GigaDevice Semiconductor Inc.

GD32F50x
Arm® Cortex®-M33 32-bit MCU

For GD32F505xx, GD32F503xx, GD32F502xx

User Manual

Revision 1.3

(May. 2026)

Table of Contents

Table of Contents **2**

List of Figures **18**

List of Tables **26**

1. System and memory architecture **29**

1.1. Arm® Cortex®-M33 processor **29**

1.2. System architecture **30**

1.3. Memory map **33**

1.3.1. On-chip SRAM memory 37

1.3.2. On-chip flash memory overview 38

1.4. Boot configuration **38**

1.5. System configuration registers **40**

1.5.1. Configuration register 0 (SYSCFG_CFG0) 40

1.5.2. Configuration register 1 (SYSCFG_CFG1) 40

1.5.3. Lockup control register (SYSCFG_LKCTL) 41

1.5.4. Bus timeout register (SYSCFG_BUSTO) 41

1.5.5. Timer input selection register (SYSCFG_TIMERCISEL) 42

1.5.6. FPU interrupt enable register (SYSCFG_FPUINTEN) 43

1.5.7. SRAM write protection register (SYSCFG_SRAMWP) 44

1.5.8. SRAM ECC status register (SYSCFG_SRAMECCSTAT) 44

1.5.9. SRAM ECC control and status register (SYSCFG_SRAMECCCS) 45

1.5.10. Bus timeout status register (SYSCFG_BUSTOSTAT) 46

1.5.11. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 7) 46

1.5.12. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 7) 48

1.5.13. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 7) 49

1.5.14. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=1, 2, 3, 4) 50

1.5.15. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=1, 2, 3, 4) 51

1.5.16. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=1, 2, 3, 4) 53

1.5.17. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=15, 16) 54

1.5.18. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=15, 16) 55

1.5.19. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=15, 16) 56

1.6. Device electronic signature **57**

1.6.1. Memory density information 58

1.6.2. Unique device ID (96 bits) 58

2. Interrupt / event controller (EXTI) **60**

2.1. Overview **60**

2.2. Characteristics **60**

2.3. Interrupts function overview 60
2.4. External interrupt and event (EXTI) block diagram 64
2.5. External Interrupt and Event function overview 64
2.6. EXTI Register 67
2.6.1. Interrupt enable register (EXTI_INTEN) 67
2.6.2. Event enable register (EXTI_EVEN) 67
2.6.3. Rising edge trigger enable register (EXTI_RTEN) 68
2.6.4. Falling edge trigger enable register (EXTI_FTEN) 68
2.6.5. Software interrupt event register (EXTI_SWIEV) 68
2.6.6. Pending register (EXTI_PD) 69
3. Direct memory access controller (DMA) 70
3.1. Overview 70
3.2. Characteristics 70
3.3. Block diagram 71
3.4. Function overview 71
3.4.1. DMA operation 71
3.4.2. Peripheral handshake 73
3.4.3. Arbitration 73
3.4.4. Address generation 74
3.4.5. Circular mode 74
3.4.6. Memory to memory mode 74
3.4.7. Channel configuration 74
3.4.8. Interrupt 75
3.4.9. DMA request mapping 75
3.5. Register definition 77
3.5.1. Interrupt flag register (DMA_INTF) 77
3.5.2. Interrupt flag clear register (DMA_INTC) 78
3.5.3. Channel x control register (DMA_CHxCTL) 78
3.5.4. Channel x counter register (DMA_CHxCNT) 80
3.5.5. Channel x peripheral base address register (DMA_CHxPADDR) 81
3.5.6. Channel x memory base address register (DMA_CHxMADDR) 81
4. DMA request multiplexer (DMAMUX) 83
4.1. Overview 83
4.2. Characteristics 83
4.3. Block diagram 84
4.4. Function overview 84
4.4.1. DMAMUX signals 85
4.4.2. DMAMUX request multiplexer 85
4.4.3. DMAMUX request generator 87

