller generates End of Transfer (EOT) interrupt. DMA will send an End of Transmission (EOT) signal to the I2C interface and generates a DMA full transfer finish interrupt.

When a master receives two or more bytes, the DMALST bit in the I2C_CTL1 register should be set. The I2C master will send NACK after the last byte. The STOP bit can be set by software to generate a STOP signal in the ISR of the DMA full transfer finish interrupt.

When a master receives only one byte, the ACKEN bit must be cleared before clearing the ADDSEND status. Software can set the STOP bit to generate a STOP signal after clearing the ADDSEND status, or in the ISR of the DMA full transfer finish interrupt.

18.3.10. Packet error checking

There is a CRC-8 calculator in I2C block to perform PEC (Packet Error Checking) for I2C data. The polynomial of the CRC is x8 + x2 + x + 1 which is compatible with the SMBus protocol. If enabled by setting PECEN bit, the PEC will calculate all the data transmitted through I2C including address. I2C is able to send out the PEC value after the last data byte or check the received PEC value with its calculated PEC using the PECTRANS bit. In DMA mode, the I2C will send or check PEC value automatically if PECEN bit and PECTRANS bit are set.

18.3.11. 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 transmission 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 is realized based on I2C hardware and it uses I2C hardware addressing, but it adds the 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 allows 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.

Time-out feature

SMBus has a time-out feature which resets devices if a communication takes too long. This explains the minimum clock frequency is 10 kHz to prevent locking up the bus. I2C can be a ‘DC’ bus, which means that a slave device stretches the master clock when performing some routines while the master is accessing it. This will notify the master that the slave is busy but does not want to lose the communication. The slave device will continue the communication after its task is completed. There is no limit in the I2C bus protocol of how long this delay can be, whereas for a SMBus system, it would be limited to 25ms. 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 solve the problem. Slave devices are not allowed to hold the clock low too long.

Packet error checking

SMBus 2.0 and 1.1 allow Packet Error Checking (PEC). In that mode, a PEC byte is appended at the end of each transaction. The byte is a CRC-8 checksum of the entire message including the address and read/write bit. The polynomial used is x8+x2+x+1 (the CRC-8-ATM HEC algorithm, initialized to zero).

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. SMBus also defines a less common "Host Notify Protocol", providing similar notifications which is based on the I2C multi-master mode but it can pass more data.

SMBus programming flow

The programming flow for SMBus is similar to normal I2C. In order to use SMBus mode, the application should configure several SMBus specific registers, respond to some SMBus specific flags and implement the upper protocols described in SMBus specification.

1. Before communication, SMBEN bit in I2C_CTL0 should be set and SMBSEL and ARPEN bits should be configured to desired values.
2. In order to support address resolution protocol (ARP) (ARPEN=1), the software should respond to HSTSMB flag in SMBus Host Mode (SMBSEL =1) or DEFSMB flag in SMBus Device Mode, and implement the function of ARP protocol.
3. In order to support SMBus Alert Mode, the software should respond to SMBALT flag and implement the related function.

18.3.12. Status, errors and interrupts

There are several status and error flags in I2C, and interrupts may be asserted from these flags by setting some register bits (refer to ***Register definition*** for detail).

Table 18-2. Event status flags

Event Flag Name	Description
SBSEND	START signal sent (master)
ADDSEND	Address sent or received
ADD10SEND	Header of 10-bit address sent
STPDET	STOP signal detected
BTC	Byte transmission completed
TBE	I2C_DATA is empty when transmitting
RBNE	I2C_DATA is not empty when receiving

Table 18-3. Error flags

Error Name	Description
BERR	Bus error
LOSTARB	Arbitration lost
OUERR	Over-run or under-run when SCL stretch is disabled.
AERR	No acknowledge received
PECERR	CRC value doesn’t match
SMBTO	Bus timeout in SMBus mode
SMBALT	SMBus Alert

18.4. Register definition

I2C0 base address: 0x4000 5400
I2C1 base address: 0x4000 5800

18.4.1. Control register 0 (I2C_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
SRESET	Reserved	SALT	PECTRANS	POAP	ACKEN	STOP	START	SS	GCEN	PECEN	ARPEN	SMBSEL	Reserved	SMBEN	I2CEN
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	SRESET	Software resets I2C, software should wait until the I2C lines are released to reset the I2C. 0: I2C is not under reset 1: I2C is under reset
14	Reserved	Must be kept at reset value.
13	SALT	SMBus Alert. Issue alert through SMBA pin. Software can set and clear this bit and hardware can clear this bit. 0: Don’t issue alert through SMBA pin 1: Issue alert through SMBA pin
12	PECTRANS	PEC transfer Software sets and clears this bit while hardware clears this bit when PEC is transferred or START / STOP signal is detected or I2CEN=0. 0: Don’t transfer PEC value 1: Transfer PEC value
11	POAP	Position of ACK and PEC when receiving This bit is set and cleared by software and cleared by hardware when I2CEN=0. 0: ACKEN bit specifies whether to send ACK or NACK for the current byte that is being received. PECTRANS bit indicates that the current receiving byte is a PEC byte. 1: ACKEN bit specifies whether to send ACK or NACK for the next byte that is to be

		received, PECTRANS bit indicates the next byte that is to be received is a PEC byte.
10	ACKEN	Whether or not to send an ACK This bit is set and cleared by software and cleared by hardware when I2CEN=0. 0: ACK will not be sent 1: ACK will be sent
9	STOP	Generate a STOP signal on I2C bus This bit is set and cleared by software and set by hardware when SMBus timeout and cleared by hardware when STOP signal is detected. 0: STOP will not be sent 1: STOP will be sent
8	START	Generate a START signal on I2C bus This bit is set and cleared by software and cleared by hardware when a START signal is detected or I2CEN=0. 0: START will not be sent 1: START will be sent
7	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
6	GCEN	Whether or not to response to a General Call (0x00) 0: Slave won't respond to a General Call 1: Slave will respond to a General Call
5	PECEN	PEC calculation enable 0: PEC calculation disable 1: PEC calculation enable
4	ARPEN	ARP protocol enable in SMBus mode 0: ARP is disabled 1: ARP is enabled
3	SMBSEL	SMBus type selection 0: Device 1: Host
2	Reserved	Must be kept at reset value.
1	SMBEN	SMBus/I2C mode switch 0: I2C mode 1: SMBus mode
0	I2CEN	I2C peripheral enable 0: I2C is disabled

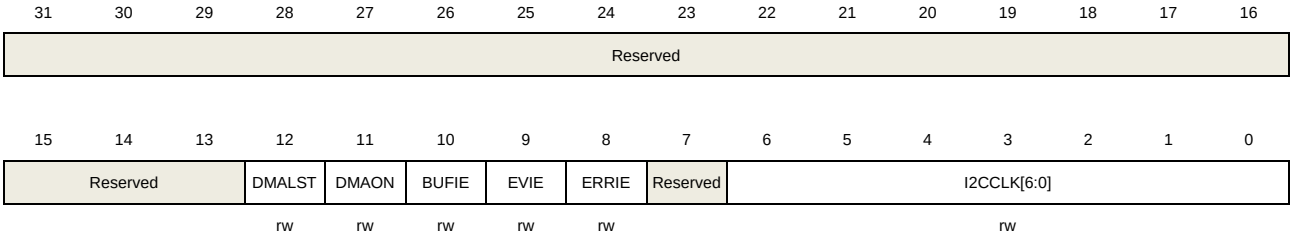
1: I2C is enabled

18.4.2. Control register 1 (I2C_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
31:13	Reserved	Must be kept at reset value.
12	DMALST	DMA last transfer configure 0: Next DMA EOT is not the last transfer 1: Next DMA EOT is the last transfer
11	DMAON	DMA mode switch 0: DMA mode switched off 1: DMA mode switched on
10	BUFIE	Buffer interrupt enable 0: Buffer interrupt is disabled, TBE = 1 or RBNE = 1 when EVIE=1 will not generate an interrupt. 1: Buffer interrupt is enabled, which means that interrupt will be generated when TBE = 1 or RBNE = 1 if EVIE=1.
9	EVIE	Event interrupt enable 0: Event interrupt is disabled 1: Event interrupt is enabled, which means that interrupt will be generated when SBSEND, ADDSEND, ADD10SEND, STPDET or BTC flag asserted or TBE=1 or RBNE=1 if BUFIE=1.
8	ERRIE	Error interrupt enable 0: Error interrupt is disabled 1: Error interrupt is enabled, which means that interrupt will be generated when BERR, LOSTARB, AERR, OUERR, PECERR, SMBTO or SMBALT flag is asserted.
7	Reserved	Must be kept at reset value.
6:0	I2CCLK[6:0]	I2C peripheral clock frequency I2CCLK[6:0] should be the frequency of input APB1 clock in MHz which is at least 2.

0000000 - 0000001: Not allowed

0000010 - 1001000: 2 MHz~72MHz

1001001 - 1111111: Not allowed due to the limitation of APB1 clock

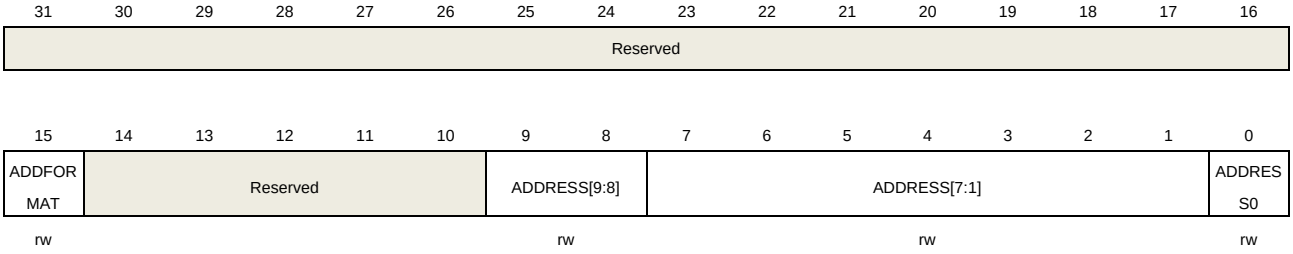
Note: In I2C standard mode, the frequencies of APB1 must be equal or greater than 2MHz. In I2C fast mode, the frequencies of APB1 must be equal or greater than 8MHz.

18.4.3. Slave address register 0 (I2C_SADDR0)

Address offset: 0x08

Reset value: 0x0000 0000

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



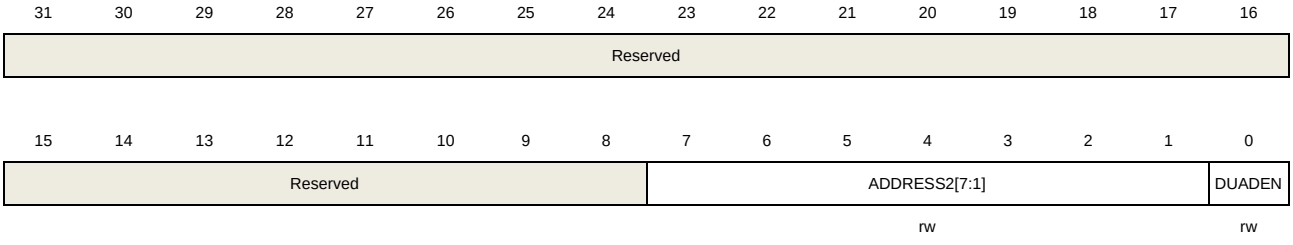
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDFORMAT	Address format for the I2C slave 0: 7-bit address 1: 10-bit address
14:10	Reserved	Must be kept at reset value.
9:8	ADDRESS[9:8]	Highest two bits of a 10-bit address
7:1	ADDRESS[7:1]	7-bit address or bits 7:1 of a 10-bit address
0	ADDRESS0	Bit 0 of a 10-bit address

18.4.4. Slave address register 1 (I2C_SADDR1)

Address offset: 0x0C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:1	ADDRESS2[7:1]	The second I2C address for the slave in Dual-Address mode
0	DUADEN	Dual-Address mode enable 0: Dual-Address mode is disabled 1: Dual-Address mode is enabled

18.4.5. Transfer buffer register (I2C_DATA)

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								TRB[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TRB[7:0]	Transmission or reception data buffer

18.4.6. Transfer status register 0 (I2C_STAT0)

Address offset: 0x14

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
SMBALT	SMBTO	Reserved	PECERR	OUERR	AERR	LOSTARB	BERR	TBE	RBNE	Reserved	STPDET	ADD10SEND	BTC	ADDSEND	SBSEND
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	SMBALT	SMBus Alert status This bit is set by hardware and cleared by writing 0.

		0: SMBA pin not pulled down (device mode) or no Alert detected (host mode) 1: SMBA pin pulled down and Alert address received (device mode) or Alert detected (host mode)
14	SMBTO	Timeout signal in SMBus mode This bit is set by hardware and cleared by writing 0. 0: No timeout error 1: Timeout event occurs (SCL is low for 25 ms)
13	Reserved	Must be kept at reset value.
12	PECERR	PEC error when receiving data This bit is set by hardware and cleared by writing 0. 0: Received PEC matches the calculated PEC 1: Received PEC doesn't match the calculated PEC, I2C will send NACK careless of ACKEN bit.
11	OUERR	Over-run or under-run situation occurs in slave mode, when SCL stretching is disabled. In slave receiving mode, if the last byte in I2C_DATA is not read out while the following byte is already received, over-run occurs. In slave transmitting mode, if the current byte is already sent out, while the I2C_DATA is still empty, under-run occurs. This bit is set by hardware and cleared by writing 0. 0: No over-run or under-run occurs. 1: Over-run or under-run occurs.
10	AERR	Acknowledge error This bit is set by hardware and cleared by writing 0. 0: No acknowledge error 1: Acknowledge error
9	LOSTARB	Arbitration lost in master mode This bit is set by hardware and cleared by writing 0. 0: No arbitration lost 1: Arbitration lost occurs and the I2C block changes back to slave mode.
8	BERR	Bus error A bus error occurs when an unexpected START or STOP signal on I2C bus. This bit is set by hardware and cleared by writing 0. 0: No bus error 1: A bus error detected
7	TBE	I2C_DATA is empty during transmitting This bit is set by hardware after it moves a byte from I2C_DATA to shift register and cleared by writing a byte to I2C_DATA. If both the shift register and I2C_DATA are empty, writing I2C_DATA won't clear TBE (refer to Programming Model for detail). 0: I2C_DATA is not empty 1: I2C_DATA is empty, software can write

6	RBNE	I2C_DATA is not empty during receiving This bit is set by hardware after it moves a byte from shift register to I2C_DATA and cleared by reading I2C_DATA. If both BTC and RBNE are asserted, reading I2C_DATA won't clear RBNE because the byte in shift register will be moved to I2C_DATA immediately. 0: I2C_DATA is empty 1: I2C_DATA is not empty, software can read
5	Reserved	Must be kept at reset value.
4	STPDET	STOP signal is detected in slave mode This bit is set by hardware and cleared by reading I2C_STAT0 and then writing I2C_CTL0. 0: STOP signal not detected in slave mode 1: STOP signal detected in slave mode
3	ADD10SEND	Header of 10-bit address is sent in master mode This bit is set by hardware and cleared by reading I2C_STAT0 and writing I2C_DATA. 0: No header of 10-bit address is sent in master mode 1: Header of 10-bit address is sent in master mode
2	BTC	Byte transmission is completed. If a byte is already received in shift register but I2C_DATA is still full in receiving mode or a byte is already sent out from shift register but I2C_DATA is still empty in transmitting mode, the BTC flag is asserted if SCL stretching enabled. This bit is set by hardware and cleared by 3 ways as follow: 1. Software clearing: reading I2C_STAT0 followed by reading or writing I2C_DATA 2. Hardware clearing: sending the STOP signal or START signal 3. Bit 0 (I2CEN bit) of the I2C_CTL0 is reset. 0: BTC not asserted 1: BTC asserted
1	ADDSEND	Address is sent and ACK is received in master mode or address is received and matches with its own address in slave mode. This bit is set by hardware and cleared by reading I2C_STAT0 and reading I2C_STAT1. 0: In slave mode, no address is received or the received address does not match witih its own address. In master mode, no address is sent or address has been sent but not received the ACK from slave. 1: In slave mode, address is received and matches witih its own address. In master mode, address has been sent and receives the ACK from slave.
0	SBSEND	START signal is sent out in master mode This bit is set by hardware and cleared by reading I2C_STAT0 and writing I2C_DATA. 0: No START signal sent

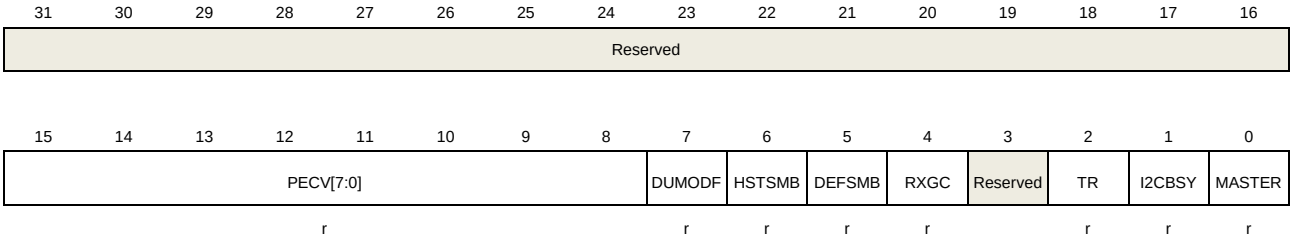
1: START signal sent

18.4.7. Transfer status register 1 (I2C_STAT1)

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.
15:8	PECV[7:0]	Packet Error Checking value that calculated by hardware when PEC is enabled.
7	DUMODF	Dual flag in slave mode indicates which address matches with the address in Dual-Address mode This bit is cleared by hardware after a STOP or a START signal or I2CEN=0 0: The address matches with SADDR0 address 1: The address matches with SADDR1 address
6	HSTSMB	SMBus host header detected in slave mode This bit is cleared by hardware after a STOP or a START signal or I2CEN=0 0: No SMBus host header is detected 1: SMBus host header is detected
5	DEFSMB	Default address of SMBus device This bit is cleared by hardware after a STOP or a START signal or I2CEN=0. 0: The default address has not been received for SMBus device 1: The default address has been received for SMBus device
4	RXGC	General call address (0x00) received. This bit is cleared by hardware after a STOP or a START signal or I2CEN=0. 0: No general call address (0x00) received 1: General call address (0x00) received
3	Reserved	Must be kept at reset value.
2	TR	Transmitter or receiver This bit indicates whether the I2C is a transmitter or a receiver. It is cleared by hardware after a STOP or a START signal or I2CEN=0 or LOSTARB=1. 0: Receiver

		1: Transmitter
1	I2CBSY	Busy flag This bit is cleared by hardware after a STOP signal 0: No I2C communication. 1: I2C communication active.
0	MASTER	A flag indicating whether I2C block is in master or slave mode. This bit is set by hardware when a START signal generates. This bit is cleared by hardware after a STOP signal or I2CEN=0 or LOSTARB=1. 0: Slave mode 1: Master mode

18.4.8. Clock configure register (I2C_CKCFG)

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
FAST	DTCY	Reserved		CLKC[11:0]											
rw	rw			rw											

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	FAST	I2C speed selection in master mode 0: Standard speed 1: Fast speed
14	DTCY	Duty cycle in fast mode 0: $T_{low}/T_{high}=2$ 1: $T_{low}/T_{high}=16/9$
13:12	Reserved	Must be kept at reset value.
11:0	CLKC[11:0]	I2C clock control in master mode In standard speed mode: $T_{high}=T_{low}=CLKC*T_{PCLK1}$ In fast speed mode if DTCY=0: $T_{high}=CLKC*T_{PCLK1}$, $T_{low}=2*CLKC*T_{PCLK1}$ In fast speed mode if DTCY=1: $T_{high}=9*CLKC*T_{PCLK1}$, $T_{low}=16*CLKC*T_{PCLK1}$ **Note:** If DTCY is 0, when PCLK1 is an integral multiple of 3, the baud rate will be more accurate. If DTCY is 1, when PCLK1 is an integral multiple of 25, the baud

rate will be more accurate.

18.4.9. Rise time register (I2C_RT)

Address offset: 0x20
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									RISETIME[6:0]						
									rw						

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:0	RISETIME[6:0]	Maximum rise time in master mode The RISETIME value should be the maximum SCL rise time incremented by 1.

19. Serial peripheral interface/Inter-IC sound (SPI/I2S)

19.1. 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.

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.

19.2. Characteristics

19.2.1. SPI characteristics

- Master or slave operation with full-duplex or half-duplex or simplex mode.
- Separate transmission and reception 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.

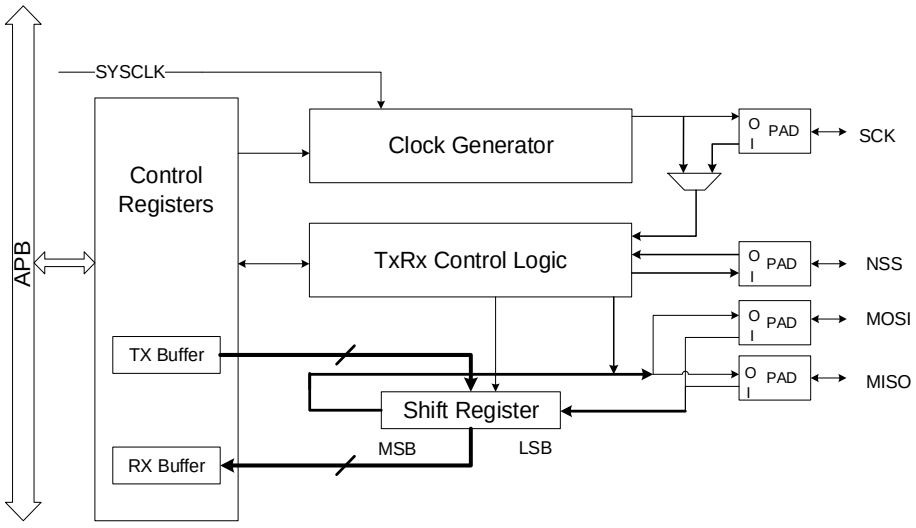
19.2.2. I2S characteristics

- 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.

19.3. SPI function overview

19.3.1. SPI block diagram

Figure 19-1. Block diagram of SPI



19.3.2. SPI signal description

Table 19-1. SPI signal description

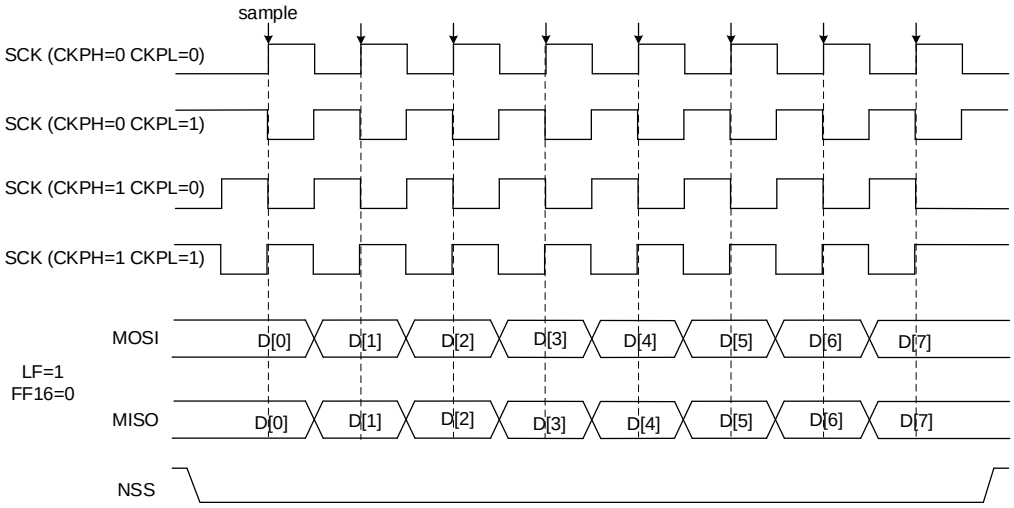
Pin name	Direction	Description
SCK	I/O	Master: SPI clock output Slave: SPI clock input

Pin name	Direction	Description
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 application. Slave in hardware NSS mode: NSS input, as a chip select signal for slave.

19.3.3. 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 SPI is in idle state and CKPH bit decides either first or second clock edge is a valid sampling edge.

Figure 19-2. SPI timing diagram in normal mode



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.

Data order is configured by LF bit in SPI_CTL0 register, and SPI will first send the LSB if LF=1, or the MSB if LF=0.

19.3.4. 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 transmits/receives data only when NSS level is low. In software NSS mode, NSS pin is not used.

Table 19-2. 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 19-3. 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

Mode	Register configuration	Description
		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.

19.3.5. SPI operation modes

Table 19-4. SPI operation 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 BDOEN: Don't care	MOSI: transmission MISO: not used
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	MOSI: not used MISO: transmission

Mode	Description	Register configuration	Data pin usage
		BDEN = 0 BDOEN: Don't care	
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 19-3. A typical full-duplex connection

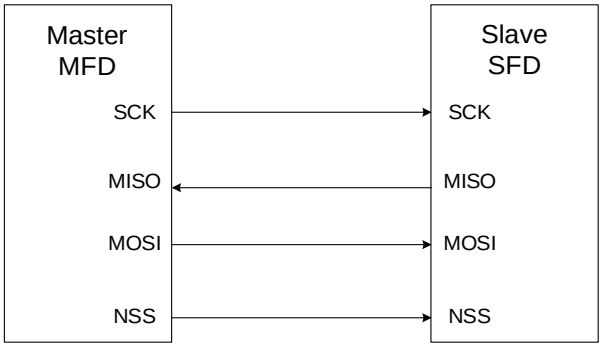


Figure 19-4. A typical simplex connection (Master: receive, Slave: transmit)

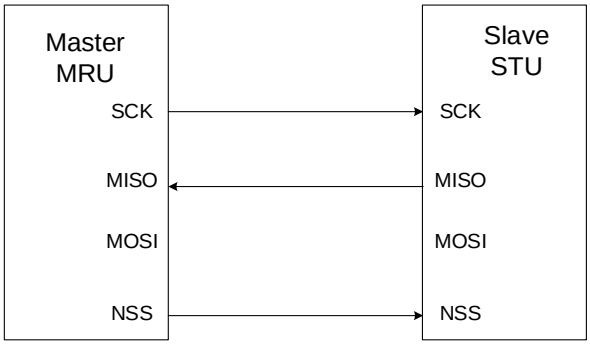


Figure 19-5. A typical simplex connection (Master: transmit only, Slave: receive)

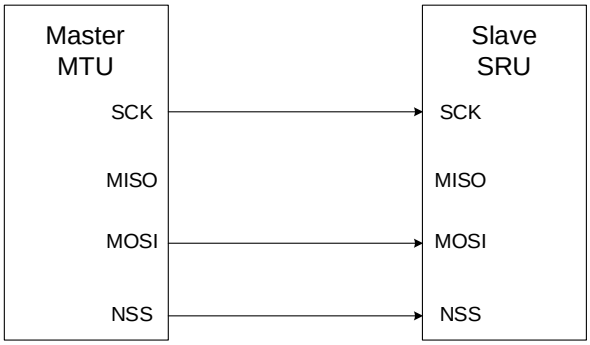
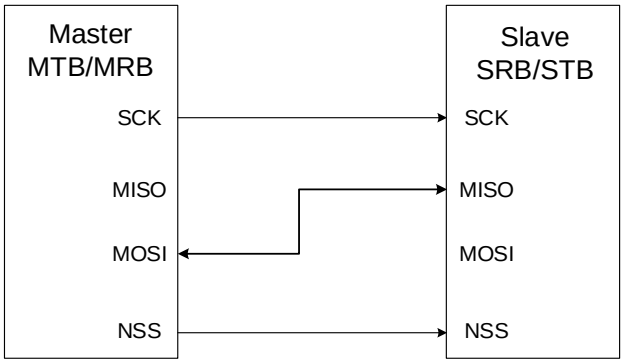


Figure 19-6. A typical bidirectional connection



SPI initialization sequence

Before transmiting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode is used, program the PSC [2:0] bits in SPI_CTL0 register to generate SCK with desired baud rate, otherwise, ignore this step.
2. Configure data format (FF16 bit in the SPI_CTL0 register).
3. Configure the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
4. Configure the frame format (LF bit in the SPI_CTL0 register).
5. Configure the NSS mode (SWNSSEN and NSSDRV bits in the SPI_CTL0 register) according to the application's demand as described above in ***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.

Data order is configured by LF bit in SPI_CTL0 register, and SPI will first send the LSB if LF=1, or the MSB if LF=0.

6. NSS function section.
7. Configure MSTMOD, RO, BDEN and BDOEN depending on the operation modes described in ***Table 19-4. SPI operation modes*** section.
8. 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 transmission buffer. In slave mode the transmission starts when SCK clock signal begins to toggle at SCK pin and NSS level is low, so application should ensure that data is already written into transmission buffer before the transmission starts in slave mode.

When SPI begins to send a data frame, it first loads this data frame from the transmission buffer to the shift register and then begins to transmit the loaded data frame. After TBE flag is set, which means the transmission 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 reception buffer and RBNE will be set. The application should read SPI_DATA register to get the received data and this will clear the RBNE flag automatically when reception buffer is empty. 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 transmission buffer is not empty.

SPI operation sequence in different modes

In full-duplex mode, either MFD or SFD, the RBNE and TBE flags should be monitored and then follow the sequences described above.

The transmission mode (MTU, MTB, STU or STB) is similar to the transmission sequence of full-duplex mode except that the RBNE bit and RXORERR bit need to be ignored.

The master reception mode (MRU or MRB) is different from the reception sequence of full-duplex mode. In MRU or MRB mode, after SPI is enabled, the SPI continuously generates SCK until the SPI is disabled. 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 except that the TBE bit need to be ignored.

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 completed.

19.3.6. 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 configure DMA modules correctly, 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.

19.3.7. 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 calculate CRC for each bit transmitted and received on lines continuously, 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 transmission 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 the DMA function is enabled, the software does not need to set the CRCNT bit, and the hardware will handle the CRC transmission and verification automatically.

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.

When SPI is configured as slave mode and the CRC function is used, the CRC calculation will still be performed even when the NSS pin is high (when the NSS signal is high, as long as there is a clock pulse on the SCK pin, the CRC calculation will continue).This situation occurs when the master device alternately communicates with multiple slave devices. At this time, it is recommended to restart the CRC function when the NSS signal is low.When the slave device is not selected (NSS signal is high) and switches to being selected as a new slave device (NSS signal is low), in order to maintain the synchronization of the next CRC calculation results between the master and slave devices, the CRC values at both ends should be cleared. It is recommended to clear the CRC value according to the following steps:

1. Disable the SPI module (SPIEN=0);
2. Clear the CRCEN bit (CRCEN=0);
3. Set the CRCEN bit (CRCEN=1);
4. Enable the SPI module (SPIEN=1).

19.3.8. SPI interrupts

Status flags

- **Transmission buffer empty flag (TBE)**

This bit is set when the transmission buffer is empty, the software can write the next data to the transmission buffer by writing the SPI_DATA register.

- **Reception buffer not empty flag (RBNE)**

This bit is set when reception buffer is not empty, which means that one data is received and stored in the reception 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 conditions

- **Configuration fault error (CONFERR)**

CONFERR is an error flag in master mode. In NSS hardware mode and 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 and MSTMOD bit are write protection 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 reception buffer contents won't be covered with the newly incoming data, so the newly incoming data is lost.

- **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.

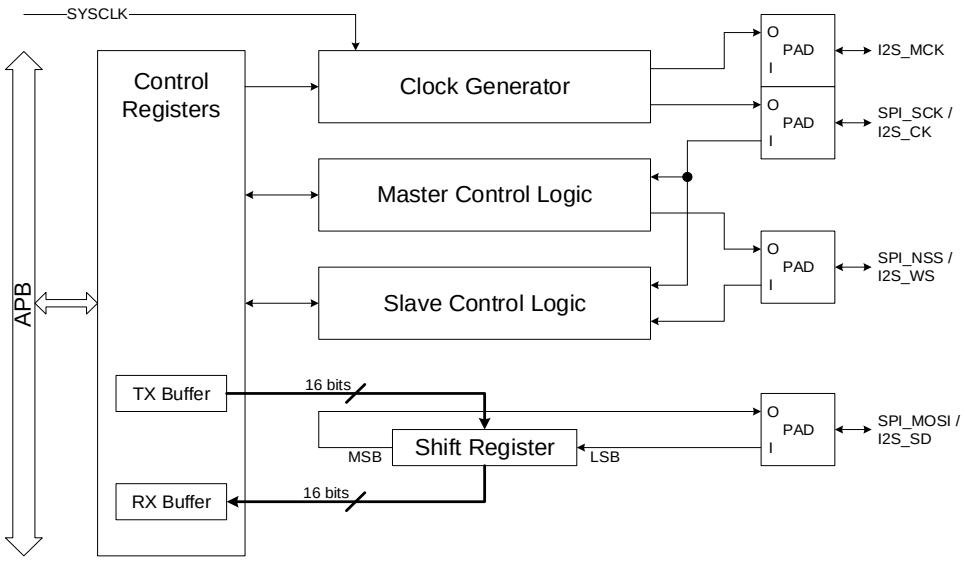
Table 19-5. SPI interrupt requests

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	

19.4. I2S function overview

19.4.1. I2S block diagram

Figure 19-7. 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 I2SCK and I2S_WS. The shift register handles the serial data transmission and reception on I2S_SD.

19.4.2. 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 equal to 256 x Fs, and Fs is the audio sampling frequency.

19.4.3. 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-multiplexed 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 bit 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-bit, 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, and I2S_WS becomes valid one clock before the data. The timing diagrams for each configuration are shown below.

Figure 19-8. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

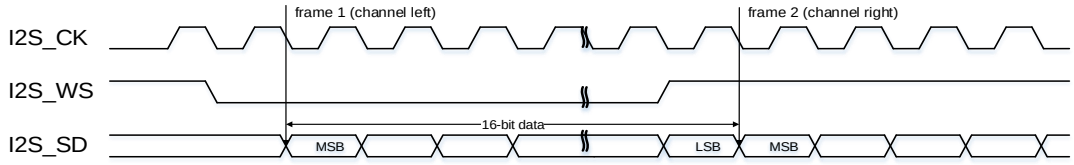
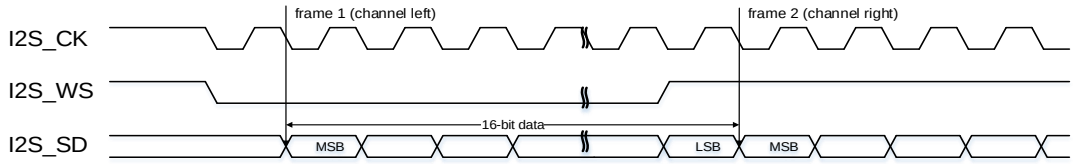


Figure 19-9. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)



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 19-10. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

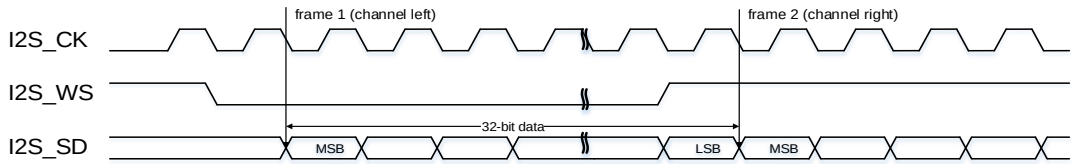
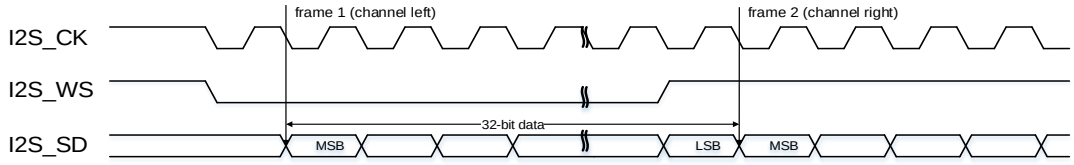


Figure 19-11. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)



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 higher 16 bits, and the second one should be the lower 16 bits.

Figure 19-12. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

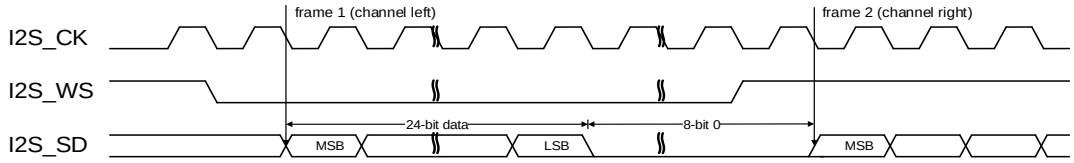
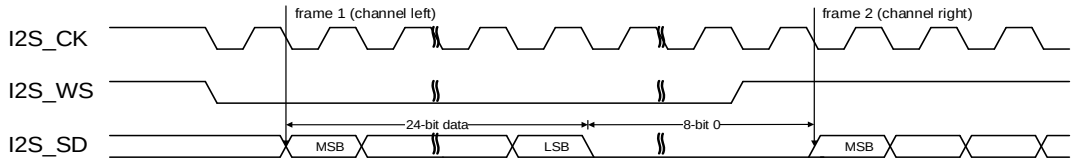


Figure 19-13. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



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 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 19-14. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

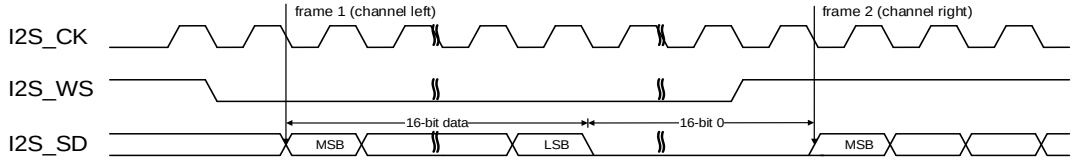
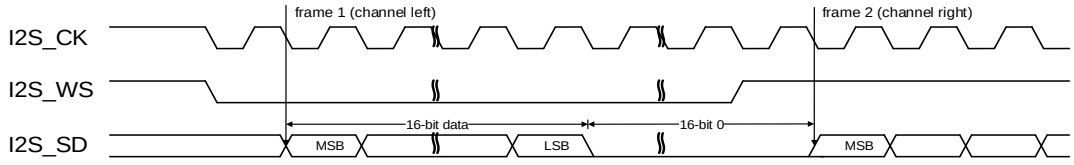


Figure 19-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 19-16. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

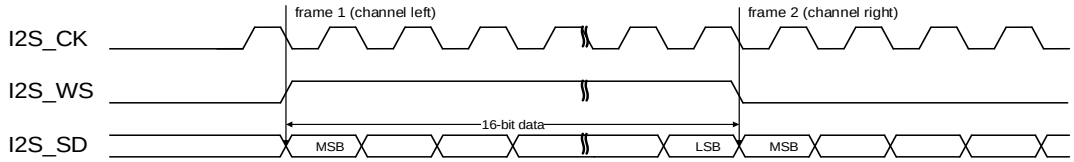


Figure 19-17. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

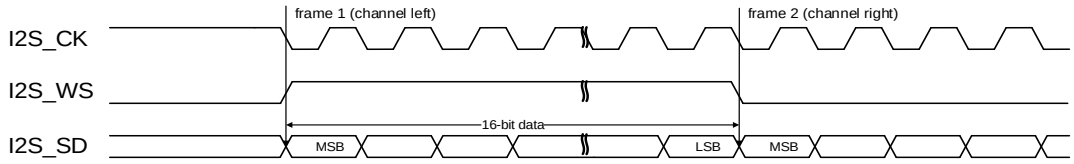


Figure 19-18. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

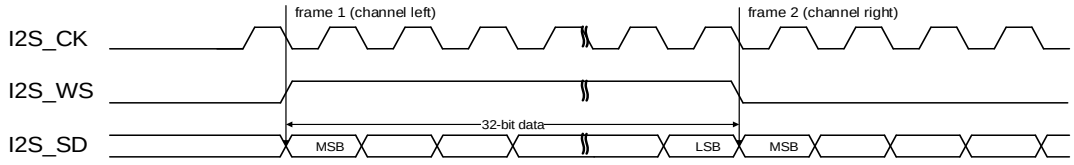


Figure 19-19. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

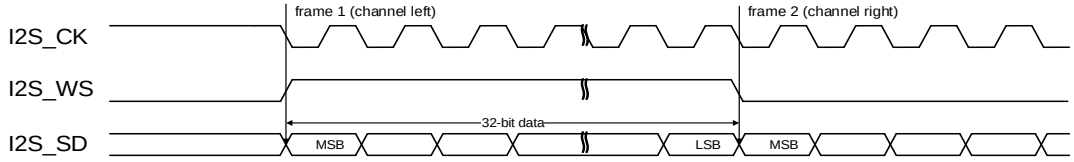


Figure 19-20. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

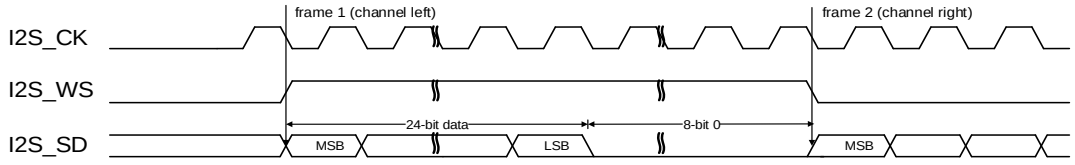


Figure 19-21. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

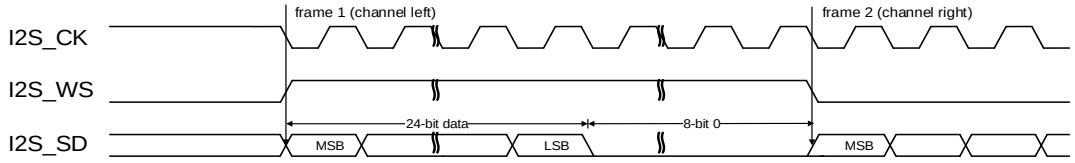


Figure 19-22. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

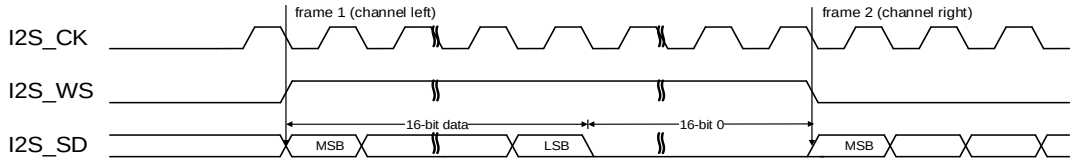
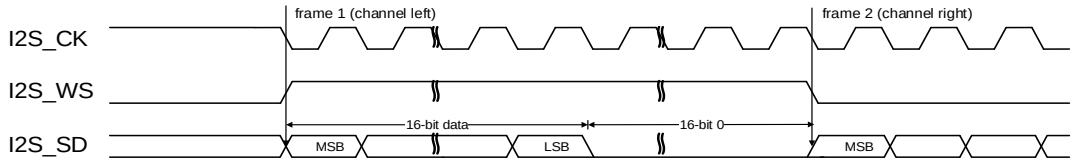


Figure 19-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 19-24. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

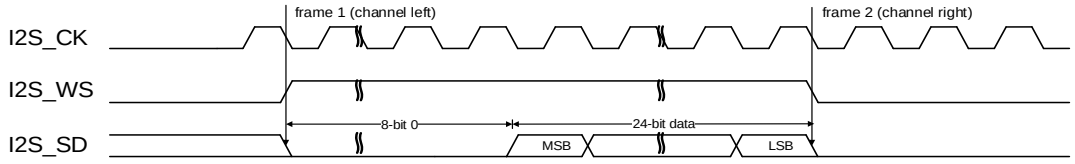
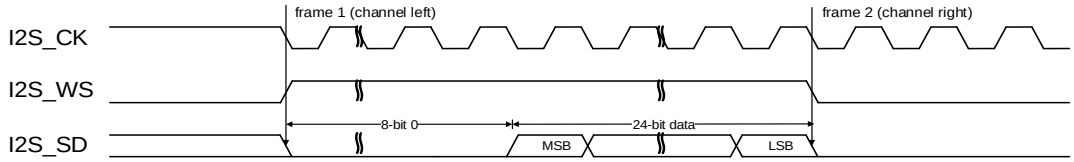


Figure 19-25. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



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 19-26. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

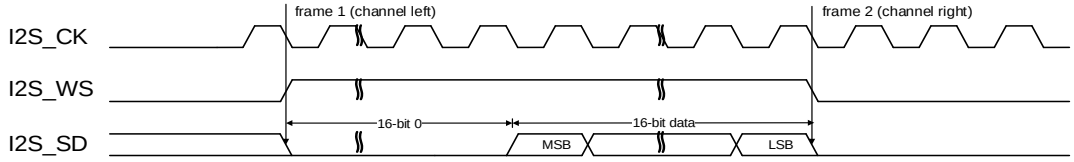
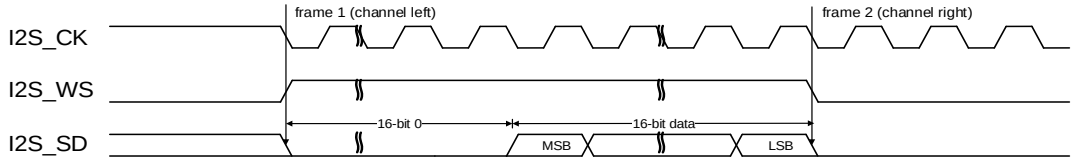