4.4.4. Channel configurations 88
4.4.5. Interrupt 89
4.4.6. DMAMUX mapping 89
4.5. Register definition 94
4.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG) 94
4.5.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF) 95
4.5.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC) 95
4.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG) 96
4.5.5. Request generator interrupt flag register (DMAMUX_RG_INTF) 97
4.5.6. Request generator interrupt flag clear register (DMAMUX_RG_INTC) 97
5. Flash memory controller (FMC) 99
5.1. Introduction 99
5.2. Characteristics 99
5.3. Function overview 99
5.3.1. Flash memory architecture 99
5.3.2. Read operations 100
5.3.3. Unlock the FMC_CTLx/FMC_OBCTLx registers 100
5.3.4. Page erase 101
5.3.5. Mass erase 102
5.3.6. Main flash programming 103
5.3.7. Option bytes modify 105
5.3.8. Option bytes description 106
5.3.9. Page erase/program protection 107
5.3.10. OTP block programming 107
5.3.11. Security protection 110
5.3.12. Frequency Control 110
5.4. FMC registers 112
5.4.1. Unlock key register 0(FMC_KEY0) 112
5.4.2. Option byte unlock key register (FMC_OBKEY) 112
5.4.3. Status register 0 (FMC_STAT0) 112
5.4.4. Control register 0(FMC_CTL0) 113
5.4.5. Address register 0 (FMC_ADDR0) 115
5.4.6. Option byte control register 0 (FMC_OBCTL0) 115
5.4.7. Option byte control register 1 (FMC_OBCTL1) 116
5.4.8. Option byte control register 2 (FMC_OBCTL2) 117
5.4.9. OTP1 configuration register (FMC_OTP1CFG) 117
5.4.10. Option bytes status register (FMC_OBSTAT) 118
5.4.11. Unlock key register 1(FMC_KEY1) 118
5.4.12. Status register 1 (FMC_STAT1) 118
5.4.13. Control register 1(FMC_CTL1) 119
5.4.14. Address register 1 (FMC_ADDR1) 120
5.4.15. OTP3 status register (FMC_OTP3_STAT) 121

5.4.16. Product ID register (FMC_PID) 122
6. Power management unit (PMU) 123
6.1. Overview 123
6.2. Characteristics 123
6.3. Function overview 123
6.3.1. Backup domain 124
6.3.2. VDD / VDDA power domain 125
6.3.3. V_{CORE} power domain 127
6.3.4. Power saving modes 129
6.4. PMU registers 132
6.4.1. Control register0 (PMU_CTL0) 132
6.4.2. Control and status register (PMU_CS) 134
6.4.3. Control register 1(PMU_CTL1) 136
7. Backup registers (BKP) 138
7.1. Introduction 138
7.2. Main features 138
7.3. Function description 138
7.3.1. RTC clock calibration 138
7.3.2. Tamper detection 138
7.4. BKP registers 140
7.4.1. Backup data register x (BKP_DATAx) (x= 0..41) 140
7.4.2. RTC signal output control register (BKP_OCTL) 140
7.4.3. Tamper pin control register (BKP_TPCTL) 141
7.4.4. Tamper control and status register (BKP_TPCS) 142
8. Reset and clock unit (RCU) 144
8.1. Reset control unit (RCTL) 144
8.1.1. Overview 144
8.1.2. Function overview 144
8.2. Clock control unit (CCTL) 145
8.2.1. Overview 145
8.2.2. Characteristics 149
8.2.3. Function overview 149
8.3. Register definition 154
8.3.1. Control register (RCU_CTL) 154
8.3.2. Clock configuration register 0 (RCU_CFG0) 156
8.3.3. Clock interrupt register (RCU_INT) 158
8.3.4. APB2 reset register (RCU_APB2RST) 162
8.3.5. APB1 reset register (RCU_APB1RST) 164
8.3.6. AHB enable register (RCU_AHBEN) 166