Figure 19-27. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 19-28. PCM standard short frame synchronization mode timing diagram

(DTLEN=00, CHLEN=0, CKPL=0)

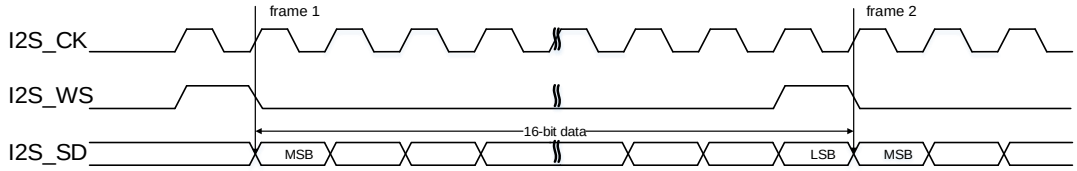


Figure 19-29. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

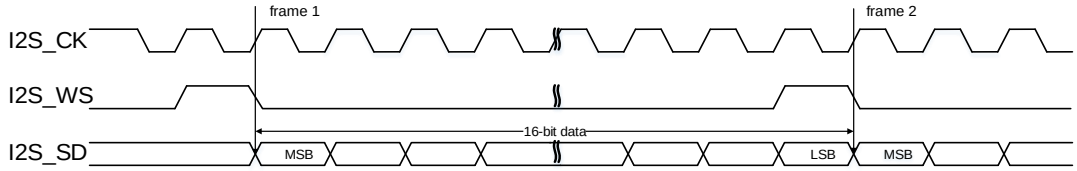


Figure 19-30. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

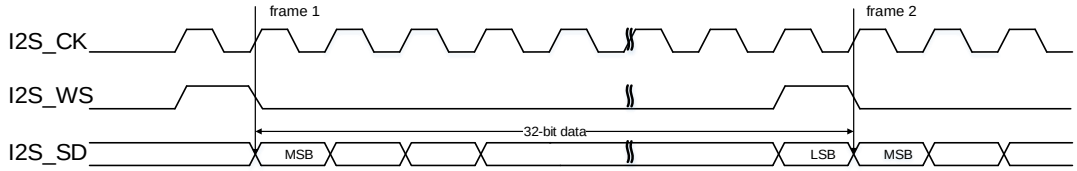


Figure 19-31. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

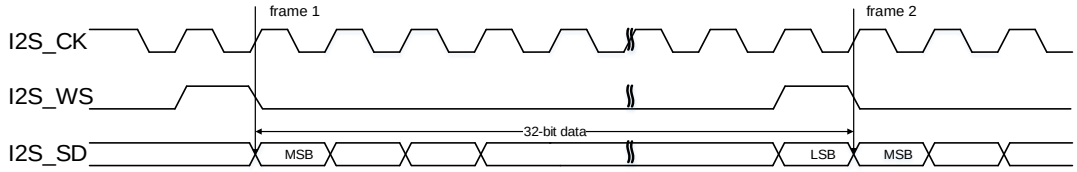


Figure 19-32. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

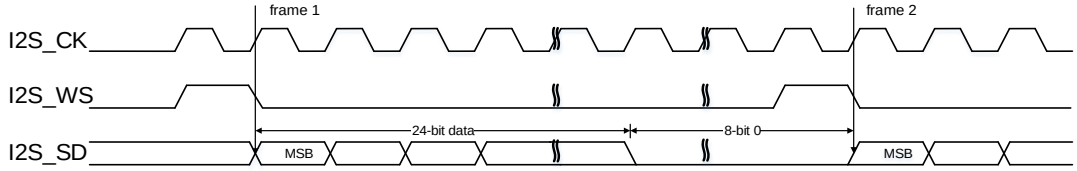


Figure19-33. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

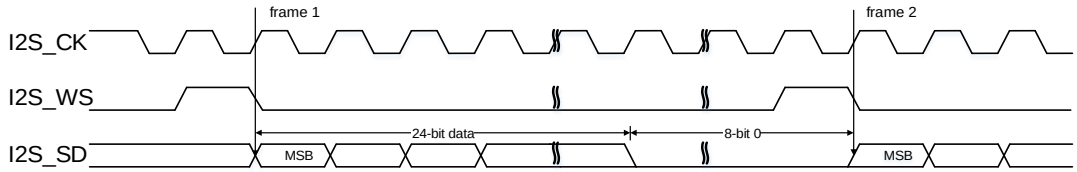


Figure 19-34. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

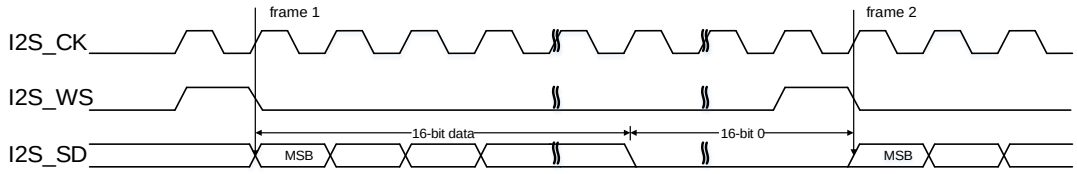
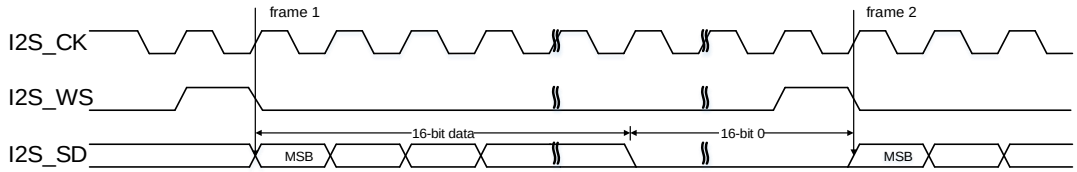


Figure 19-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 19-36. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

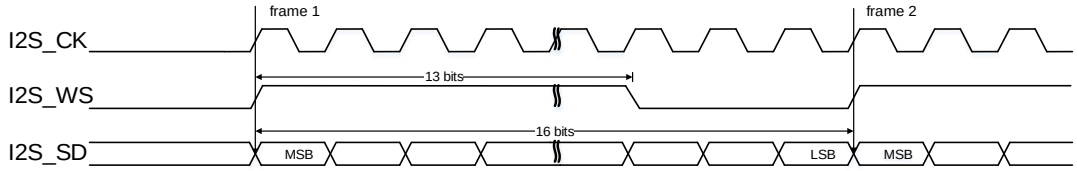


Figure19-37. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

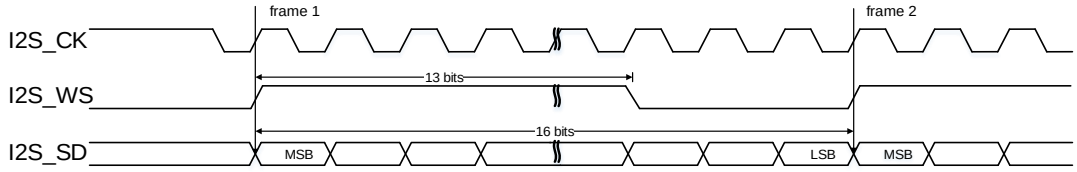


Figure 19-38. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

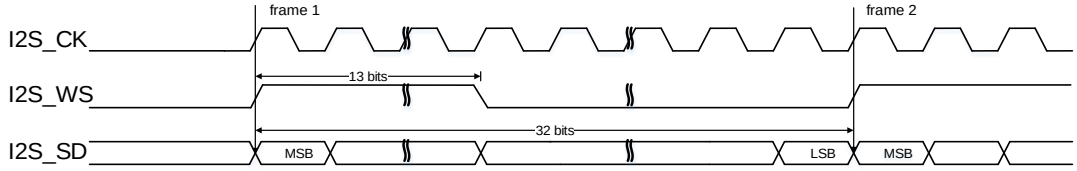


Figure 19-39. PCM standard long frame synchronization mode timing diagram

(DTLEN=10, CHLEN=1, CKPL=1)

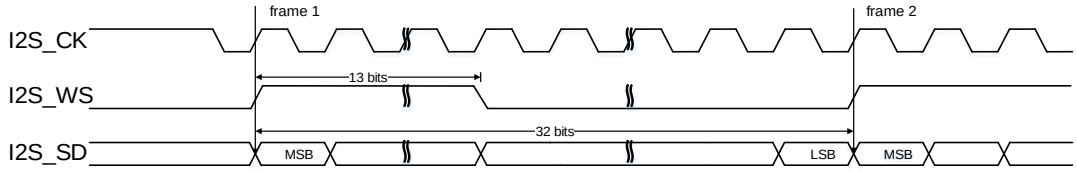


Figure 19-40. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

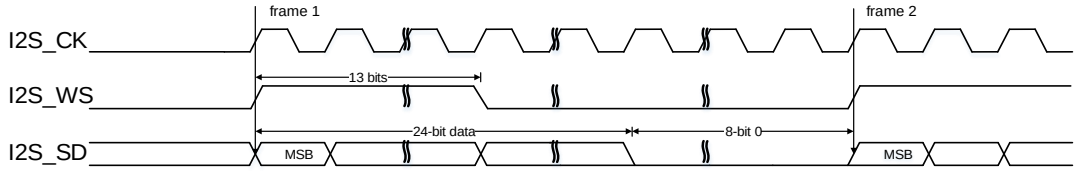


Figure 19-41. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

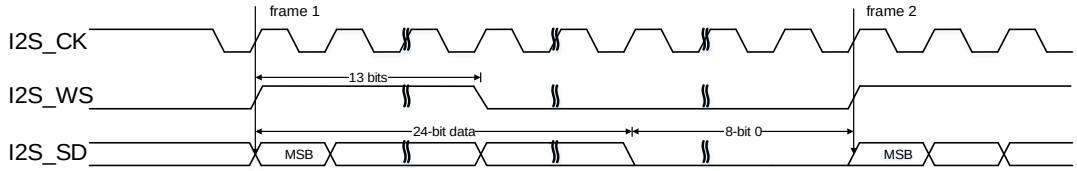


Figure 19-42. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

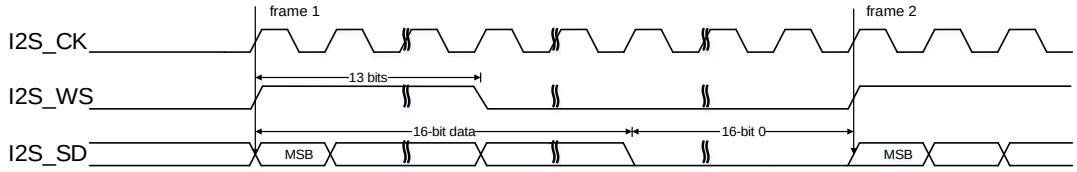
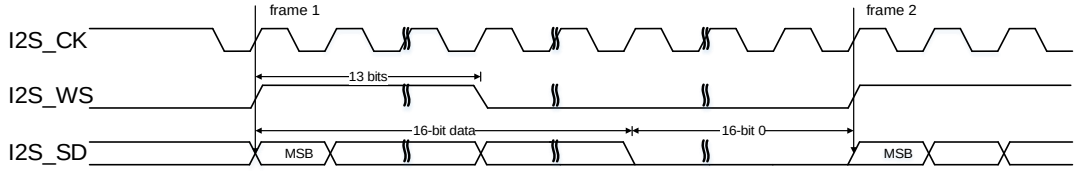
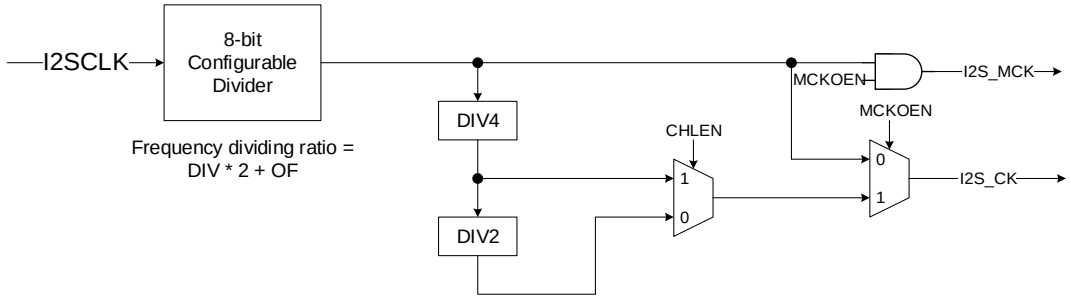


Figure 19-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



19.4.4. I2S clock

Figure 19-44. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as ***Figure 19-44. 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 19-6. I2S bitrate calculation formulas***.

Table 19-6. 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 19-7. Audio sampling frequency calculation formulas***.

Table 19-7. 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).

19.4.5. Operation

Operation modes

The operation mode is selected by the I2SOPMOD[1:0] 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 19-8. Direction of I2S interface signals for each operation mode***.

Table 19-8. Direction of I2S interface signals for each operation mode

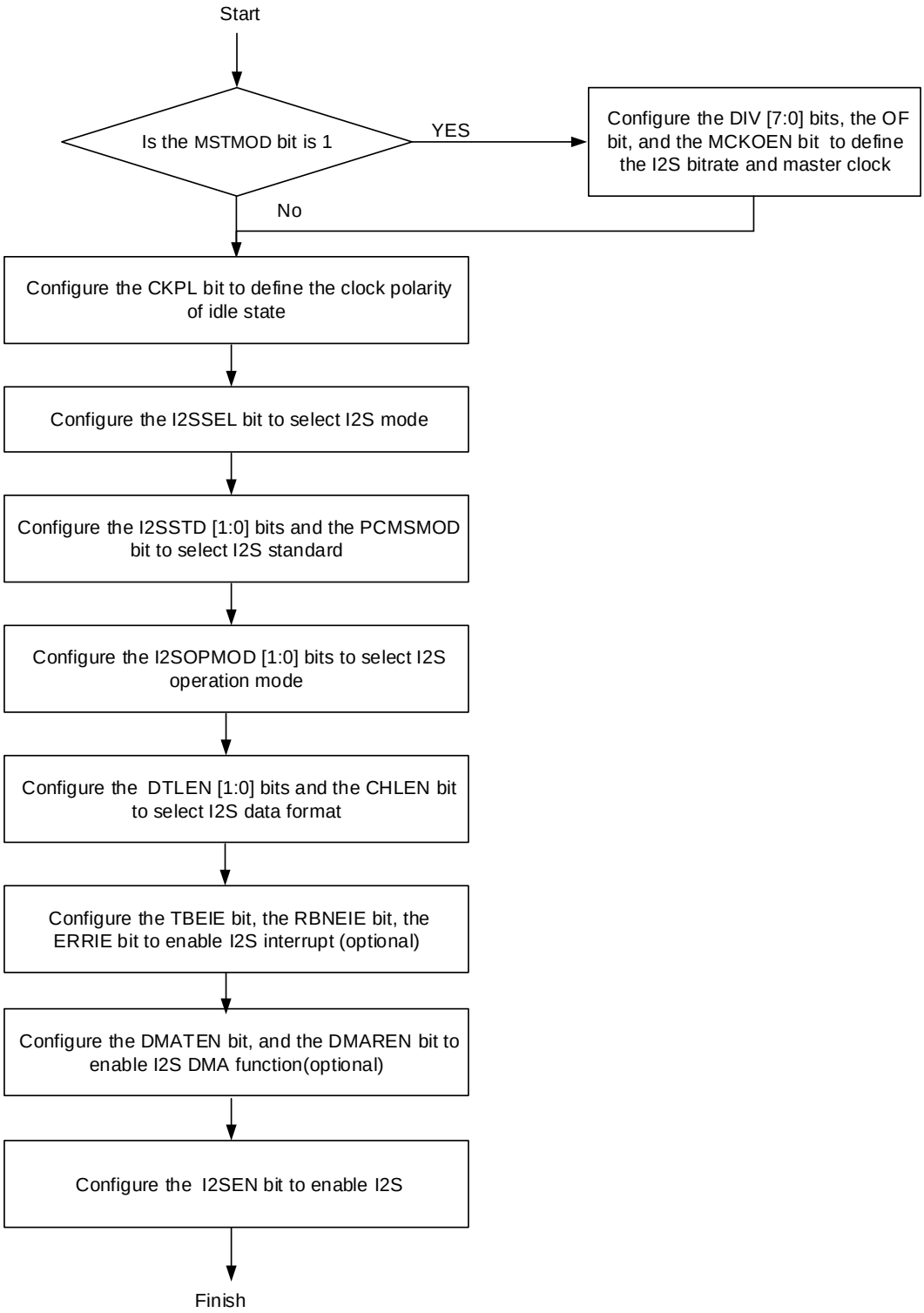
Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
Master transmission	Output or NU(1)	Output	Output	Output
Master reception	Output or NU(1)	Output	Output	Input
Slave transmission	Input or NU(1)	Input	Input	Output
Slave reception	Input or NU(1)	Input	Input	Input

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 below ***Figure 19-45. I2S initialization sequence.***

Figure 19-45. 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 transmission buffer is empty, and an interrupt will be generated if the

TBEIE bit in the SPI_CTL1 register is set. At the beginning, the transmission 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 transmission 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 transmission 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 side 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.

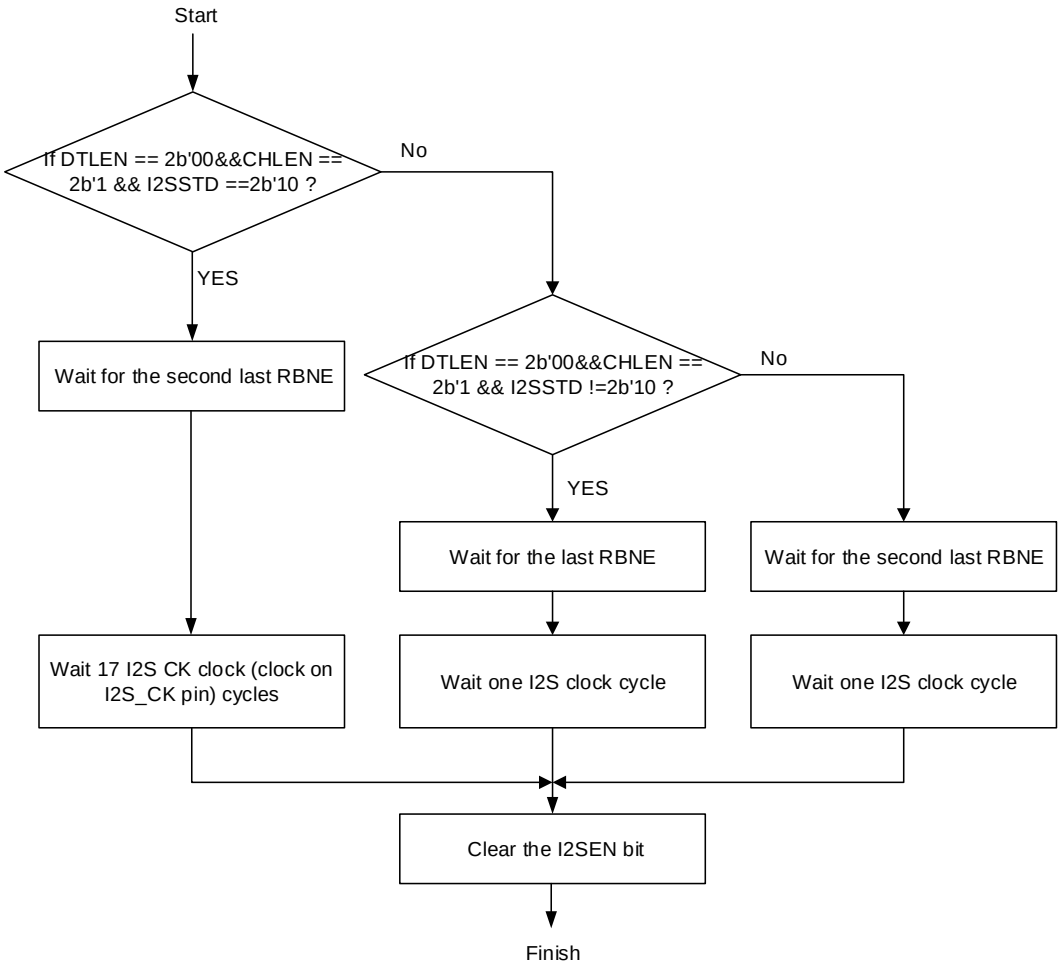
I2S master reception sequence

The RBNE flag is used to control the reception sequence. As is mentioned before, the RBNE flag indicates the reception 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 reception buffer is empty (RBNE is low). When a reception sequence finishes, the received data in the shift register is loaded into the reception 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 the channel side which 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 shown as below ***Figure 19-46. I2S master reception disabling sequence***.

Figure 19-46. 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.

19.4.6. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

19.4.7. 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.

- **Transmission buffer empty flag (TBE)**

This bit is set when the transmission buffer is empty, the software can write the next data to the transmission buffer by writing the SPI_DATA register.

- **Reception buffer not empty flag (RBNE)**

This bit is set when reception buffer is not empty, which means that one data is received and stored in the reception 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. In the transmission mode, the I2SCH flag is updated every time TBE changes from 0 to 1. In the reception mode, the I2SCH flag is updated every time RBNE changes from 0 to 1. This flag will not generate any interrupt.

Error conditions

There are three error flags:

- **Transmission underrun error flag (TXURERR)**

This situation occurs when the transmission buffer is empty when the valid SCK signal starts in slave transmission mode.

- **Reception overrun error flag (RXORERR)**

This situation occurs when the reception buffer is full and a newly incoming data has been completely received. When overrun occurs, the data in reception 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 enabled bits are summed up in the ***Table 19-9. I2S interrupt.***

Table 19-9. I2S interrupt

Interrupt flag	Description	Clear method	Interrupt enable bit
TBE	Transmission buffer empty	Write SPI_DATA register	TBEIE
RBNE	Reception 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	

19.5. Register definition

SPI0/I2S0 base address: 0x4001 3000

SPI1 base address: 0x4000 3800

SPI2/I2S2 base address: 0x4000 3C00

19.5.1. Control register 0 (SPI_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by byte (8-bit) or 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	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 data transfers between the MOSI pin of master and the MISO pin of 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: Disable CRC calculation. 1: Enable CRC calculation.
12	CRCNT	CRC next transfer 0: Next transfer is data 1: Next transfer is CRC value 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

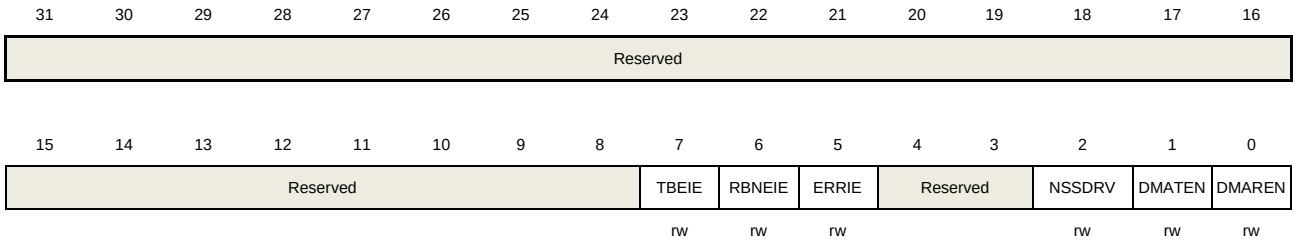
		SPI_DATA register. In receive only mode, set this bit after the second last data is received.
11	FF16	Data frame format 0: 8-bit data frame format 1: 16-bit data frame format
10	RO	Receive only When BDEN is cleared, this bit determines the direction of transfer. 0: Full-duplex mode 1: Receive-only mode
9	SWNSSEN	NSS software mode selection 0: NSS hardware mode. The NSS level depends on NSS pin. 1: NSS software mode. The NSS level depends on SWNSS bit.
8	SWNSS	NSS pin selection in NSS software mode 0: NSS pin is pulled low. 1: NSS pin is pulled high. This bit has an effect only when the SWNSSEN bit is set.
7	LF	LSB first mode 0: Transmit MSB first 1: Transmit LSB first
6	SPIEN	SPI enable 0: Disable SPI peripheral. 1: Enable SPI peripheral.
5:3	PSC[2:0]	Master clock prescaler selection 000: PCLK/ 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: 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.

19.5.2. Control register 1 (SPI_CTL1)

Address offset: 0x04
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
7	TBEIE	Transmit buffer empty interrupt enable 0: Disable TBE interrupt 1: Enable TBE interrupt. An interrupt is generated when the TBE bit is set
6	RBNEIE	Receive buffer not empty interrupt enable 0: Disable RBNE interrupt 1: Enable RBNE interrupt. An interrupt is generated when the RBNE bit is set.
5	ERRIE	Errors interrupt enable 0: Disable error interrupt 1: Enable error interrupt. An interrupt is generated when the CRCERR bit or the CONFERR bit or the RXORERR bit or the TXURERR bit is set.
4:3	Reserved	Must be kept at reset value.
2	NSSDRV	Drive NSS output 0: Disable master NSS output 1: Enable master NSS output
1	DMATEN	Transmit buffer DMA enable 0: Disable transmit buffer DMA 1: Enable transmit buffer DMA, when the TBE bit in SPI_STAT is set, it will be a DMA request on corresponding DMA channel.
0	DMAREN	Receive buffer DMA enable 0: Disable receive buffer DMA 1: Enable receive buffer DMA, when the RBNE bit in SPI_STAT is set, it will be a

DMA request on corresponding DMA channel.

19.5.3. Status register (SPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0002

This register can be accessed by byte (8-bit) or 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 I2S Mode: 0: No I2S format error 1: I2S format error occurs. This bit is set by hardware and is able to be 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 is able to be cleared by writing 0.

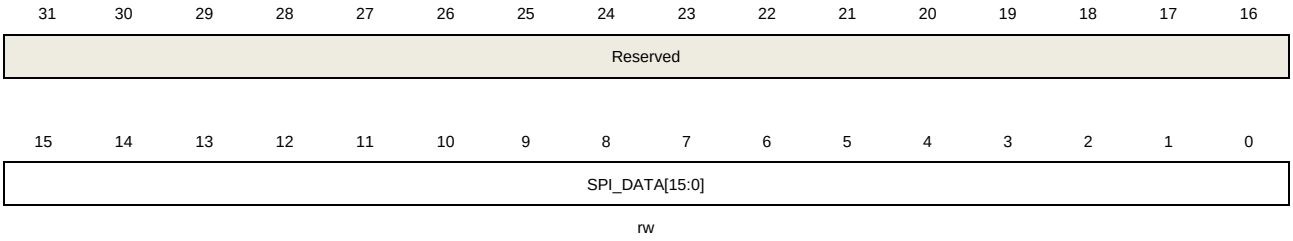
		This bit is not used in I2S mode.
3	TXURERR	Transmission underrun error bit 0: No transmission underrun error occurs. 1: Transmission underrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_STAT register. This bit is not used in SPI mode.
2	I2SCH	I2S channel side 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.

19.5.4. Data register (SPI_DATA)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit) or 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 transmission buffer and reception buffer. Write data to SPI_DATA will save the data to transmission buffer and read data from SPI_DATA will get the data from reception 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, transmission buffer and reception buffer are 8-bits. If the Data frame format is set to 16-bit data, the SPI_DATA [15:0] is used for

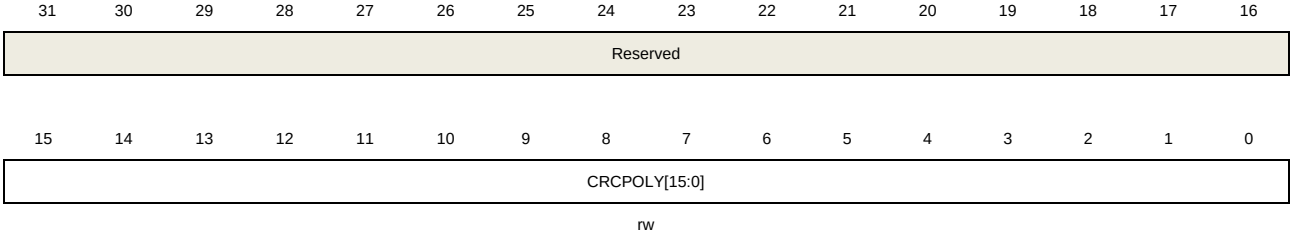
transmission and reception, transmission buffer and reception buffer are 16-bit.

19.5.5. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

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



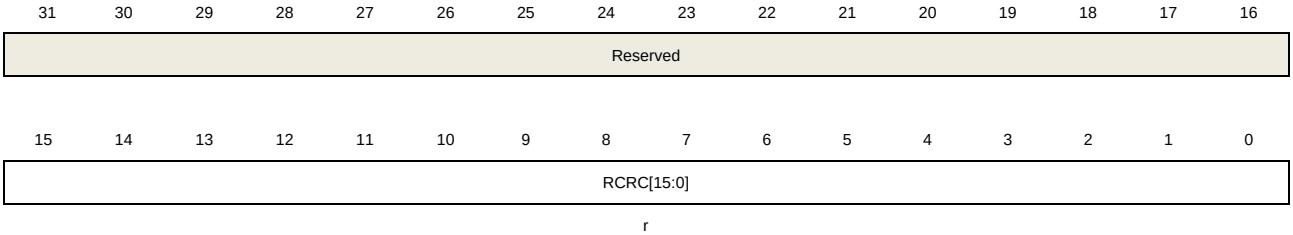
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRCPOLY[15:0]	CRC polynomial register This register contains the CRC polynomial and it is used for CRC calculation. The default value is 0007h.

19.5.6. RX CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit) or 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 register When the CRCERRN 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.

19.5.7. TX CRC register (SPI_TCRC)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit) or 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
TCRC[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	TX CRC register 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 format (LF bit of the SPI_CTL0) will get different CRC value. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

19.5.8. I2S control register (SPI_I2SCTL)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by byte (8-bit) or 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 mode or I2S mode is disabled.
10	I2SEN	I2S enable 0: Disable I2S 1: Enable I2S 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 mode 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 mode 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 mode 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 mode 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.

19.5.9. I2S clock prescaler register (SPI_I2SPSC)

Address offset: 0x20

Reset value: 0x0000 0002

This register can be accessed by byte (8-bit) or 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						MCKOEN	OF	DIV[7:0]							
						rw	rw	rw							

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	MCKOEN	I2S_MCK output enable 0: Disable I2S_MCK output 1: Enable I2S_MCK output This bit should be configured when I2S mode 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 mode 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 mode is disabled. These bits are not used in SPI mode.

20. HDMI-CEC controller(HDMI-CEC)

20.1. Overview

The products of the GD32F150xx series integrate the HDMI-CEC controller inside to support the CEC protocol. Consumer Electronics Control (CEC) belongs to a part of HDMI (High-Definition Multimedia Interface) standard. CEC as a kind of protocol, provides the advanced control functions of all kinds of audio-visual products in a user environment. Users can flexibly implement control functions through the HDMI-CEC controller.

20.2. Characteristics

- HDMI-CEC controller complies with HDMI-CEC v1.4 Specification
- Two clock source options for 32.768KHz CEC clock:
  - LXTAL oscillator
  - IRC8M oscillator with settled prescaler (IRC8M/244)
- For ultra low-power applications, HDMI-CEC controller can work in Deep-sleep mode
- Programmable SFT(Signal Free Time) value for arbitration priority:
  - User configure
  - Auto configure by controller as HDMI-CEC protocol specification
- Programmable own address(OAD)
- Listen mode supports user receiving messages on the CEC line but not disturb the CEC line.
- Receive bit-tolerance function support for higher compatibility
- Supports the function of detecting various error states
  - Bit error: Bit period short error(BPSE), Bit period long error(BPLE), Bit rising error(BRE)
  - Transmission error(TERR)
  - Transmission underrun (TU)
  - Reception overrun (RO)
  - Arbitration fail (ARBF)
- Programmable error-bit generation
  - BPSE detection will always generate error-bit
  - BPLE detection will generate error-bit if BPLEG=1
  - BRE detection will generate error-bit if BREG=1

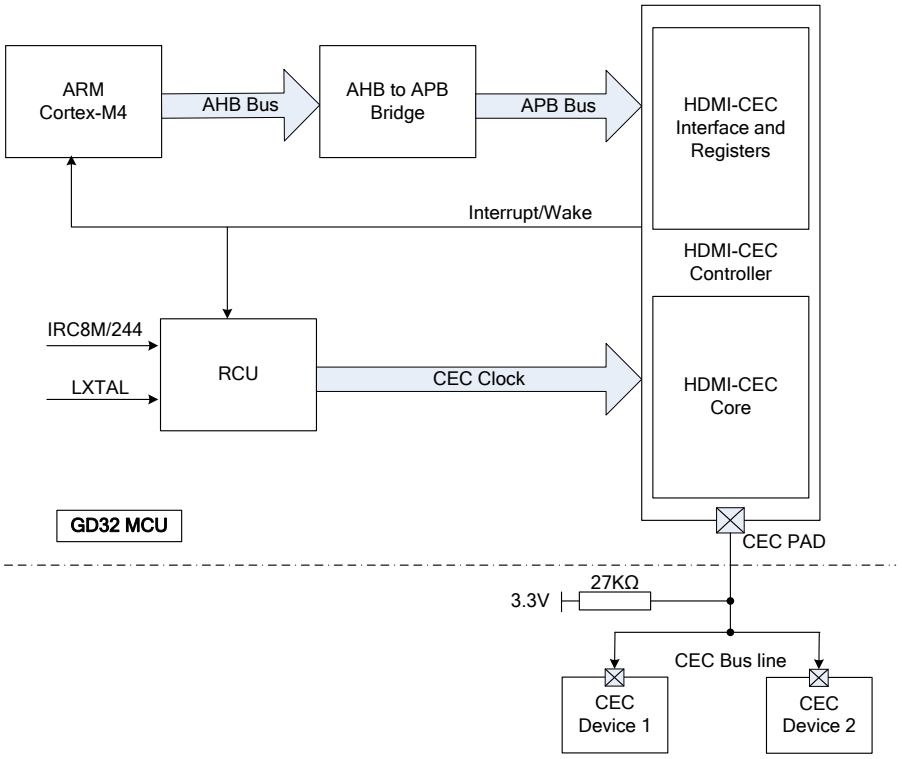
20.3. Function overview

20.3.1. CEC bus pin

The CEC device communicates with others by only one bidirectional line. When the CEC

device is in the output state, in order to allow a wired-and connection, the CEC pin need to be configured in alternate function open drain mode, and an external 27kΩ resister is needed for pulling-up the CEC pin to a +3.3V supply voltage.

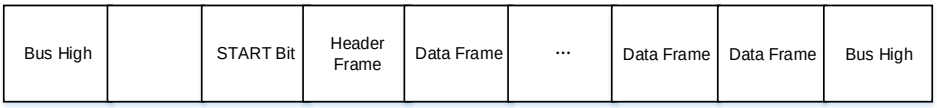
Figure 20-1. HDMI-CEC Controller block diagram



20.3.2. Message description

A complete message includes one or more frames and the message structure is shown as below:

Figure 20-2. Message Structure



The frame has two types:

1) **Header frame**: The first frame in the message which followed the start-bit consists of the source logical address field and the destination logical address field. The Header frame is always needed.

2) **Data frame**: The frames in the message followed the header frame. Data frame is optional.

All frames are ten bits long and have the same basic structure as shown below:

Table 20-1. Frame Structure

Frame Structure									
7	6	5	4	3	2	1	0		

Information bits	ENDOM	ACK

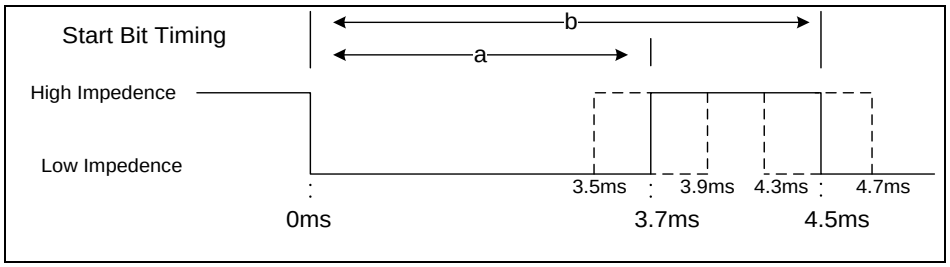
The information bits are data, opcodes or addresses, dependent on context. The control bits, ENDOM and ACK, are always present and always have the same usage.

20.3.3. Bit timing description

All bits timing in the message are divided into two types: Start bit and Data bit.

1) Start Bit: The start bit has to be validated by its low duration(a) and its total duration(b) showed as below:

Figure 20-3. Start Bit Timing



2) Data Bit: The valid data bit timing is constrained as below:

Figure 20-4. Data Bit Timing

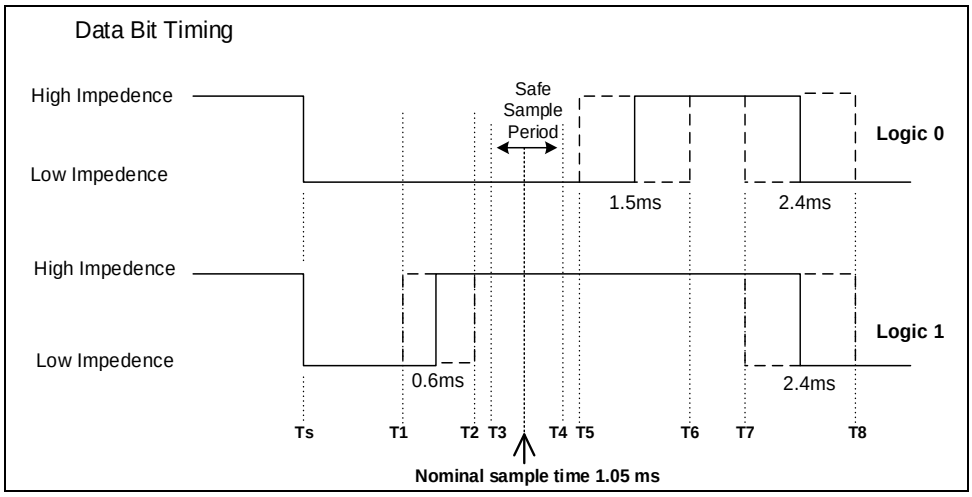


Table 20-2. Data Bit Timing parameter table

Ts	Time (ms)	The bit start event.
T1	0.4ms	When indicating a logical 1, T1 as the earliest time for a low - high transition.
T2	0.8ms	When indicating a logical 1, T2 as the latest time for a low - high transition.
T3	0.85ms	The earliest time it is safe to sample the signal line to determine its state.
T4	1.25ms	The latest time it is safe to sample the signal line to determine its state.
T5	1.3ms	T5 as the earliest time that a device is allowed to return to a high impedance state(logical 0).
T6	1.7ms	T6 as the latest time that a device is allowed to return to a high impedance

Ts	Time (ms)	The bit start event.
		state(logical 0).
T7	2.05ms	T7 as the earliest time for the start of a following bit.
	2.4ms	As a nominal data bit period.
T8	2.75ms	T8 as the latest time for the start of a following bit.

20.3.4. Arbitration

CEC line arbitration starts from the front edge of the start bit to the end of the Initiator address bits among the Header Block. In the meantime the Initiator should monitor the CEC line. During this period, if low impedance is detected when sending high impedance state then it should assume that it has lost the arbitration.

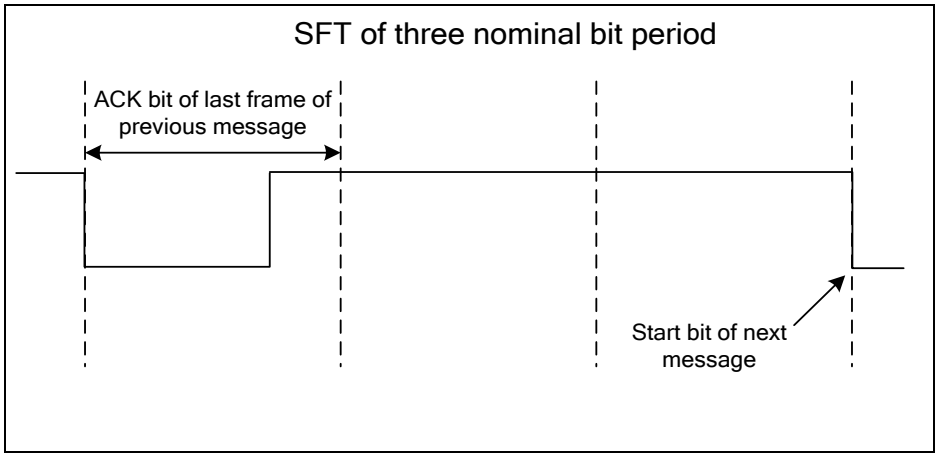
Figure 20-5. The process of CEC line arbitration



Before attempting to transmit or re-transmit a frame, a device shall ensure that the CEC line has been inactive for a number of bit periods.

This signal free time is defined as the time since the start of the final bit of the previous frame.

Figure 20-6. The free time of signal



The length of the required signal free time depends on the current status of the control signal line and the initiating device. If SFT=0x0, the HDMI-CEC controller's SFT will perform as table below:

Table 20-3. The relationship betwwen Signal Free Time and precondition

Precondition	Signal Free Time (nominal data bit periods)
Present Initiator wants to send another message immediately after its previous message	≥7
New Initiator wants to send a message	≥5

Precondition	Signal Free Time (nominal data bit periods)
Previous attempt to send message unsuccessful	≥3

This means that there is an opportunity for other devices to gain access to the CEC line during the periods mentioned above to send their own messages after the current device has finished sending its current message.

If SFT is not 0x0, the corresponding user configure SFT will be performed.

20.3.5. SFT option bit description

SFT option bit support another way for saving bus inactive time through setting more SFT counter's start time point.

When SFTOPT = 0, the SFT timer will start at the STAOM bit asserted when the controller is in the idle state.

When SFTOPT = 1, the SFT timer will start at the CEC bus is in idle state and the SFT time will be saved if you configure the STAOM after SFT done because the controller will start transmit without any latency.

When SFTOPT = 1, some other cases will also start the SFT counter:

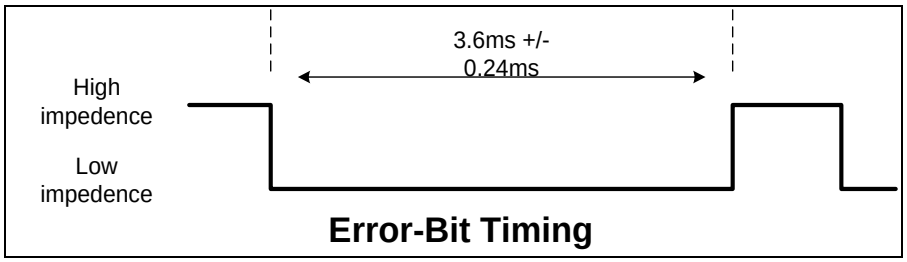
- In case of regular TX/RX complete(TEND/REND asserted)
- In case of transmission not complete such as the TERR, TAERR or TU asserted.
- In case of receiving progress, if some error detected and error bit is generated, the SFT timer will start when output error bit finished.

20.3.6. Error definition

Error-Bit

If some errors are occurred and corresponding generation configure is enabled the HDMI-CEC controller will generate an error bit on the CEC pin for indicating. Error bit period definition is shown as below:

Figure 20-7. Erro bit period



Frame error

CEC protocol defines that each frame of message need the acknowledgement to confirm the communication is successful. For broadcast(destination address=0xF), the ACK bit should be logic 1 and for singlecast(destination address<0xF), the ACK bit should be logic 0, otherwise the frame error occurs(TAERR/RAE flag asserted).

Another frame error situation is that the CEC bus pin voltage is different from CEC pad when HDMI-CEC controller is under initiator state(TERR flag asserted).

Bit rising error(BRE)

The BRE bit indicates whether rising edge is detected during the BRE checking window. If BREIE=1, the CEC interrupt is generated after BRE is set.

If BRES=1, the controller will stop receiving message and if BREG=1 the error-bit will be generated.

If BRES=1 in broadcast the BRE flag asserted, the error bit will be generated to notify the initiator the error. If you do not want to generate the error bit for BRE detection you can configure the BREG=0 and BCNG=1.

Note: The BRES=0 and BREG=1 configuration must be avoided.

Bit period short error(BPSE)

The BPSE bit indicates whether the period of the neighboring falling edge is shorter than expected. If BPSEIE=1, the CEC interrupt is generated after BPSE is set.

If the BPSE flag asserted, the error bit will must be generated except one of the cases below:

1) BCNG = 1

2) LMEN = 1

3) Receiving broadcast

Bit period long error(BPLE)

The BPLE bit indicates whether the period of the neighboring falling edge is longer than expected. If BPLEIE=1, the CEC interrupt is generated after BPLE is set.