8.3.7. APB2 enable register (RCU_APB2EN) 168
8.3.8. APB1 enable register (RCU_APB1EN) 171
8.3.9. Backup domain control register (RCU_BDCTL) 173
8.3.10. Reset source/clock register (RCU_RSTSCK) 175
8.3.11. AHB reset register (RCU_AHBRST) 176
8.3.12. Clock configuration register 1 (RCU_CFG1) 178
8.3.13. PLL bandwidth configuration register (RCU_PLLBWCFG) 180
8.3.14. Deep-sleep mode voltage register (RCU_DSV) 180
8.3.15. Clock frequency monitor configuration register 0 (RCU_CKFMCFG0) 181
8.3.16. Clock frequency monitor configuration register 1 (RCU_CKFMCFG1) 182
8.3.17. Clock frequency monitor configuration register 2 (RCU_CKFMCFG2) 183
8.3.18. Clock frequency monitor configuration register 3 (RCU_CKFMCFG3) 184
8.3.19. Additional clock control register (RCU_ADDCTL) 185
8.3.20. Additional clock interrupt register (RCU_ADDINT) 187
8.3.21. APB1 additional reset register (RCU_ADDAPB1RST) 187
8.3.22. APB1 additional enable register (RCU_ADDAPB1EN) 188
8.3.23. LOCK register (RCU_LOCK) 188

9. Clock trim controller (CTC) 190

9.1. Overview 190

9.2. Characteristics 190

9.3. Function overview 190

9.3.1. REF sync pulse generator 191
9.3.2. CTC trim counter 191
9.3.3. Frequency evaluation and automatically trim process 192
9.3.4. Software program guide 193

9.4. Register definition 195

9.4.1. Control register 0 (CTC_CTL0) 195
9.4.2. Control register 1 (CTC_CTL1) 196
9.4.3. Status register (CTC_STAT) 197
9.4.4. Interrupt clear register (CTC_INTC) 199

10. General-purpose and alternate-function I/Os (GPIO and AFIO) 201

10.1. Overview 201

10.2. Characteristics 201

10.3. Function overview 201

10.3.1. GPIO pin configuration 202
10.3.2. External interrupt/event lines 203
10.3.3. Alternate functions (AF) 204
10.3.4. Additional functions 204
10.3.5. Input configuration 204
10.3.6. Output configuration 204
10.3.7. Analog configuration 205

10.3.8. Alternate function (AF) configuration 205
10.3.9. GPIO locking function 206
10.4. Register definition 207
10.4.1. Port control register (GPIOx_CTL, x = A...E) 207
10.4.2. Port output mode register (GPIOx_OMODE, x = A...E) 208
10.4.3. Port output speed register (GPIOx_OSPD, x = A...E) 210
10.4.4. Port pull-up / pull-down register (GPIOx_PUD, x = A...E) 212
10.4.5. Port input status register (GPIOx_ISTAT, x = A...E) 213
10.4.6. Port output control register (GPIOx_OCTL, x = A...E) 214
10.4.7. Port bit operate register (GPIOx_BOP, x = A...E) 214
10.4.8. Port configuration lock register (GPIOx_LOCK, x = A...E) 215
10.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x = A...E) 216
10.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x = A...E) 217
10.4.11. Bit clear register (GPIOx_BC, x = A...E) 218
10.4.12. EXTI sources selection register 0 (AFIO_EXTISS0) 218
10.4.13. EXTI sources selection register 1 (AFIO_EXTISS1) 219
10.4.14. EXTI sources selection register 2 (AFIO_EXTISS2) 220
10.4.15. EXTI sources selection register 3 (AFIO_EXTISS3) 221
11. Trigger selection controller (TRIGSEL) 223
11.1. Overview 223
11.2. Characteristics 223
11.3. Function overview 223
11.4. Internal connect 224
11.5. Register definition 230
11.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0) 230
11.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1) 230
11.5.3. Trigger selection for EXTOUT register 2 (TRIGSEL_EXTOUT_2) 231
11.5.4. Trigger selection for EXTOUT register 3 (TRIGSEL_EXTOUT_3) 232
11.5.5. Trigger selection for TIMER0_ITI register (TRIGSEL_TIMER0ETI) 232
11.5.6. Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ITI) 233
11.5.7. Trigger selection for TIMER2_ITI register (TRIGSEL_TIMER2ITI) 234
11.5.8. Trigger selection for TIMER3_ITI register (TRIGSEL_TIMER3ITI) 234
11.5.9. Trigger selection for TIMER4_ITI register (TRIGSEL_TIMER4ITI) 235
11.5.10. Trigger selection for TIMER7_ITI register (TRIGSEL_TIMER7ITI) 235
11.5.11. Trigger selection for TIMER15_ITI register (TRIGSEL_TIMER15ITI) 236
11.5.12. Trigger selection for TIMER16_ITI register (TRIGSEL_TIMER16ITI) 236
11.5.13. Trigger selection for DAC register (TRIGSEL_DAC) 237
11.5.14. Trigger selection for ADC0_ROUTRG register (TRIGSEL_ADC0_ROUTRG) 238
11.5.15. Trigger selection for ADC0_INSTRG register (TRIGSEL_ADC0_INSTRG) 238
11.5.16. Trigger selection for ADC1_ROUTRG register (TRIGSEL_ADC1_ROUTRG) 239
11.5.17. Trigger selection for ADC1_INSTRG register (TRIGSEL_ADC1_INSTRG) 240
11.5.18. Trigger selection for ADC2_ROUTRG register (TRIGSEL_ADC2_ROUTRG) 240