When BPLE asserted, controller will stop receiving message and generate error bit if in one of the cases below:

1) BPLEG=1 in both singlecast and broadcast

2) BCNG=0 in broadcast

Figure 20-8. The timing of Bit Period Long Error

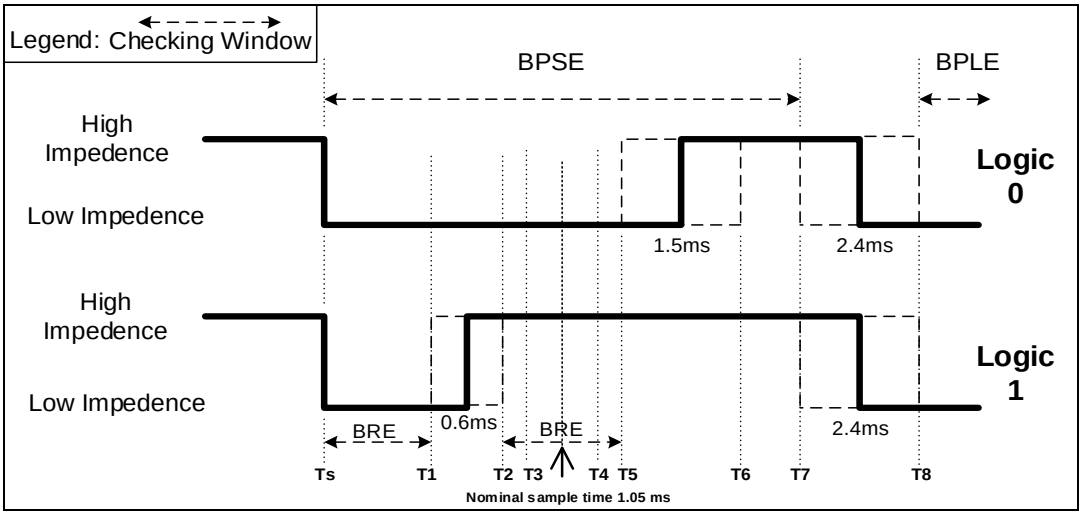


Table 20-4. Error Handling Timing parameter table

Symbol	RTOL	Time(ms)	Description
Ts	-	0ms	The bit start event.
T1	1	0.3ms	When indicating a logical 1, T1 as the earliest time for a low - high transition.
	0	0.4ms	
T2	0	0.8ms	When indicating a logical 1, T2 as the latest time for a low - high transition.
	1	0.9ms	
T3	-	0.85ms	The earliest time it is safe to sample the signal line to determine its state.
T4	-	1.25ms	The latest time it is safe to sample the signal line to determine its state.
T5	1	1.2ms	T5 as the earliest time that a device is allowed to return to a high impedance state(logical 0).
	0	1.3ms	
T6	0	1.7ms	T6 as the latest time that a device is allowed to return to a high impedance state(logical 0).
	1	1.8ms	
T7	1	1.85ms	T7 as the earliest time for the start of a following bit.
	0	2.05ms	
		2.4ms	As a nominal data bit period.
T8	0	2.75ms	T8 as the latest time for the start of a following bit.
	1	2.95ms	

Transmission error detection(TERR)

The TERR is set when the initiator find low impedance on the CEC bus when it is transmitting high impedance. TERR will also generate CEC interrupt if TERRIE=1.

When TERR asserted the transmission is aborted and the software can retry the transmission.

TERR check window is depending on the different bit state of the frame shown as below:

Figure 20-9. Transmission error detection

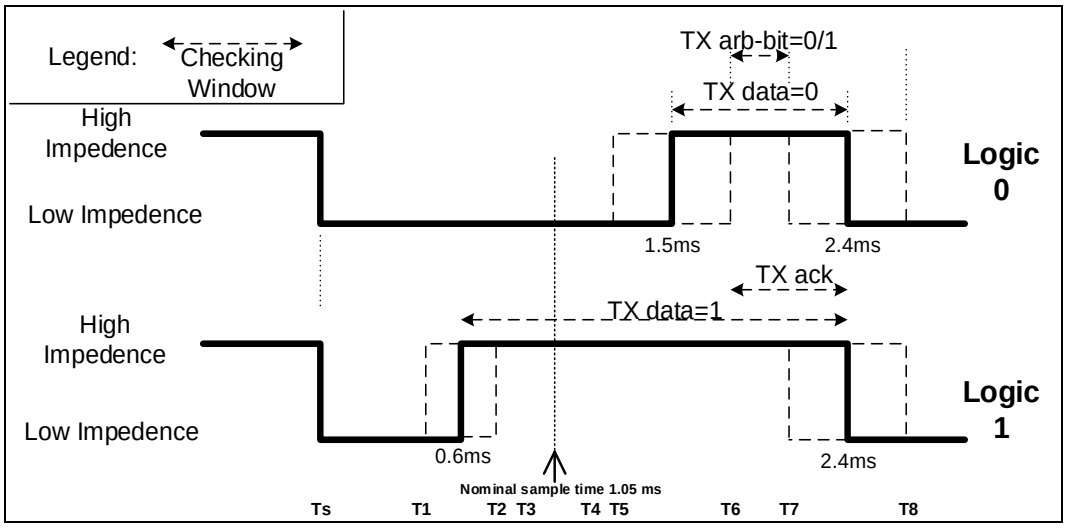


Table 20-5. TERR Timing parameter table

Symbol	RTOL	Time(ms)	Description
Ts	-	0ms	The bit start event.
T1	1	0.3ms	The earliest time for a low - high transition when indicating a logical 1.
	0	0.4ms	
T2	0	0.8ms	The latest time for a low - high transition when indicating a logical 1.
	1	0.9ms	
T3	-	0.85ms	The earliest time it is safe to sample the signal line to determine its state.
T4	-	1.25ms	The latest time it is safe to sample the signal line to determine its state.
T5	1	1.2ms	The earliest time a device is permitted return to a high impedance state (logical 0).
	0	1.3ms	
T6	0	1.7ms	The latest time a device is permitted return to a high impedance state (logical 0).
	1	1.8ms	
T7	1	1.85ms	The earliest time for the start of a following bit.
	0	2.05ms	
		2.4ms	The nominal data bit period.
T8	0	2.75ms	The latest time for the start of a following bit.
	1	2.95ms	

20.3.7. HDMI-CEC interrupt

There 13 interrupts in HDMI-CEC controller are made up of corresponding flag and interrupt enable bit:

Table 20-6. HDMI-CEC Interrupt

No.	Interrupt event in HDMI-CEC	Event flag	Interrupt enable bit
1	Arbitration fail	ARBF	ARBFIE
2	TX Byte Request	TBR	TBRIE

No.	Interrupt event in HDMI-CEC	Event flag	Interrupt enable bit
3	Transmission end	TEND	TENDIE
4	Transmit Byte buffer underrun	TU	TUIE
5	Transmit error	TERR	TERRIE
6	Transmit acknowledge error	TAERR	TAERRIE
7	Byte Received	BR	BRIE
8	Reception end	REND	RENDIE
9	Receive Overrun	RO	ROIE
10	Bit rising error	BRE	BREIE
11	Bit Period Short Error	BPSE	BPSEIE
12	Bit Period Long Error	BPLE	BPLEIE
13	RX acknowledge error	RAE	RAEIE

Note: Any HDMI-CEC interrupt will wake up the chip from Deep-sleep Mode.

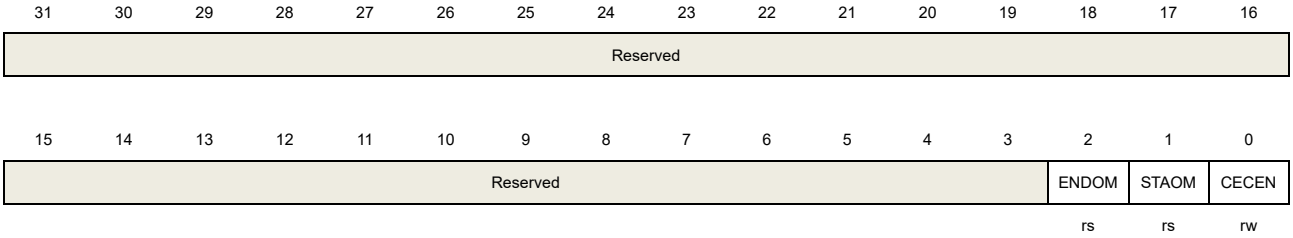
20.4. Register definition

HDMI-CEC base address: 0x4000 7800

20.4.1. Control register (CEC_CTL)

Address offset: 0x00
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	ENDOM	ENDOM bit value in the next frame in Transmit mode. ENDOM can only be set by software when CECEN=1. ENDOM is cleared by hardware in the same situation of clearing STAOM. 0: Next frame send 0 in ENDOM bit position 1: Next frame send 1 in ENDOM bit position
1	STAOM	Start of sending a message. STAOM can only be set by software when CECEN=1. STAOM is cleared by hardware if any of these flags asserted: TEND, TU, TAERR, TERR or CECEN is cleared. If the message consists of only header frame, ENDOM should be set before configuring header frame in TDATA. After setting STAOM, the SFT counter will start and when SFT is done the Start-bit will performance on CEC line. Software can abort sending the message through clearing CECEN bit under STAOM=1. 0: No CEC transmission is on-going 1: CEC transmission is pending or executing
0	CECEN	Enable/disable HDMI-CEC controller. CECEN bit is configured by software. 0: Disable HDMI-CEC controller. Abort any sending message state and clear ENDOM/STAOM 1: Enable CEC controller and go into receiving state if STAOM=0

20.4.2. Configuration register (CEC_CFG)

Address offset: 0x04

Reset value: 0x0000 0000

Note: This register can only be write when CECEN=0.

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
LMEN	OAD [14:0]														
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							SFTOPT	BCNG	BPLEG	BREG	BRES	RTOL	SFT[2:0]		
							rw	rw	rw	rw	rw	rw	rw		

Bits	Fields	Descriptions
31	LMEN	Listen Mode Enable Bit This bit is set and cleared by software. 0: Only receive broadcast and singlecast in OAD address with appropriate ACK 1: Receive broadcast and singlecast in OAD address with appropriate ACK and receive message whose destination address is not in OAD without feedback ACK
30:16	OAD[14:0]	Own Address Each bit of OAD represents one destination address. For example: if OAD[0]=1,the controller will receive the message being sent to address 0x0. This means the controller can be configured to have multiple own addresses. Broadcast message is always received. After received destination address(last 4 bit of HEADER frame),if it is valid in the OAD, the controller will feedback with positive acknowledge ,if it is not valid in the OAD and LMEN=1,the controller will receive the message with no acknowledge, if it is not valid in the OAD and LMEN=0,the controller will not receive the message.
15:9	Reserved	Must be kept at reset value.
8	SFTOPT	The SFT start option bit This bit is set and cleared by software. 0: SFT counter starts counting when STAOM is asserted 1: SFT counter starts automatically after transmission/reception enabled
7	BCNG	Do not generate an Error-bit in broadcast message This bit is set and cleared by software. 0: In broadcast mode, BRE and BPLE will generate an Error-bit on CEC line and if LMEN=1, BPSE will also generate an Error-bit 1: Error-bit is not generated in the same condition as above
6	BPLEG	Generate an Error-bit when detected BPLE in singlecast This bit is set and cleared by software. 0: Not generate an Error-bit on CEC line when detected BPLE in singlecast

		1: Generate an Error-bit on CEC line when detected BPLE in singlecast
5	BREG	Generate an Error-bit when detected BRE in singlecast This bit is set and cleared by software. 0: Not generate an Error-bit on CEC line when detected BRE in singlecast 1: Generate an Error-bit on CEC line when detected BRE in singlecast
4	BRES	Whether stop receive message when detected BRE This bit is set and cleared by software. 0: Do not stop reception for BRE and data bit is sampled at nominal time(1.05ms) 1: Stop reception for BRE
3	RTOL	Reception bit timing tolerance This bit is set and cleared by software. 0: Standard bit timing tolerance 1: Extended bit timing tolerance
2:0	SFT[2:0]	Signal Free Time This bit is set and cleared by software. If SFT=0x0, the SFT time will perform as HDMI-CEC protocol description and if not, the SFT time is fixed configured by software. The start point is the falling edge of the ACK bit. 0x0: - 3x Standard data-bit period if SFT counter is start because of unsuccessful transmission(ARBF=1,TERR=1,TU=1 or TAERR=1) - 5x Standard data-bit period if CEC controller is the new initiator - 7x Standard data-bit period if CEC controller has successful completed transmission 0x1: 1.5x nominal data bit periods 0x2: 2.5x nominal data bit periods 0x3: 3.5x nominal data bit periods 0x4: 4.5x nominal data bit periods 0x5: 5.5x nominal data bit periods 0x6: 6.5x nominal data bit periods 0x7: 7.5x nominal data bit periods

20.4.3. Transmit data register (CEC_TDATA)

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	TDATA[7:0]
	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data register These bits are write only and contain the data byte to be transmit.

20.4.4. Receive data register (CEC_RDATA)

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								RDATA[7:0]							
								r							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data register These bits are read only and contain the last data byte which has been received from the CEC line.

20.4.5. Interrupt Flag Register (CEC_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			TAERR	TERR	TU	TEND	TBR	ARBF	RAE	BPLE	BPSE	BRE	RO	REND	BR
			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
31:13	Reserved	Must be kept at reset value.

12	TAERR	Transmit ACK Error flag. This bit is set by hardware and cleared by software writing 1. The ACK bit is received 1 in singlecast and is received 0 in broadcast will assert the flag. TAERR will stop sending message and clear STAOM and ENDOM.
11	TERR	Transmit Error This bit is set by hardware and cleared by software writing 1. TERR is asserted if the controller is in initiator state and the CEC line is low impedance but it does not pull it down. TERR will stop sending message and clear STAOM and ENDOM.
10	TU	Transmit data buffer underrun This bit is set by hardware and cleared by software writing 1. TU is asserted if the software does not write data before sending the next byte. TU will stop sending message and clear STAOM and ENDOM.
9	TEND	Transmit successfully end This bit is set by hardware and cleared by software writing 1. TEND is asserted if the all frames of the message are successfully transmitted. TEND will clear STAOM and ENDOM bit.
8	TBR	Transmit Byte data request This bit is set by hardware and cleared by software writing 1. TBR is asserted when the 4th bit of current frame is transmitted and software should write data into TDATA within 6 nominal data-bit periods
7	ARBF	Arbitration fail This bit is set by hardware and cleared by software writing 1. ARBF is asserted when either situation is occurs: external CEC device pull down the CEC line for sending start bit when controller is in SFT state or the controller and CEC device sending the start bit at the same time but the controller's initiator address priority is lower. If ARBF is asserted, the controller will get into reception state and after finish receiving the message the controller will retry to send message. During receiving and sending message, the STAOM will keep set.
6	RAE	Receive ACK Error This bit is set by hardware and cleared by software writing 1. RAE is asserted if ACK=0 in broadcast or ACK=1 in singlecast under LMEN=1 and destination address is not in OAD. RAE will stop receiving message.
5	BPLE	Bit Period Long Error This bit is set by hardware and cleared by software writing 1. BPLE is asserted when the data-bit is out of the maximum period. BPLE will stop receiving message and generate an error-bit if BPLEG=1 in singlecast or BCNG=0 in broadcast.

4	BPSE	Bit Period Short Error This bit is set by hardware and cleared by software writing 1. BPSE is asserted if a data-bit period is less than the minimal period.
3	BRE	Bit Rising Error This bit is set by hardware and cleared by software writing 1. BRE is asserted if the rising edge in a period is occurs in unexpected time.
2	RO	Receive Overrun This bit is set by hardware and cleared by software writing 1. RO is asserted when a new byte is received and BR is also set. RO will stop receiving message and send an incorrect ACK bit.
1	REND	End of Reception This bit is set by hardware and cleared by software writing 1. REND is asserted if the controller received the whole message with feedback correct ACK. REND is asserted at the same time of BR.
0	BR	Byte received This bit is set by hardware and cleared by software writing 1. BR is asserted if the controller received the whole message with feedback correct ACK. When BR is asserted, the RDATA is valid.

20.4.6. Interrupt enable register (CEC_INTEN)

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			TAERRIE	TERRIE	TUIE	TXENDIE	TBRIE	ARBFIE	RAEIE	BPLEIE	BPSEIE	BREIE	ROIE	RENDIE	BRIE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	TAERRIE	TAERR Interrupt Enable. This bit is set and cleared by software. 0: TAERR interrupt disable 1: TAERR interrupt enable
11	TERRIE	TERR Interrupt Enable. This bit is set and cleared by software.

		0: TERR interrupt disable 1: TERR interrupt enable
10	TUIE	TU Interrupt Enable. This bit is set and cleared by software. 0: TU interrupt disable 1: TU interrupt enable
9	TENDIE	TEND Interrupt Enable. This bit is set and cleared by software. 0: TEND interrupt disable 1: TEND interrupt enable
8	TBRIE	TBR Interrupt Enable. This bit is set and cleared by software. 0: TBR interrupt disable 1: TBR interrupt enable
7	ARBFIE	ARBF Interrupt Enable. This bit is set and cleared by software. 0: ARBF interrupt disable 1: ARBF interrupt enable
6	RAEIE	RAE Interrupt Enable. This bit is set and cleared by software. 0: RAE interrupt disable 1: RAE interrupt enable
5	BPLEIE	BPLE Interrupt Enable. This bit is set and cleared by software. 0: BPLE interrupt disable 1: BPLE interrupt enable
4	BPSEIE	BPSE Interrupt Enable. This bit is set and cleared by software. 0: BPSE interrupt disable 1: BPSE interrupt enable
3	BREIE	BRE Interrupt Enable. This bit is set and cleared by software. 0: BRE interrupt disable 1: BRE interrupt enable
2	ROIE	RO Interrupt Enable. This bit is set and cleared by software. 0: RO interrupt disable 1: RO interrupt enable
1	RENDIE	REND Interrupt Enable.

		This bit is set and cleared by software. 0: REND interrupt disable 1: REND interrupt enable
0	BRIE	BR Interrupt Enable. This bit is set and cleared by software. 0: BR interrupt disable 1: BR interrupt enable

21. Touch sensing interface (TSI)

21.1. Overview

Touch Sensing Interface (TSI) provides a convenient solution for touch keys, sliders and capacitive proximity sensing applications. The controller builds on charge transfer method. Placing a finger near fringing electric fields adds capacitance to the system and TSI is able to measure this capacitance change using charge transfer method.

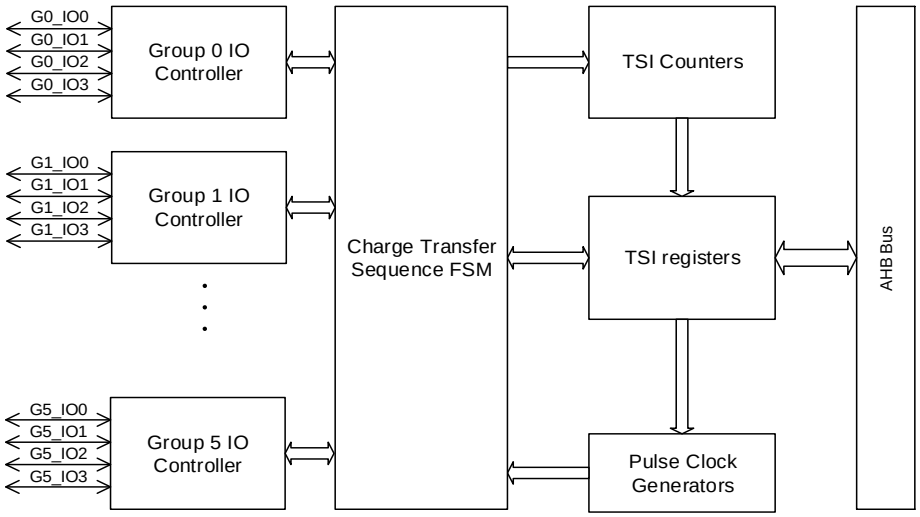
21.2. Characteristics

- Charge transfer sequence fully controlled by hardware.
- 6 fully parallel groups implemented.
- Configurable 18 IOs for capacitive sensing channel pins and 6 IOs for sample pins.
- Configurable transfer sequence frequency.
- Able to implement the user specific charge transfer sequences.
- Sequence end and error flags / configurable interrupts.
- Support spread spectrum function.

21.3. Function Overview

21.3.1. TSI block diagram

Figure 21-1. Block diagram of TSI module



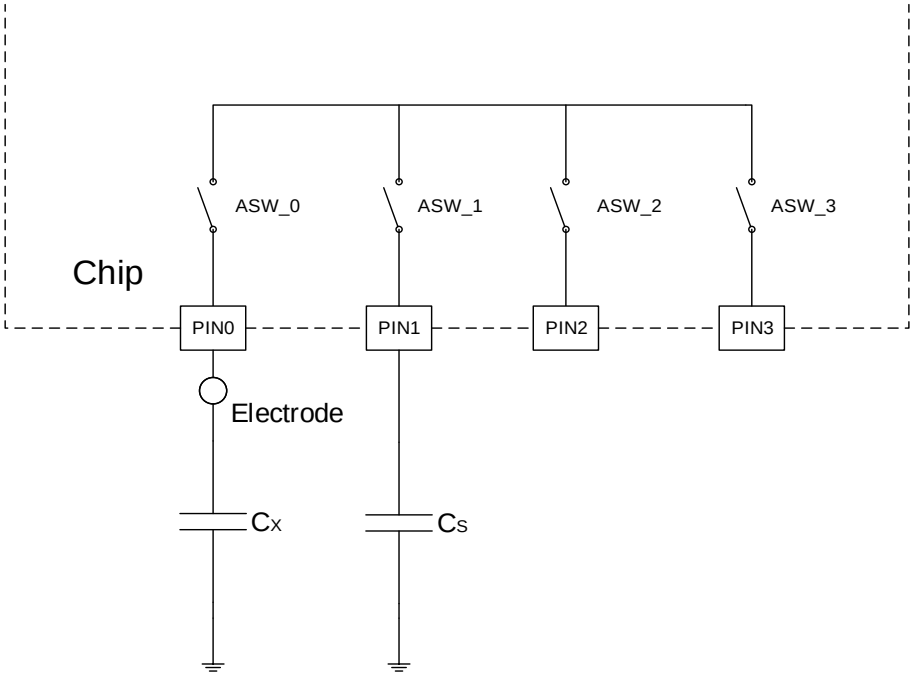
21.3.2. Touch sensing technique overview

There are different technologies for touch sensing, such as optical, resistive, capacitive, strain,

etc. Detecting the change of a system is the key problem and goal in these technologies. The TSI module is designed to use charge transfer method which detects the capacitive change of an electrode when touched or close to it by a finger. In order to detect the capacitive change, TSI performs a charge transfer sequence including several charging, transfer steps until satisfying the termination condition. The number of these steps indicates the capacitance of an electrode. So the application is able to detect the change of capacitance by monitoring the number of the charge transfer.

As shown in ***Figure 21-2. Block diagram of sample pin and channel pin***, there are 4 pins in one group and each pin has an analog switch connected to a common point which is the key component to implement charge transfer. There should be a sample pin and one or more channel pin(s) configured in one group. In ***Figure 21-2. Block diagram of sample pin and channel pin***, PIN0 is a channel pin and PIN1 is a sample pin while PIN2 and PIN3 are unused. An electrode connecting PIN0 is designed on PCB board. A sample capacitor C_s connected to sample pin PIN1 is also required. Now the capacitance of the channel pin PIN0 includes C_x and the capacitance introduced by the electrode, so capacitance of PIN0 increases when a finger is touching while the capacitance of PIN1 remains unchanged. Thus, the finger's touching can be detected if the capacitance of PIN0 can be measured. In TSI module, a charge-transfer sequence is performed to measure the capacitance of the channel pin(s) in a group, which will be detailed in next section.

Figure 21-2. Block diagram of sample pin and channel pin



21.3.3. Charge transfer sequence

In order to measure the capacitance of a channel pin, charge transfer sequence is performed in chip. The sequence shown in ***Table 21-1. Pin and analog switch state in a charge-transfer sequence*** is described based on the connection of ***Figure 21-2. Block diagram of sample pin and channel*** , i.e. PIN0 is channel pin and PIN1 is sample pin.

Table 21-1. Pin and analog switch state in a charge-transfer sequence

Step	Name	ASW_0	ASW_1	PIN0	PIN1
1	Discharge	Close	Close	Input Floating	Pull Down
2	Buffer Time1	Open	Open	Input Floating	Input Floating
3	Charge	Open	Open	Output High	Input Floating
4	Extend Charge	Open	Open	Output High	Input Floating
5	Buffer Time2	Open	Open	Input Floating	Input Floating
6	Charge Transfer	Close	Close	Input Floating	Input Floating
7	Buffer Time3	Open	Open	Input Floating	Input Floating
8	Compare	Open	Open	Input Floating	Input Floating

1. Discharge

Both C_x and C_s are discharged by closing ASW_0 and ASW_1 and configuring PIN0 to input floating and PIN1 to pull down. This step is the initial operation for a correct charge transfer sequence and is performed by software before starting a charge transfer sequence. Discharging time in this step should be guaranteed to ensure that the voltage of C_x and C_s are discharged to zero.

2. Buffer Time1

Buffer time with ASW_0 and ASW_1 open, PIN0 and PIN1 are configured to input floating.

3. Charge

Channel pin PIN0 is configured to output high, in order to charge C_x. PIN1 is configured to input floating and ASW_0 and ASW_1 remain open during this step. The charging time should be configured to ensure that the voltage of C_x is charged to V_{DD}.

4. Extend Charge

This is an optional step in a charge-transfer sequence and the behavior of all pins and analog switches in this step is the same as Step 3. The only difference between this and step 3 is the duration time, which is configurable in TSI registers. The duration of this step changes in each loop of a charge-transfer sequence, spreading the spectrum.

5. Buffer Time2

Buffer time with ASW_0 and ASW_1 open, PIN0 and PIN1 are configured to input floating.

6. Charge transfer

ASW_0 and ASW_1 are closed, PIN0 and PIN1 are configured to input floating to transfer charge from C_x to C_s. The transfer time should be configured to ensure the full transfer after that the voltage of C_x and C_s will be equal.

7. Buffer Time3

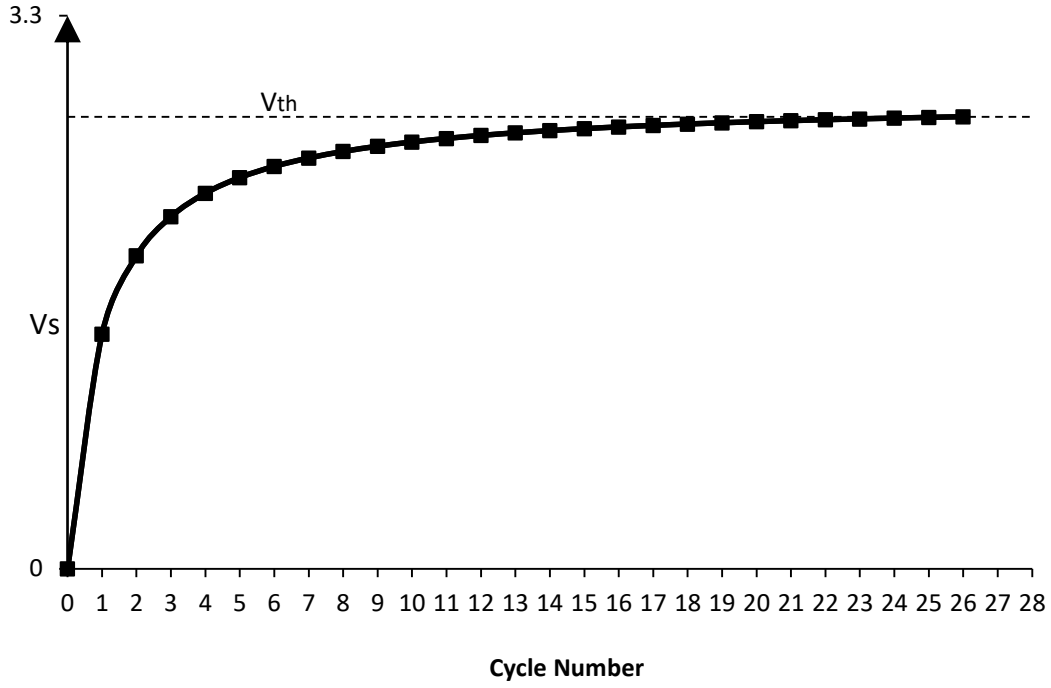
Buffer time with ASW_0 and ASW_1 open, PIN0 and PIN1 are configured to input floating.

8. Compare

ASW_0, ASW_1, PIN0 and PIN1 remain the configuration of step 7. At this step, the voltage of sample pin PIN1 is compared to a threshold called V_{th}. If voltage of PIN1 is lower than V_{th}, the sequence returns to Step 2 and continues, otherwise, the sequence ends.

The voltage of sample pin V_s is zero after initial step 1 and increases after each charge-transfer cycle, as shown in ***Figure 21-3. Voltage of a sample pin during charge-transfer sequence.*** A larger C_x will cause a greater increase during a cycle. The sequence stops when V_s reaches V_{th}. Each group has a counter which records the number of cycles performed on it to reach V_{th}. At the end of charge-transfer sequence, the group counter is read out to estimate the C_x, i.e. a smaller counter values indicates a larger C_x.

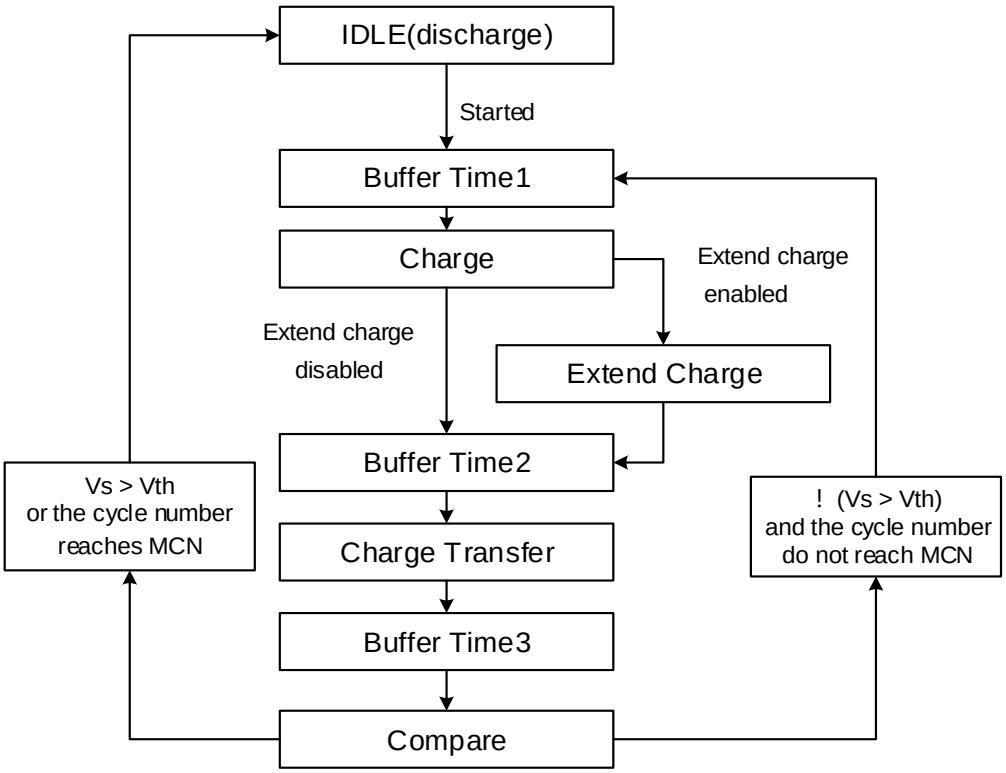
Figure 21-3. Voltage of a sample pin during charge-transfer sequence



21.3.4. Charge transfer sequence FSM

A hardware Finite-state machine (FSM) is designed in chip to perform the charge transfer sequence described in the previous section as shown in ***Figure 21-4. FSM flow of a charge-transfer sequence***.

Figure 21-4. FSM flow of a charge-transfer sequence



This FSM remains in IDLE state after reset. There are 2 kinds of start condition defined by TRGMOD bit in TSI_CTL register:

TRGMOD = 0: Software trigger mode. In this mode, the FSM starts after TSIS bit in TSI_CTL register is written 1 by software.

TRGMOD = 1: Hardware trigger mode. In this mode, the FSM starts when a falling or rising edge on TSITG pin is detected.

Once started, the FSM runs following the flow described in ***Figure 21-4. FSM flow of a charge-transfer sequence***. The FSM leaves a state if the duration time of this state reaches defined value, and goes into the next state.

The Extend Charge state is present only if the ECEN bit is set in TSI_CTL register. This state is designed to implement spread spectrum function, which will extend the duration of the pulse high state with different extend time according to current FSM cycle number. So in other word, the charge frequency becomes dynamic and not fixed. In case of noisy application environment, enabling this function can improve the robustness of TSI. At the same time, system's electromagnetic emissions will be reduced.

In comparing state, the FSM compares voltage of the sample pin in every enabled group and the threshold voltage. If all sample pins' voltage reaches the threshold, FSM returns IDLE state and stops, otherwise, it returns to Buffer Time1 state and begins the next cycle. As shown in ***Figure 21-3. Voltage of a sample pin during charge-transfer sequence***, after 27

cycles, V_s (the voltage of sample pin) reaches V_{th} (the threshold voltage).

There is also a max cycle number defined by MCN in TSI_CTL register. When the cycle number reaches MCN, FSM returns to IDLE state and stops after Compare State, whether V_s reaches V_{th} or not.

21.3.5. Clock and duration time of states

There are 3 clocks in TSI module: HCLK, CTCLK (Charge Transfer Clock) and ECCLK (Extend Charge Clock). HCLK is system clock and drives TSI's register and FSM. CTCLK is divided from HCLK with division factor defined by CTCDIV and is used for calculating the duration time of the charge state and charge transfer state. ECCLK is divided from HCLK with division factor defined by ECCDIV and is used to calculate the maximum duration time of extend charge state. ECCLK and CTCLK are independent of each other.

The duration time of each state except extend charge state is fixed in each loop according to the configuration of the register.

The duration time of Buffer Time1, Buffer Time2 and Buffer Time3 are fixed to 2 HCLK periods. The duration time of charge state and charge transfer state is defined by CDT and CTDT bits (see TSI_CTL register section for detail).

Generally, the variation ranges of extend charge frequency is limited to between 10% and 50%. The duration time of extend charge state changes in each cycle of the charge-transfer FSM, and the maximum duration time is defined by ECDT[6:0] in TSI_CTL register. If the extend charge state is enabled, the longest change time is when cycle number is ECDT+2. The duration time of Extend Charge state in each cycle is presented in ***Table 21-2. Duration time of extend charge state in each cycle***.

Table 21-2. Duration time of extend charge state in each cycle

Cycle number	Number of ECCLKs in extend charge state
1	0
2	1
…	
ECDT	ECDT-1
ECDT+1	ECDT
ECDT+2	ECDT+1
ECDT+3	ECDT
ECDT+4	ECDT-1
…	…
2*ECDT+1	2
2*ECDT+2	1
2*ECDT+3	0
2*ECDT+4	1
2*ECDT+5	2
…	…

21.3.6. PIN mode control of TSI

There are 4 pins in each group and each of these pins is able to be used as a sample pin or channel pin. Only one pin in a group should be configured as sample pin, and channel pins can be more than one. The sample pin and channel pin(s) should not be configured as the same pin in any case.

When a PIN is configured in GPIO (see chapter GPIO) used by TSI, the pin's mode is controlled by TSI. Generally, each pin has 3 modes: input, output high and output low.

The mode of a channel pin or a sample pin during a charge-transfer sequence is described in ***Table 21-1. Pin and analog switch state in a charge-transfer sequence*** which PIN0 represents a channel pin and PIN1 represents a sample pin, i.e. the charge-transfer FSM take control of these channels or sample pins' mode and the states of related analog switches when the sequence is on-going. When the sequence is in IDLE state, PINMOD bit in TSI_CTL register defines the mode of these pins. Pins that are configured in GPIO used by TSI but neither sample nor channel in TSI register is called free pins whose mode is defined by PINMOD bit in TSI_CTL, too.

21.3.7. Analog switch (ASW) and I/O hysteresis mode

A channel or sample pin's analog switch is controlled by charge-transfer sequence when FSM is running, as shown in ***Table 21-1. Pin and analog switch state in a charge-transfer sequence***. When the FSM is IDLE, these pins' analog switches are controlled by GxPy bits in TSI_ASW register. All free pin's analog switches are controlled by GxPy bits too.

TSI takes control of the analog switches when FSM is IDLE, even if these pins are not configured to be used by TSI in GPIO. The user is able to perform user-defined charge-transfer sequence by writing GxPy bits to control these analog switches, while controlling pin mode directly in GPIO.

TSI controller has the highest priority of GPIO. When TSI is enable, this configuration is available regardless of the GPIO mode, controlled by GPIO registers or other peripherals

Disable the GPIO's schmitt trigger hysteresis of TSI Pins by resetting GxPy bit in TSI_PHM register could improve the system immunity.

21.3.8. TSI operation flow

The normal operation flow of TSI is listed below:

1. System initialization, such as system clock configuration, TSI related GPIO configuration, etc.
2. Program TSI_CTL, TSI_INTEN, TSI_CHCFG, TSI_SAMPCFG and TSI_GCTL register according to demand.
3. Enable TSI by setting TSIEN bit in TSI_CTL register.

4. If configured as software trigger mode (TRGMOD = 0), charging transfer sequence starts by setting TSIS bit. If configured as hardware trigger mode (TRGMOD = 1), charging transfer sequence is started by falling / rising edge on the trigger pin.

5. Wait for the CTCF or MNERR flag in TSI_INTF register and clear these flags by setting CCTCF or CMNERR bit in TSI_INTC register.

6. Read out the CYCN bits in TSI_GxCYCN registers.

21.3.9. TSI flags and interrupts

Table 21-3. TSI errors and flags

Flag name	Description	Cleared by
CTCF	TSI stops when the sample voltage of all enabled pins group reach V_{th}.	Set CCTCF bit in TSI_INTC
MNERR	TSI stops when the cycle number reaches the maximum value.	Set CMNERR bit in TSI_INTC

21.3.10. TSI GPIOs

Table 21-4. TSI pins

TSI group	TSI pins	GPIO pins
TSI_GRP0	PIN0	PA0
	PIN1	PA1
	PIN2	PA2
	PIN3	PA3
TSI_GRP1	PIN0	PA4
	PIN1	PA5
	PIN2	PA6
	PIN3	PA7
TSI_GRP2	PIN0	PC5
	PIN1	PB0
	PIN2	PB1
	PIN3	PB2
TSI_GRP3	PIN0	PA9
	PIN1	PA10
	PIN2	PA11
	PIN3	PA12
TSI_GRP4	PIN0	PB3
	PIN1	PB4
	PIN2	PB6
	PIN3	PB7
TSI_GRP5	PIN0	PB11
	PIN1	PB12

TSI group	TSI pins	GPIO pins
	PIN2	PB13
	PIN3	PB14

21.4. Registers definition

TSI base address: 0x4002 4000

21.4.1. Control register (TSI_CTL)

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
CDT[3:0]				CTDT[3:0]				ECDT[6:0]							ECEN
rw				rw				rw							rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECDIV[0]	CTCDIV[2:0]			Reserved				MCN[2:0]			PINMOD	EGSEL	TRGMOD	TSIS	TSIEN
rw	rw							rw			rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	CDT[3:0]	Charge state duration time These bits are set and cleared by software, which controls the duration time of charge state in a charge-transfer sequence. 0000: $1\times t_{CTCLK}$ 0001: $2\times t_{CTCLK}$ 0010: $3\times t_{CTCLK}$ … 1111: $16\times t_{CTCLK}$
27:24	CTDT[3:0]	Charge transfer state duration time These bits are set and cleared by software, which controls the duration time of charge transfer state in a charge-transfer sequence. 0000: $1\times t_{CTCLK}$ 0001: $2\times t_{CTCLK}$ 0010: $3\times t_{CTCLK}$ … 1111: $16\times t_{CTCLK}$
23:17	ECDT[6:0]	Extend charge state maximum duration time These bits are set and cleared by software, which controls the maximum duration time of extend charge transfer state in a charge-transfer sequence. 0000000: $1\times t_{ECCLK}$ 0000001: $2\times t_{ECCLK}$ 0000010: $3\times t_{ECCLK}$ … 1111111: $128\times t_{ECCLK}$ **Note:** Extend charge state is only present when ECEN bit in TSI_CTL register is

		set.
16	ECEN	Extend charge state enable. 0: Extend charge disabled 1: Extend charge enabled
15	ECDIV[0]	Extend charge clock (ECCLK) division factor. ECCLK is divided from HCLK and ECDIV defines the division factor. 000: $f_{ECCLK}=f_{HCLK}$ 001: $f_{ECCLK}=f_{HCLK}/2$
14:12	CTCDIV[2:0]	Charge transfer clock(CTCLK) division factor. CTCLK is divided from HCLK and CTCDIV defines the division factor. 0000: $f_{CTCLK}=f_{HCLK}$ 0001: $f_{CTCLK}=f_{HCLK}/2$ 0010: $f_{CTCLK}=f_{HCLK}/4$ 0011: $f_{CTCLK}=f_{HCLK}/8$ … 0111: $f_{CTCLK}=f_{HCLK}/128$
11:8	Reserved	Must be kept at reset value.
7:5	MCN[2:0]	Max cycle number of a sequence These bits define the max cycle number of a charge-transfer sequence which stops after reaching this number. 000: 255 001: 511 010: 1023 011: 2047 100: 4095 101: 8191 110: 16383 111: Reserved
4	PINMOD	Pin mode This bit defines a TSI pin’s mode when charge-transfer sequence is IDLE. 0: TSI pin will output low when IDLE 1: TSI pin will keep floating-input mode when IDLE
3	EGSEL	Edge selection This bit defines the edge type in hardware trigger mode. 0: Falling edge 1: Rising edge
2	TRGMOD	Trigger mode selection 0: Software trigger mode, sequence will start after TSIS bit is set. 1: Hardware trigger mode, sequence will start after a falling / rising edge on trigger pin detected.

1	TSIS	TSI start This bit is set by software to start a charge-transfer sequence in software trigger mode and reset by hardware when the sequence stops. After setting this bit, software can reset it to stop the started sequence manually. 0: TSI is not started 1: TSI is started.
0	TSIEN	TSI enable 0: TSI module is disabled 1: TSI module is enabled

21.4.2. Interrupt enable register (TSI_INTEN)

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														MNERRIE	CTCFIE
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	MNERRIE	Max cycle number error interrupt enable 0: MNERR interrupt is disabled 1: MNERR interrupt is enabled
0	CTCFIE	Charge-transfer complete flag interrupt enable 0: CTCF interrupt is disabled 1: CTCF interrupt is enabled

21.4.3. Interrupt flag clear register (TSI_INTC)

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
Reserved														CMNERR	CCTCF
														w	w

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CMNERR	Clear max cycle number error 0: Reserved 1: Clear MNERR
0	CCTCF	Clear charge-transfer complete flag 0: Reserved 1: Clear CTCF

21.4.4. Interrupt flag register (TSI_INTF)

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
Reserved														MNERR	CTCF
														r	r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	MNERR	Max cycle number error flag This bit is set by hardware after charge-transfer sequence stops because it reaches the max cycle number defined by MCN[2:0]. This bit is cleared by writing 1 to CMNERR bit in TSI_INTC register. 0: No max count error 1: Max count error
0	CTCF	Charge-transfer complete flag This bit is set by hardware after charge-transfer sequence stops because all enabled group's sample pins reach the threshold voltage or because the cycle number reaches the value defined by MCN[2:0]. This bit is cleared by writing 1 to CCTCF bit in TSI_INTC register. 0: Charge-transfer not complete 1: Charge-transfer complete

21.4.5. Pin hysteresis mode register (TSI_PHM)

Address offset: 0x10

Reset value: 0xFFFF FFFF

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								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	GxPy	Pin hysteresis mode This bit is set and cleared by software. 0: Pin GxPy Schmitt trigger hysteresis mode disabled 1: Pin GxPy Schmitt trigger hysteresis mode enabled

21.4.6. Analog switch register (TSI_ASW)

Address offset: 0x18

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								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	GxPy	Analog switch state. This bit is set and cleared by software. 0: Analog switch of GxPy is open 1: Analog switch of GxPy is closed

21.4.7. Sample configuration register (TSI_SAMPCFG)

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								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	GxPy	Sample pin mode This bit is set and cleared by software. 0: Pin GxPy is not a sample pin 1: Pin GxPy is a sample pin

21.4.8. Channel configuration register (TSI_CHCFG)

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								G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	GxPy	Channel pin mode This bit is set and cleared by software. 0: Pin GxPy is not a channel pin 1: Pin GxPy is a channel pin

21.4.9. Group control register (TSI_GCTL)

Address offset: 0x30

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										GC5	GC4	GC3	GC2	GC1	GC0
										r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										GE5	GE4	GE3	GE2	GE1	GE0
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
21:16	GCx	Group complete This bit is set by hardware when charge-transfer sequence for an enabled group is complete. It is cleared by hardware when a new charge-transfer sequence starts. 0: Charge-transfer for group x is not complete 1: Charge-transfer for group x is complete
15:6	Reserved	Must be kept at reset value.
5:0	GEx	Group enable This bit is set and cleared by software. 0: Group x is disabled 1: Group x is enabled

21.4.10. Group x cycle number registers (TSI_GxCYCN) (x = 0…5)

Address offset: 0x34 + 0x04 * x

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		CYCN[13:0]													
		rw													

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	CYCN[13:0]	Cycle number

These bits reflect the cycle number for a group as soon as a charge-transfer sequence completes. They are cleared by hardware when a new charge-transfer sequence starts.

22. Universal Serial Bus full-speed device interface (USBD)

The USBD is only available on GD32F150 series.