11.5.19. Trigger selection for ADC2_INSTRG register (TRIGSEL_ADC2_INSTRG) 241
11.5.20. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN) 241
11.5.21. Trigger selection for TIMER0_CHBRKIN register (TRIGSEL_TIMER0CHBRKIN) 242
11.5.22. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN) 243
11.5.23. Trigger selection for TIMER7_CHBRKIN register (TRIGSEL_TIMER7CHBRKIN) 243
11.5.24. Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN) 244
11.5.25. Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN) 245

12. TIMER (TIMERx) 246

12.1. Advanced timer (TIMERx, x=0, 7) 247
12.1.1. Overview 247
12.1.2. Characteristics 247
12.1.3. Block diagram 248
12.1.4. Function overview 248
12.1.5. Registers definition (TIMERx, x=0, 7) 289

12.2. General level0 timer (TIMERx, x=1,2,3,4) 333
12.2.1. Overview 333
12.2.2. Characteristics 333
12.2.3. Block diagram 333
12.2.4. Function overview 334
12.2.5. Registers definition (TIMERx, x=1,2,3,4) 354

12.3. General level3 timer (TIMERx, x=15,16) 379
12.3.1. Overview 379
12.3.2. Characteristics 379
12.3.3. Block diagram 379
12.3.4. Function overview 380
12.3.5. Register definition (TIMERx, x=15,16) 407

12.4. Basic timer (TIMERx, x=5,6) 441
12.4.1. Overview 441
12.4.2. Characteristics 441
12.4.3. Block diagram 441
12.4.4. Function overview 441
12.4.5. Registers definition (TIMERx, x=5,6) 445

13. Real-time Clock(RTC) 450

13.1. Overview 450

13.2. Characteristics 450

13.3. Function overview 450
13.3.1. RTC reset 451
13.3.2. RTC reading 451
13.3.3. RTC configuration 452
13.3.4. RTC flag assertion 452