22.1. Overview

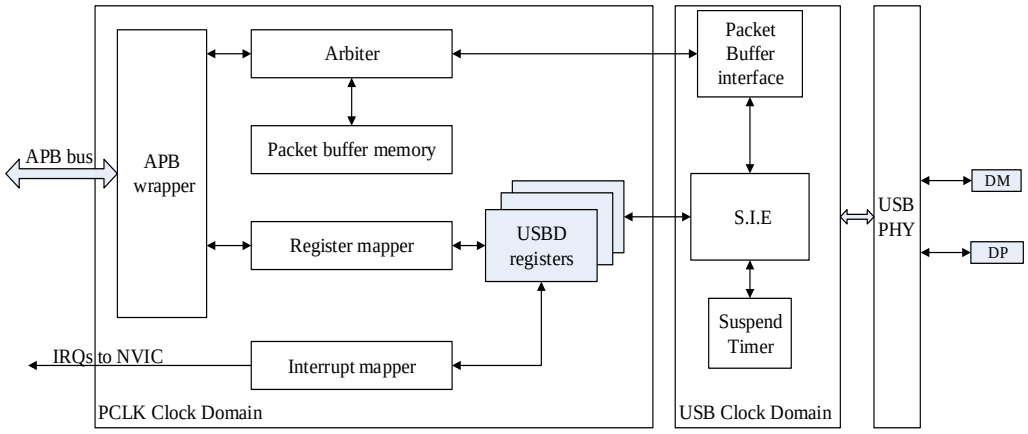
The Universal Serial Bus full-speed device interface (USBD) module provides a device solution for implementing a USB 2.0 full-speed compliant peripheral. It contains a full-speed internal USB PHY and no more external PHY chip is needed. USBD supports all the four types of transfer (control, bulk, interrupt and isochronous) defined in USB 2.0 protocol.

22.2. Main features

- USB 2.0 full-speed device controller.
- Support up to 8 configurable bidirectional endpoints.
- Support double-buffered bulk/isochronous endpoints.
- Each endpoint supports control, bulk, isochronous or interrupt transfer types (exclude endpoint 0, endpoint 0 only support control transfer).
- Support USB suspend/resume operations.
- Dedicated 512-byte SRAM used for data packet buffer.
- Integrated USB PHY.

22.3. Block diagram

Figure 22-1. USBD block diagram



22.4. Signal description

Table 22-1. USBD signal description

I/O port	Type	Description
DM	Input/Output	Differential D-
DP	Input/Output	Differential D+

Note: As soon as the USBD is enabled, these pins are connected to the USBD internal transceiver automatically.

22.5. Clock configuration

According to the USB standard definition, the USB full-speed module adopt fixed 48MHz clock. It is necessary to configure two clock for using USBD, one is the USB controller clock, its frequency must be configured to 48MHz, and the other one is the APB1 to USB interface clock which is also APB1 bus clock, its frequency can be above or below 48MHz.

Note: In order to meet the system requirements of packet buffer interface and USB data transfer rate, the frequency of the APB1 bus clock must be greater than 24MHz, so as to avoid data buffer overflow and underflow.

48MHz clock of USB controller can be generated by dividing MCU internal or external crystal oscillator by a programmable prescaler, then multiplicating the frequency through PLL.

- Regard two frequency division of 8MHz internal oscillator as the input of the PLL, then 12 frequencies doubling the clock.
- Regard 8MHz external oscillator as the input of the PLL, firstly frequency doubling, then adopt USB frequency divider to divide frequency.

When the USB clock is generated by external crystal, only 4 USB frequency prescaler can be used as 1, 1.5, 2 and 2.5. Thus, for obtaining 48MHz clock, PLL frequencies doubling could be configured as 48MHz, 72MHz.

Note: Regardless of using internal or external crystal oscillator to generate USB clock, the clock accuracy must reach ±500ppm. If the accuracy of the USB clock cannot meet the condition, data transfer may not conform to the requirements of the USB specification, and even it may cause USB not working directly.

22.6. Function overview

22.6.1. USB endpoints

USBD supports 8 USB endpoints that can be individually configured.

Each endpoint supports:

- Single/Double buffer (endpoint 0 can't use double buffer).
- One endpoint buffer descriptor.
- Programmable buffer starting address and buffer length.
- Configurable response to a packet.
- Control transfer (endpoint 0 only).

Endpoint buffer

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus. To efficiently manage USB endpoint communications, USBD implements a dedicated data packet buffer of 512-bytes SRAM memory accessed directly by the USB peripheral. It is mapped to the APB1 peripheral memory, from 0x4000 6000 to 0x4000 6400. The total capacity is 1KB, but USBD uses actually only 512 bytes for the bus width reason.

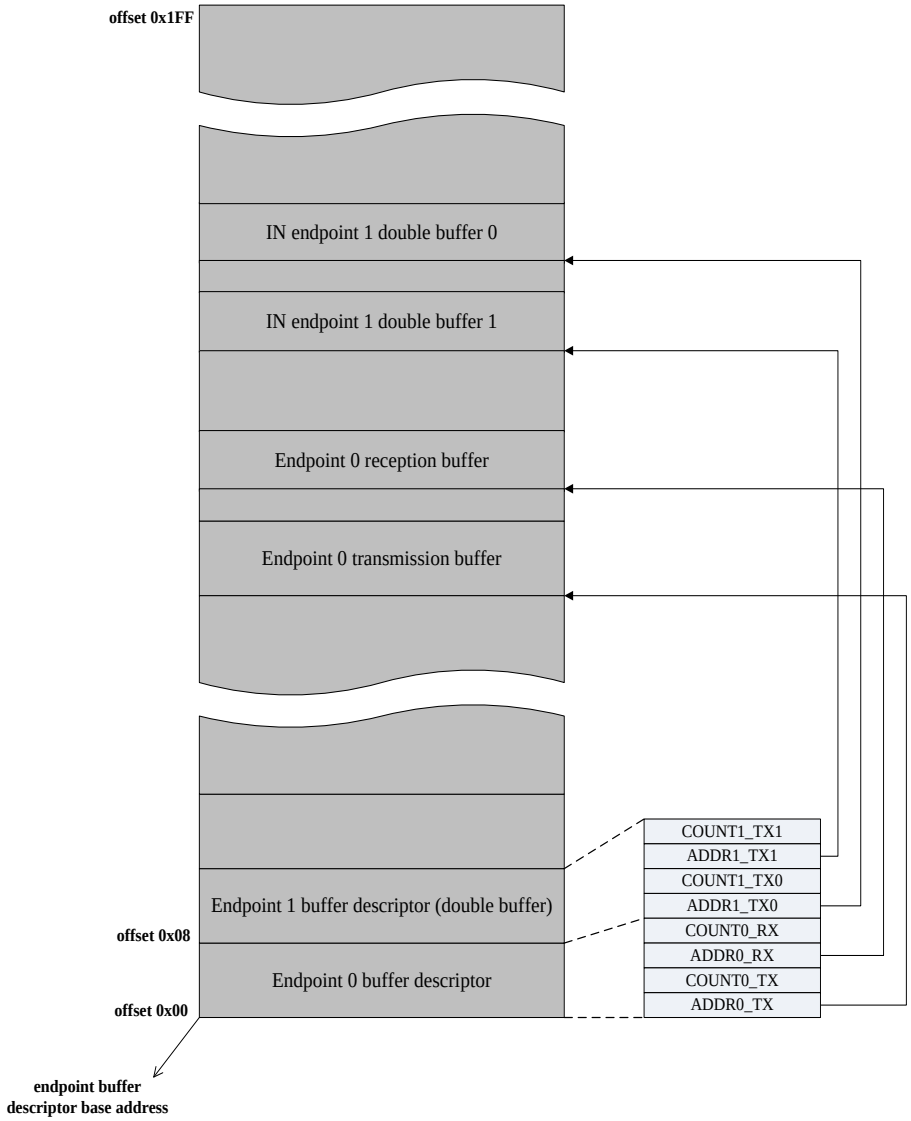
Each endpoint can be associated with one or two data packet buffers used to store the current data payload. The bidirectional endpoint has usually two buffers, one is used for transmission and the other one is for reception. The mono-directional endpoint only has one buffer for data operation.

Endpoint buffer descriptor table

USBD implements an endpoint buffer descriptor table which defines the buffer address and length and which is also located in the endpoint data packet buffer. The endpoint buffer descriptor is used as a communication port between the application firmware and the SIE in system memory. Every endpoint direction requires two 16-bit words buffer descriptor. Therefore, each table entry includes 4 16-bit words (Tx and Rx two direction) and is aligned to 8-byte boundary. When an endpoint is double-buffered, the SIE will use the two buffers in ping-pong operation mode. The endpoint buffer descriptor table is pointed to by the USBD endpoint buffer address register.

The relationship between endpoint buffer descriptor table entries and packet buffer areas is depicted in ***Figure 22-2. An example with buffer descriptor table usage (USBD_BADDR = 0).***

Figure 22-2. An example with buffer descriptor table usage (USBD_BADDR = 0)



Note: This figure is not drawn on the actual scale, and it is addressed through the USB bus 16-bit mode.

Double-buffered endpoints

The double-buffered feature is used to improve bulk transfer performance. To implement the new flow control scheme, the USB peripheral should know which packet buffer is currently in use by the application software, so to be aware of any conflict. Since in the USBD_EPxCS register, there are two data toggle bits (TX_DTG and RX_DTG) but only one is used by USBD for hardware data handling (due to the unidirectional constraint required by double-buffering feature), the other one can be used by the application software to show which buffer it is currently using. This new buffer flag is called software buffer bit (SW_BUF). In ***Table 22-2. Double-buffering buffer flag definition***, the correspondence between USBD_EPxCS register bits and DTG/SW_BUF definition is explained.

Table 22-2. Double-buffering buffer flag definition

Buffer flag	Tx endpoint	Rx endpoint
DTG	TX_DTG (USBD_EPxCS bit 6)	RX_DTG (USBD_EPxCS bit 14)
SW_BUF	RX_DTG (USBD_EPxCS bit 14)	TX_DTG (USBD_EPxCS bit 6)

The DTG bit and the SW_BUF bit are responsible for the flow control. When a transfer completes, the USB peripheral toggle the DTG bit; when the data have been copied, the application software need to toggle the SW_BUF bit. Except for the first time, if the value of DTG bit is equal to the SW_BUF's, the transfer will pause, and the host is NAK. When the two bits are not equal, the transfer resume.

Table 22-3. Double buffer usage

Endpoint Type	DTG	SW_BUF	Packet buffer used by the USB peripheral	Packet buffer used by the application software
OUT	0	1	EPxTBADDR/EPxTBCNT buffer description table locations.	EPxRBADDR/EPxRBCNT buffer description table locations.
	1	0	EPxRBADDR/EPxRBCNT buffer description table locations.	EPxTBADDR/EPxTBCNT buffer description table locations.
IN	0	1	EPxTBADDR/EPxTBCNT buffer description table locations.	EPxRBADDR/EPxRBCNT buffer description table locations.
	1	0	EPxRBADDR/EPxRBCNT buffer description table locations.	EPxTBADDR/EPxTBCNT buffer description table locations.

Endpoint memory requests arbitration

As the USBD is connected to the APB1 bus through an APB1 interface, so USB APB1 interface will accept memory requests coming from the APB1 bus and from the USB interface. The arbiter will resolve the conflicts by giving priority to APB1 accesses, while always reserving half of the memory bandwidth to complete all USB transfers. This time-duplex scheme implements a virtual dual-port SRAM that allows memory access, when an USB transaction is happening. Multiword APB1 transfers of any length are also allowed by this scheme.

22.6.2. Operation procedure

USB transaction process

After the endpoint is configured and a transaction is required, the hardware will detect the token packet. When a token is recognized by the USBD, the data transfer is performed. When all the data has been transferred, the proper handshake packet over the USBD is generated or expected according to the direction of the transfer.

After the transaction process is completed, an endpoint-specific interrupt is generated. In the interrupt routine, the application can process it accordingly.

Transaction formatting is performed by the hardware, including CRC generation and checking.

Once the endpoint is enabled, endpoint control and status register, buffer address and COUNT filed should not be modified by the application software. When the data transfer operation is completed, notified by a STIF interrupt event, they can be accessed again to re-enable a new operation.

IN transaction

When a configured and valid endpoint receives an IN token packet, it will send the data packet to the host. If the endpoint is not valid, a NAK or STALL handshake is sent according to the endpoint status.

In the data packet transfer process, a configured data PID will be sent firstly, then the actual data in endpoint buffer memory is loaded into the output shift register to be transmitted. After the data are sent, the computed CRC will be sent by hardware.

When receiving the ACK sent from the host, then the USB peripheral will toggle the data PID and set the endpoint status to be NAK. At the same time, the successful transfer interrupt will be triggered. In the interrupt service routine, application fill the data packet memory with data, start next transfer by re-enable the endpoint by setting the endpoint status VALID.

OUT and SETUP transaction

USBD handles these two tokens more or less in the same way, the differences in the handling of SETUP packets will be detailed in the following section about control transfer.

After the received endpoint is configured and enabled, host will send OUT/SETUP token to the device. When receiving the token, USBD will access the endpoint buffer descriptor to initialize the endpoint buffer address and length. Then the received data bytes subsequently are packed in words (LSB mode) and transferred to the endpoint buffer. When detecting the end of data packet, the computed CRC and received CRC are compared. If no errors occur, an ACK handshake packet is sent to the host.

When the transaction is completed correctly, USBD will toggle the data PID and set the endpoint status to be NAK. Then the endpoint successful transfer interrupt will be triggered by hardware. In the interrupt service routine, the application can get the transaction type and read the received data from the endpoint buffer. After the received data is processed, the application should initiate further transactions by setting the endpoint status valid.

If any error happens during reception, the USBD set the error interrupt bit and still copy data into the packet memory buffer, but will not send the ACK packet. The USBD itself can recover from reception errors and continue to handle next transfer. The USBD never override outside the data buffer, which is controlled by the internal register configured. The received 2-byte CRC is also copied to the packet memory buffer, immediately following data bytes. If the length of data is greater than actually allocated length, the excess data are not copied. This

is a buffer overrun situation. A STALL handshake is sent, and this transaction fails.

If an addressed endpoint is not valid, a NAK or STALL handshake packet is sent instead of the ACK, according to the endpoint status and no data is written to the endpoint data buffers.

Control transfers

Control transfers require that a SETUP transaction be started from the host to a device to describe the type of control access that the device should perform. The SETUP transaction is followed by zero or more control DATA transactions that carry the specific information for the requested access. Finally, a STATUS transaction completes the control transfer and allows the endpoint to return the status of the control transfer to the client software. After the STATUS transaction for a control transfer is completed, the host can advance to the next control transfer for the endpoint.

USBD always use endpoint 0 in two directions as default control endpoint to handle control transfers. It is aware of the number and direction of data stages by interpreting the contents of SETUP transaction, and is required to set the unused direction endpoint 0 status to STALL except the last data stage.

At the last data stage, the application software set the opposite direction endpoint 0 status to NAK. This will keep the host waiting for the completion of the control operation. If the operation completes successfully, the software will change NAK to VALID, otherwise to STALL. If the status stage is an OUT, the STATUS_OUT bit should be set, so that a status transaction with non-zero data will be answered STALL to indicate an error happen.

According to USB specification, device isn't allowed to abort current command and then start new command, so that device must answer a SETUP packet with an ACK handshake packet, not with a NAK or STALL handshake packet.

When the configured control endpoint 0 receives a SETUP token, the USBD accepts the data, performing the required data transfers and sends back an ACK handshake. If there is unsuccessfully handling data transfer about previously issued request, the USB discard SETUP token and regard current condition as error, and then urge the host to send the request token again.

Isochronous transfers

Isochronous transfers can guarantee constant data rate and bounded latency, but do not support data retransmission in response to errors on the bus. Consequently, the isochronous transaction does not have a handshake phase, and have no ACK packet after the data packet. Data toggling is not supported, and DATA0 PID is only used to start a data packet.

The isochronous endpoint status only can be set DISABLED and VALID, any other value is illegal. The application software can implement double-buffering to improve performance. By swapping transmission and reception data packet buffer on each transaction, the application software can copy the data into or out of a buffer, at the same time the USB peripheral handle the data transmission or reception of data in another buffer. The DTOG bit indicates which

buffer that the USB peripheral is currently using.

The application software initializes the DTOG according to the first buffer to be used. At the end of each transaction, the RX_ST or TX_ST bit is set, depending on the enabled direction regardless of CRC errors or buffer-overrun conditions (if errors occur, the ERRIF bit will be set). At the same time, The USB peripheral will toggle the DTOG bit, but will not affect the STAT bit.

22.6.3. USB events and interrupts

Each USB action is always initiated by the application software, driven by one USB interrupt or event. After system reset, the application needs to wait for a succession of USB interrupts and events.

Reset events

System and power-on reset

Upon system and power-on reset, the application software should first provide all required clock to the USB module and interface, then de-assert its reset signal so to be able to access its registers, last switch on the analog part of the device related to the USB transceiver.

The USB firmware should do as follows:

- Reset CLOSE bit in USBD_CTL register.
- Wait for the internal reference voltage to be stable.
- Clear SETRST bit in USBD_CTL register.
- Clear the USBD_INTF register to remove the spurious pending interrupt and then enable other unit.

USB reset (RESET interrupt)

When this event occurs, the USB peripheral status is the same as the moment system reset.

The USB firmware should do as follows:

- Set USBEN bit in USBD_DADDR register to enable USB module in 10ms.
- Initialize the USBD_EP0CS register and its related packet buffers.

Suspend and resume events

The USB module can be forced to place in low-power mode (SUSPEND mode) by writing in the USB control register (USBD_CTL) whenever required. At this time, all static power consumption is avoided and the USB clock can be slowed down or stopped. It will be resumed when detect activity at the USB bus while in low-power mode.

The USB protocol insists on power management by the USB device. This becomes even more important if the device draws power from the bus (bus-powered device). The following constraints should be met by the bus-powered device.

- A device in the non-configured state should draw a maximum of 100mA from the USB

bus.

- A configured device can draw only up to what is specified in the Max Power field of the configuration descriptor. The maximum value is 500mA.
- A suspended device should draw a maximum of 500uA.

A device will go into the suspend state if there is no activity on the USB bus for more than 3ms. A suspended device wakes up, if RESUME signaling is detected.

USBD also supports software initiated remote wakeup. To initiate remote wakeup, the application software must enable all clocks and clear the suspend bit after MCU is waked up. This will cause the hardware to generate a remote wakeup signal upstream.

Setting the SETSPS bit to 1 enables the suspend mode, and it will disable the check of SOF reception. Setting the LOWM bit to 1 will shut down the static power consumption in the analog USB transceivers, but the RESUME signal is still able to be detected.

USB Interrupts

USBD has three interrupts: low-priority interrupt, high-priority interrupt and wakeup interrupt. Software can configure these interrupts to route the interrupt condition to these entries in the NVIC table. An interrupt will be generated when both the interrupt status bit and the corresponding interrupt enable bit are set. The interrupt status bit is set by hardware if the interrupt condition occurs (irrespective of the interrupt enable bit).

- Low-priority interrupt (Channel 37): triggered by all USB events.
- High-priority interrupt (Channel 38): triggered only by a correct transfer event for isochronous and double-buffer bulk transfer.
- Wakeup interrupt (Channel 42): triggered by the wakeup events.

22.6.4. Operation guide

This section describes the operation guide for USBD.

USBD register initialization sequence

1. Clear the CLOSE bit in USBD_CTL register, then clear the SETRST bit.
2. Clear USBD_INTF register to remove any spurious pending interrupt.
3. Program USBD_BADDR register to set endpoint buffer base address.
4. Set USBD_CTL register to enable interrupts.
5. Wait for the reset interrupt (RSTIF).
6. In the reset interrupt, initialize default control endpoint 0 to start enumeration process and program USBD_BADDR to set the device address to 0 and enable USB module function.
7. Configure endpoint 0 and prepare to receive SETUP packet.

Endpoint initialization sequence

1. Program USBD_EPxTBADDR or USBD_EPxRBADDR registers with transmission or reception data buffer address.
2. Program the EP_CTL and EP_KCTL bits in USBD_EPxCS register to set endpoint type and buffer kind according to the endpoint usage.
3. If the endpoint is a single buffer endpoint:
   1) Initialize the endpoint data toggle bit by programming the TX_DTG or RX_DTG bit in USBD_EPxCS register, but endpoint 0 needs to set them to 1 and 0 respectively for control transfer.
   2) Configure endpoint status by programming the TX_STA bit or RX_STA bit in USBD_EPxCS register, but both of them are set to ‘10 (NAK) if use endpoint 0 to initialize the control transfer.

   If the endpoint is a double buffer endpoint:
   1) Both transmission and reception toggle fields need to be programmed. If the endpoint is a Tx endpoint, clear the TX_DTG and RX_DTG bit in USBD_EPxCS register, or if endpoint is a Rx endpoint, it needs to toggle TX_DTG bit.
   2) Program USBD_EPxTBCNT and USBD_EPxRBCNT register to set transfer data bit count.
   3) Endpoint transmission and reception status both need to be configured. If the endpoint is a Tx endpoint, set the TX_STA bit to be NAK and RX_STA bit to be DISABLED, or the endpoint is a Rx endpoint, set the RX_STA bit to be VALID and TX_STA bit to be DISABLED.

SETUP and OUT data transfers

1. Program USBD_EPxRBCNT register to set BLKSIZ and EPRCNT filed, these filed defines the endpoint buffer length.
2. Configure the endpoint status to be VALID to enable the endpoint to receive data by programming USBD_EPxCS register.
3. Wait for successful transfer interrupt (STIF).
4. In the interrupt handler, application can get the transaction type by reading the STEUP bit in USBD_EPxCS register. Then application will read the data payload from the endpoint data buffer with the start address defined in USBD_EPxRBAR register. Last application will interpret the data and process the corresponding transaction.

IN data transfers

1. Program USBD_EPxTBCNT register to set EPTXCNT filed, this filed defines the endpoint buffer length.
2. Configure the endpoint status to be VALID to enable the endpoint to transmit data by programming USBD_EPxCS register.

3. Wait for successful transfer interrupt (STIF).
4. In the interrupt handler, application needs to update user buffer length and location pointer. Then application fill the endpoint buffer with user buffer data. Last application will configure the endpoint status to be VALID to start next transfer.