13.4. RTC Register 454
13.4.1. RTC interrupt enable register(RTC_INTEN) 454
13.4.2. RTC control register(RTC_CTL) 454
13.4.3. RTC prescaler high register (RTC_PSCH) 455
13.4.4. RTC prescaler low register(RTC_PSCL) 456
13.4.5. RTC divider high register (RTC_DIVH) 456
13.4.6. RTC divider low register (RTC_DIVL) 456
13.4.7. RTC counter high register(RTC_CNTH) 457
13.4.8. RTC counter low register (RTC_CNTL) 457
13.4.9. RTC alarm high register(RTC_ALRMH) 458
13.4.10. RTC alarm low register (RTC_ALRML) 458
14. Watchdog timer (WDGT) 459
14.1. Free watchdog timer (FWDGT) 459
14.1.1. Overview 459
14.1.2. Characteristics 459
14.1.3. Function overview 459
14.1.4. Register definition 462
14.2. Window watchdog timer (WWDGT) 465
14.2.1. Overview 465
14.2.2. Characteristics 465
14.2.3. Function overview 465
14.2.4. Register definition 468
15. Debug (DBG) 470
15.1. Introduction 470
15.2. JTAG/SW function description 470
15.2.1. Switch JTAG or SW interface 470
15.2.2. Pin assignment 470
15.2.3. JTAG daisy chained structure 471
15.2.4. Debug reset 471
15.2.5. JEDEC-106 ID code 471
15.3. Debug hold function description 471
15.3.1. Debug support for power saving mode 471
15.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT and CAN 472
15.4. DBG registers 473
15.4.1. ID code register (DBG_ID) 473
15.4.2. Control register (DBG_CTL) 473
16. Cyclic redundancy checks management unit (CRC) 477
16.1. Overview 477
16.2. Characteristics 477
16.3. Function overview 478

16.4. Register definition 479
16.4.1. Data register (CRC_DATA) 479
16.4.2. Free data register (CRC_FDATA) 479
16.4.3. Control register (CRC_CTL) 480
16.4.4. Initialization data register (CRC_IDATA) 480
16.4.5. Polynomial register (CRC_POLY) 481
17. True random number generator (TRNG) 482
17.1. Overview 482
17.2. Characteristics 482
17.3. Function overview 482
17.3.1. Operation flow 483
17.3.2. Error flags 483
17.4. Register definition 484
17.4.1. Control register (TRNG_CTL) 484
17.4.2. Status register (TRNG_STAT) 484
17.4.3. Data register (TRNG_DATA) 485
18. Cryptographic Acceleration Unit (CAU) 487
18.1. Overview 487
18.2. Characteristics 487
18.3. CAU data type 487
18.4. Cryptographic acceleration processor 489
18.4.1. AES cryptographic acceleration processor 489
18.5. Operating modes 491
18.6. CAU DMA interface 492
18.7. CAU interrupts 492
18.8. Register definition 494
18.8.1. Control register (CAU_CTL) 494
18.8.2. Status register 0 (CAU_STAT0) 495
18.8.3. Data input register (CAU_DI) 496
18.8.4. Data output register (CAU_DO) 496
18.8.5. DMA enable register (CAU_DMAEN) 497
18.8.6. Interrupt enable register (CAU_INTEN) 497
18.8.7. Status register 1 (CAU_STAT1) 498
18.8.8. Interrupt flag register (CAU_INTF) 498
18.8.9. Key registers (CAU_KEY0..3(H / L)) 499
19. Hash Acceleration Unit (HAU) 502
19.1. Overview 502
19.2. Characteristics 502