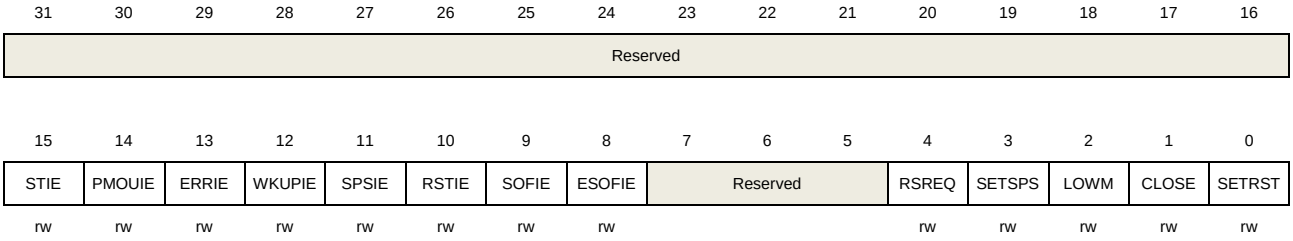
22.7. Register definition

USBD base address: 0x4000 5C00

22.7.1. USBD control register (USBD_CTL)

Address offset: 0x40
Reset value: 0x0003

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	STIE	Successful transfer interrupt enable. 0: Successful transfer interrupt disabled. 1: Interrupt generated when STIF bit in USBD_INTF register is set.
14	PMOUIE	Packet memory overrun/underrun interrupt enable. 0: No interrupt generated when packet memory overrun / underrun. 1: Interrupt generated when PMOUIF bit in USBD_INTF register is set.
13	ERRIE	Error interrupt enable. 0: Error interrupt disabled 1: Interrupt generated when ERRIF bit in USBD_INTF register is set.
12	WKUPIE	Wakeup interrupt enable 0: Wakeup interrupt disabled 1: Interrupt generated when WKUPIF bit in USBD_INTF register is set.
11	SPSIE	Suspend state interrupt enable 0: Suspend state interrupt disabled 1: Interrupt generated when SPSIF bit in USB_IFR register is set.
10	RSTIE	USB reset interrupt enable. 0: USB reset interrupt disabled 1: Interrupt generated when RSTIF bit in USBD_INTF register is set.
9	SOFIE	Start of frame interrupt enable 0: Start of frame interrupt disabled

		1: Interrupt generated when SOFIF bit in USBD_INTF register is set.
8	ESOFIE	Expected start of frame interrupt enable 0: Expected start of frame interrupt disabled 1: Interrupt generated when ESOFIF bit in USBD_INTF register is set.
7:5	Reserved	Must be kept at reset value
4	RSREQ	Resume request The software set a resume request to the USB host, and the USB host should drive the resume sequence according the USB specifications 0: No resume request 1: Send resume request.
3	SETSPS	Set suspend The software should set suspend state when SPSIF bit in USBD_INTF register is set. 0: Not set suspend state. 1: Set suspend state.
2	LOWM	Low-power mode When set this bit, the USB goes to low-power mode at suspend state. If resume from suspend state, the hardware reset this bit. 0: No effect 1: Go to low-power mode at suspend state.
1	CLOSE	Close state When this bit is set, the USBD goes to close state, and completely close the USBD and disconnected from the host. 0: Not in close state 1: In close state.
0	SETRST	Set reset When this bit is set, the USBD peripheral should be reset. 0: No reset 1: A reset generated.

22.7.2. USBD interrupt flag register (USBD_INTF)

Address offset: 0x44
Reset value: 0x0000

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
STIF	PMOUIF	ERRIF	WKUPIF	SPSIF	RSTIF	SOFIF	ESOFIF	Reserved			DIR	EPNUM[3:0]			

r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	STIF	Successful transfer interrupt flag This bit set by hardware when a successful transaction completes
14	PMOUIF	Packet memory overrun/underrun interrupt flag This bit set by hardware to indicate that the packet memory is inadequate to hold transfer data. The software writes 0 to clear this bit.
13	ERRIF	Error interrupt flag This bit set by hardware when an error happens during transaction. The software writes 0 to clear this bit.
12	WKUPIF	Wakeup interrupt flag This bit set by hardware in the SUSPEND state to indicate that activity is detected. The software writes 0 to clear this bit.
11	SPSIF	Suspend state interrupt flag When no traffic happen in 3ms, hardware set this bit to indicate a SUSPEND request. The software writes 0 to clear this bit.
10	RSTIF	USB reset interrupt flag Set by hardware when the USB RESET signal is detected. The software writes 0 to clear this bit.
9	SOFIF	Start of frame interrupt flag Set by hardware when a new SOF packet arrives, The software writes 0 to clear this bit.
8	ESOFIF	Expected start of frame interrupt flag Set by the hardware to indicate that a SOF packet is expected but not received. The software writes 0 to clear this bit.
7:5	Reserved	Must be kept at reset value
4	DIR	Direction of transaction Set by the hardware to indicate the direction of the transaction 0: IN type 1: OUT type
3:0	EPNUM[3:0]	Endpoint Number Set by the hardware to identify the endpoint which the transaction is directed to

22.7.3. USBD status register (USBD_STAT)

Address offset: 0x48

Reset value: 0x0XXX where X is undefined

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
RX_DP	RX_DM	LOCK	SOFLN[1:0]		FCNT[10:0]										
r	r	r	r		r										

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	RX_DP	Receive data + line status Represent the status on the DP line
14	RX_DM	Receive data - line status Represent the status on the DM line
13	LOCK	Locked the USB Set by the hardware indicate that at the least two consecutive SOF have been received
12:11	SOFLN[1:0]	SOF lost number Increment every ESOFIF happens by hardware Cleared once the reception of SOF
10:0	FCNT[10:0]	Frame number counter The Frame number counter incremented every SOF received.

22.7.4. USBD device address register (USBD_DADDR)

Address offset: 0x4C
Reset value: 0x0000

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								USBEN	USBDAR[6:0]						
								rw	rw						

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	USBEN	USB device enable Set by software to enable the USB device 0: The USB device disabled. No transactions handled.

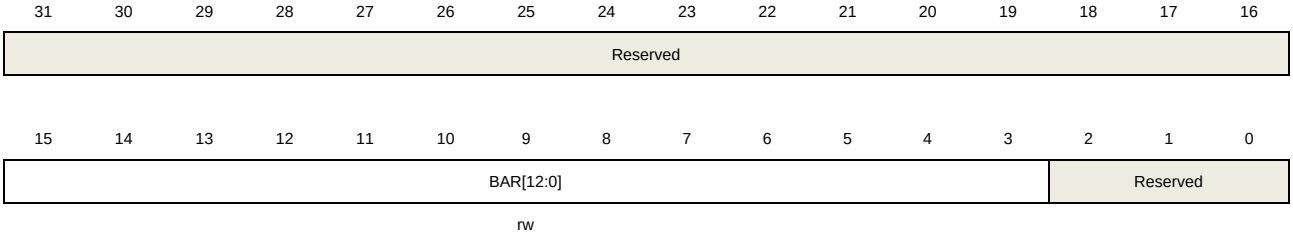
1: The USB device enabled.

6:0	USBDAR[6:0]	USBD device address After bus reset, the address is reset to 0x00. If the enable bit is set, the device will respond on packets for function address DEV_ADDR

22.7.5. USBD buffer address register (USBD_BADDR)

Address offset: 0x50
Reset value: 0x0000

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

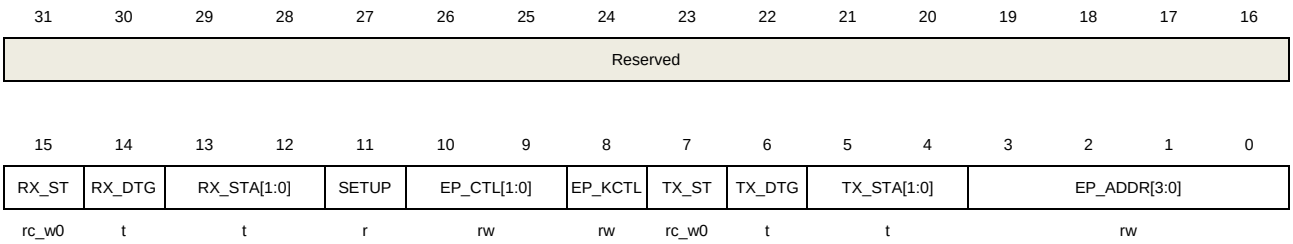


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:3	BAR[12:0]	Buffer address Start address of the allocation buffer(512byte on-chip SRAM), used for buffer descriptor table, packet memory
2:0	Reserved	Must be kept at reset value

22.7.6. USBD endpoint x control and status register (USBD_EPxCS), x=[0..7]

Address offset: 0x00 to 0x1C
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	RX_ST	Reception successful transferred Set by hardware when a successful OUT/SETUP transaction complete Cleared by software by writing 0
14	RX_DTG	Reception data PID toggle This bit represent the toggle data bit (0=DATA0,1=DATA1)for non-isochronous endpoint Used to implement the flow control for double-buffered endpoint Used to swap buffer for isochronous endpoint
13:12	RX_STA[1:0]	Reception status bits Toggle by writing 1 by software Remain unchanged by writing 0 Refer to the table below
11	SETUP	Setup transaction completed Set by hardware when a SETUP transaction completed.
10:9	EP_CTL[1:0]	Endpoint type control Refer to the table below
8	EP_KCTL	Endpoint kind control The exact meaning depends on the endpoint type Refer to the table below
7	TX_ST	Transmission successful transfer Set by hardware when a successful IN transaction complete Clear by software
6	TX_DTG	Transmission data PID toggle This bit represent the toggle data bit (0=DATA0,1=DATA1)for non-isochronous endpoint Used to implement the flow control for double-buffered endpoint Used to swap buffer for isochronous endpoint
5:4	TX_STA[1:0]	Status bits, for transmission transfers Refer to the table below
3:0	EP_ADDR	Endpoint address Used to direct the transaction to the target endpoint

Table 22-4. Reception status encoding

RX_STA[1:0]	Meaning
00	**DISABLED:** ignore all reception requests of this endpoint
01	**STALL:** STALL handshake status
10	**NAK**: NAK handshake status
11	**VALID**: enable endpoint for reception

Table 22-5. Endpoint type encoding

EP_CTL[1:0]	Meaning
00	**BULK:** bulk endpoint

EP_CTL[1:0]	Meaning
01	**CONTROL:** control endpoint
10	**ISO**: isochronous endpoint
11	**INTERRUPT**: interrupt endpoint

Table 22-6. Endpoint kind meaning

EP_CTL[1:0]		EP_KCTL Meaning
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT

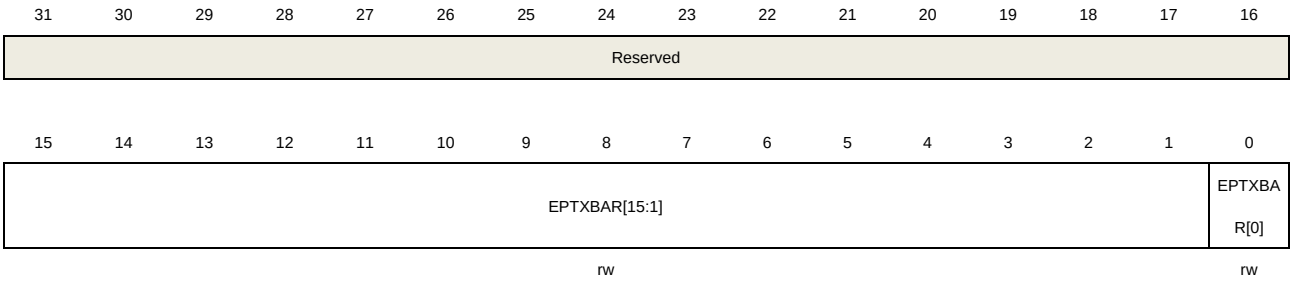
Table 22-7. Transmission status encoding

TX_STA[1:0]	Meaning
00	**DISABLED:** ignore all transmission requests of this endpoint
01	**STALL:** STALL handshake status
10	**NAK**: NAK handshake status
11	**VALID**: enable endpoint for transmission

22.7.7. USBD endpoint x transmission buffer address register (USBD_EPxTBADDR), x=[0..7]

Address offset: [USBD_BADDR] + x * 16
USB local address: [USBD_BADDR] + x * 8

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



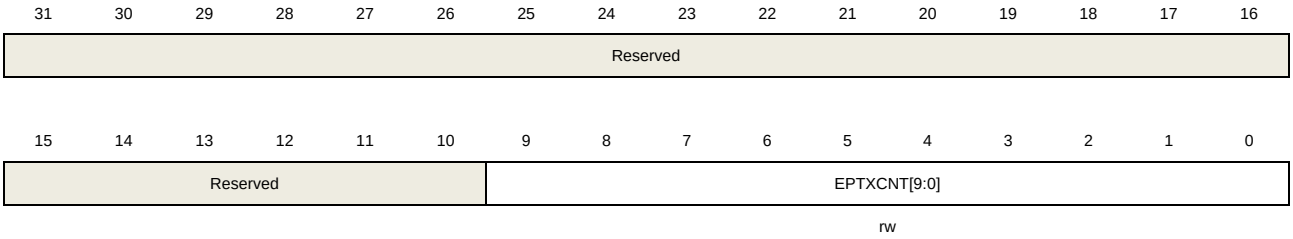
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:1	EPTXBAR[15:1]	Endpoint transmission buffer address Start address of the packet buffer containing data to be sent when receive next IN token
0	EPTXBAR[0]	Must be set to 0

22.7.8. USBD endpoint x transmission buffer byte count register (USBD_EPxTBCNT), x=[0..7]

Address offset: [USBD_BADDR] + x * 16 + 4

USB local Address: [USBD_BADDR] + x * 8 + 2

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

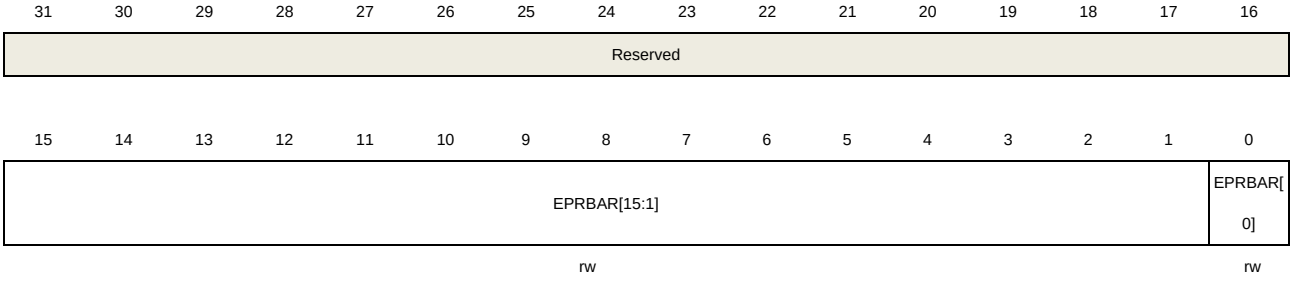


Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9:0	EPTXCNT[9:0]	Endpoint transmission byte count The number of bytes to be transmitted at next IN token

22.7.9. USBD endpoint x reception buffer address register (USBD_EPxRBADDR), x=[0..7]

Address offset: [USBD_BADDR] + x * 16 + 8
USB local Address: [USB_BADDR] + x * 8 + 4

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

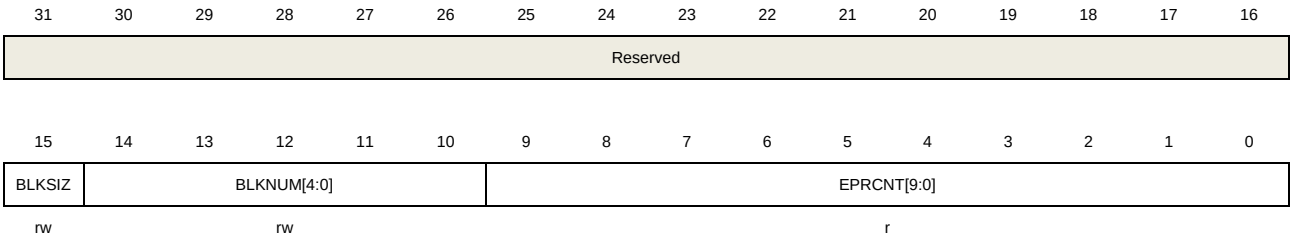


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:1	EPRBAR[15:1]	Endpoint reception buffer address Start address of packet buffer containing the data received by the endpoint at the next OUT/SETUP token
0	EPRBAR[0]	Must be set to 0

22.7.10. USBD endpoint x reception buffer byte count register (USBD_EPxRBCNT), x=[0..7]

Address offset: [USBD_BADDR] + x * 16 + 12
USB local Address: [USBD_BADDR] + x * 8 + 6

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	BLKSIZ	Block size 0: block size is 2 bytes 1: block size is 32 bytes
14:10	BLKNUM[4:0]	Block number The number of blocks allocated to the packet buffer
9:0	EPRCNT[9:0]	Endpoint reception byte count The number of bytes to be received at next OUT/SETUP token

23. Appendix

23.1. List of abbreviations used in register

Table 23-1. 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.

23.2. List of terms

Table 23-2. 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.

23.3. Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

24. Revision history

Table 24-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Mar.18, 2014
2.0	Add GD32F170/190 Products	Jan.15, 2016
3.0	Adapt To New Name Convention	Jun.24, 2016
3.0.1	Proofreading	Mar.30, 2017
3.1.0	Proofreading	Jan.22, 2018
3.2	1. Modify the format according to the version specification. 2. Modify the register median domain name and clock tree of RCU module. 3. Modify the OPA register offset address. 4. Modify CMP register offset address. 5. Modify the protection status option byte SPC and its complement byte value to 0xa55a, and modify the maximum safety protection status value to 0x33cc. 6. Add the base address of usbd register. 7. Optimize the I2S clock generation structure diagram. 8. Modify the description of ewif in wdgt module. 9. Update the SPI / I2S module description about the four wire host mode.	Nov.28, 2019
3.3	1. Modify the description of dir bit of USBD_INTF register. 2. Modify figure18-6 to 18-8 in Chapter I2C.	Mar.13, 2020
3.4	1. Modify the format of the baud rate formula in the CAN chapter. 2. In chapter 13.1.3 of the WDGT module, add notes about entering deepsleep/standby mode immediately after feeding the WDGT. 3. In chapter 10.4.3 of the ADC module, add a description about the delay after the ADC starts.	Jul.1, 2020
3.5	1. Delete the description of GD32F130 and GD32F170 in ***Function overview*** of the CMP chapter. 2. Modify the parameters in ***Table 13 1. Min / max FWDGT timeout period at 40 kHz (IRC40K)*** of WDGT chapter. 3. Modify the range of HXTAL in ***Figure 4 2. Clock tree of GD32F130xx and GD32F150xx devices*** of the RCU chapter to 3~25MHz.	Dec.15, 2021
3.6	1. Update ***Power management unit (PMU)*** chapter. 2. Update ***Serial peripheral interface/Inter-IC sound (SPI/I2S)*** chapter. 3. Update ***Inter-integrated circuit interface (I2C)*** chapter. 4. Update ***Direct memory access controller (DMA)*** chapter. 5. Update ***Comparator (CMP)*** chapter. 6. Update ***General-purpose I/Os (GPIO)*** chapter. 7. Update ***Reset and clock unit (RCU)*** chapter. 8. Update ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter. 9. Update ***Analog to digital converter (ADC)*** chapter.	Jul.25, 2022

Revision No.	Description	Date
	10. Delete the SLCD, CAN, IVREF and OPA chapters.	
3.7	1. Update ***Direct memory access controller (DMA)*** chapter. 2. Update ***Power management unit (PMU)*** chapter. 3. Update ***Comparator (CMP)*** chapter. 4. Update ***Reset and clock unit (RCU)*** chapter. 5. Update ***Flash memory controller (FMC)*** chapter. 6. Update ***Debug (DBG)*** chapter. 7. Update ***Universal Serial Bus full-speed device interface (USBD)*** chapter. 8. Update ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter. 9. Update ***Interrupt / event controller (EXTI)*** chapter.I 10. Update ***Touch Sensing Interface (TSI)*** chapter. 11. Update ***Serial peripheral interface/Inter-IC sound (SPI/I2S)*** chapter.	Jun.15, 2023
3.8	1. Update ***General-purpose I/Os (GPIO)*** chapter. 2. Update ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter. 3. Update ***Comparator (CMP)*** chapter. 4. Update ***Digital-to-analog converter (DAC)*** chapter. 5. Update ***TIMER (TIMERx)*** chapter.	Jan.5, 2024
3.9	1. Add more descriptions of the CH0IF bit field in TIMERx_INTF register in chapter ***TIMER (TIMERx)***. 2. Correct the spelling error of the word "Philips" in chapter ***Serial peripheral interface/Inter-IC sound (SPI/I2S)***. 3. Modify the description of reading the general data register and deleted the electrical characteristics in chapter ***Analog to digital converter (ADC)***. 4. Update 2'b10(downcount only) to 2'b01(downcount only) in ***Figure 15-8. and Figure15-38 Center-aligned counter timechart*** of chapter ***TIMER (TIMERx)*** . 5. Update the ***Table 23-3. Double buffer usage*** in chapter ***Universal Serial Bus full-speed device interface (USBD)*** 22.6.1. 6. Remove the chart content about VBUS in chapter ***Universal Serial Bus full-speed device interface (USBD)*** 22.3 and 22.4. 7. Deleted the sleeping signal from figure 3-1 in chapter ***Power management unit (PMU)***	Feb.20, 2025
4.0	1. Modify the description of the TX pin configuration in Section 17.3.11 of ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter for half-duplex communication mode: Change the TX pin configuration to open-drain mode. 2. Update the description of the TSIEN bit field in the TSI_CTL0 register in ***Touch Sensing Interface (TSI)*** chapter. 3. Modify the SMBUS timeout value in the ***Inter-integrated circuit interface (I2C)*** chapter, changing it from 35ms to 25ms.	Aug.08, 2025

Revision No.	Description	Date
	4. Add a note in Section 19.4.4 of the ***Serial peripheral interface/Inter-IC sound (SPI/I2S)*** chapter: The configuration value of the I2S serial clock must be set to less than 1/6 of the PCLK clock (excluding 1/6).	
4.1	1. Added a CRC usage Note in section 19.3.7 of the ***Serial peripheral interface/Inter-IC sound (SPI/I2S)*** chapter. 2. Change 1.2V to V_{CORE}, VDD pin to VDD, and added a description of V_{CORE} in section 3.3.3 in chapter ***Power management unit (PMU)*** chapter. 3. Remove "start bit" from the noise error condition caused by receive sampling in section 17.3.4 of ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter. 4. Add a description "When configuring PSC[7:0], the baud rate register also needs to be configured as 16*PSC[7:0]." for low-power division configuration in section 17.3.10 of ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter. 5. Add the note "When oversampling by 16, RT[23:0] cannot be configured as 0xFFFFFF." for USART_RT bit filed in section 17.4.6 of ***Universal synchronous asynchronous receiver transmitter (USART)*** chapter. 6. Add the note "There must be an interval of 7 or more IRC40K clock cycles between two consecutive watchdog reload operations." in section 13.1.3 of ***Watchdog timer (WDGT)*** chapter. 7. Add the description for the RUD bit: "This bit is reset by hardware after the update operation of FWDGT_RLD register." in section 13.1.4 of ***Watchdog timer (WDGT)*** chapter. 8. Add the description for the PUD bit: "This bit is reset by hardware after the update operation of FWDGT_PSC register." in section 13.1.4 of ***Watchdog timer (WDGT)*** chapter.	Jan.31, 2026

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, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) 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.

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.

© 2026 GigaDevice Semiconductor Inc. – All rights reserved