19.3. HAU data type 502
19.4. HAU core 504
19.4.1. Automatic data padding 504
19.4.2. Digest computing 505
19.4.3. Hash mode 506
19.5. HAU interrupt 506
19.5.1. Input FIFO interrupt 506
19.5.2. Calculation completion interrupt 506
19.6. Register definition 507
19.6.1. Control register (HAU_CTL) 507
19.6.2. Data input register (HAU_DI) 508
19.6.3. Configuration register (HAU_CFG) 508
19.6.4. Data output register (HAU_DO0..7) 509
19.6.5. Interrupt enable register (HAU_INTEN) 512
19.6.6. Status and flag register (HAU_STAT) 512
20. Analog-to-digital converter (ADC) 514
20.1. Overview 514
20.2. Characteristics 514
20.3. Pins and internal signals 515
20.4. Function overview 516
20.4.1. ADC clock 516
20.4.2. ADC enable 516
20.4.3. Routine and inserted sequence 516
20.4.4. Operation modes 517
20.4.5. Inserted sequence management 521
20.4.6. Convert data latch 522
20.4.7. Conversion result threshold monitor 523
20.4.8. Data storage mode 524
20.4.9. Sample time configuration 524
20.4.10. External trigger configuration 525
20.4.11. DMA request 525
20.4.12. ADC internal channels 525
20.4.13. Programmable resolution (DRES) 526
20.4.14. On-chip hardware oversampling 526
20.5. ADC sync mode 528
20.5.1. Free mode 530
20.5.2. Routine parallel mode 530
20.5.3. Inserted parallel mode 530
20.5.4. Routine follow-up fast mode 531
20.5.5. Routine follow-up slow mode 532
20.5.6. Inserted trigger rotation mode 533

20.5.7. Combined routine parallel & inserted parallel mode 534
20.5.8. Combined routine parallel & inserted trigger rotation mode 534
20.5.9. Combined inserted parallel & routine follow-up mode 535
20.6. ADC interrupts 535
20.7. Register definition 537
20.7.1. Status register (ADC_STAT) 537
20.7.2. Control register 0 (ADC_CTL0) 538
20.7.3. Control register 1 (ADC_CTL1) 540
20.7.4. Sample time register 0 (ADC_SAMPT0) 541
20.7.5. Sample time register 1 (ADC_SAMPT1) 542
20.7.6. Inserted channel data offset register x (ADC_IOFFx) (x = 0..3) 543
20.7.7. Watchdog 0 high threshold register (ADC_WD0HT) 544
20.7.8. Watchdog 0 low threshold register (ADC_WD0LT) 544
20.7.9. Routine sequence register 0 (ADC_RSQ0) 544
20.7.10. Routine sequence register 1 (ADC_RSQ1) 545
20.7.11. Routine sequence register 2 (ADC_RSQ2) 545
20.7.12. Inserted sequence register (ADC_ISQ) 546
20.7.13. Latch data register x (ADC_LDATAx) (x= 0..3) 547
20.7.14. Routine data register (ADC_RDATA) 547
20.7.15. Inserted data register (ADC_IDATA) 548
20.7.16. Latch data control register(ADC_LDCTL) 548
20.7.17. Oversample control register (ADC_OVSAMPCTL) 549
21. Digital-to-analog converter (DAC) 552
21.1. Overview 552
21.2. Characteristics 552
21.3. Function overview 553
21.3.1. DAC enable 553
21.3.2. DAC output buffer 554
21.3.3. DAC data configuration 554
21.3.4. DAC trigger 554
21.3.5. DAC conversion 554
21.3.6. DAC noise wave 555
21.3.7. DAC output voltage 556
21.3.8. DMA request 556
21.4. Register definition 557
21.4.1. DACx control register 0 (DAC_CTL0) 557
21.4.2. DACx software trigger register (DAC_SWT) 558
21.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH) 559
21.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH) 559
21.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH) 560
21.4.6. DACx_OUT0 data output register (DAC_OUT0_DO) 560
21.4.7. DACx_OUT0 status register 0 (DAC_STAT0) 561

22. Comparator (CMP) 562
22.1. Overview 562
22.2. Characteristic 562
22.3. Function overview 562
22.3.1. CMP clock and reset 563
22.3.2. CMP I/O configuration 563
22.3.3. CMP hysteresis 564
22.3.4. CMP register write protection 564
22.3.5. CMP output blanking 565
22.3.6. CMP digital filter 565
22.3.7. CMP voltage scaler function 566
22.3.8. CMP interrupt 566
22.4. Register definition 567
22.4.1. Status register (CMP_STAT) 567
22.4.2. Interrupt flag clear register (CMP_IFC) 567
22.4.3. CMP0 control/status register (CMP0_CS) 568
23. Universal synchronous/asynchronous receiver /transmitter (USART) 571
23.1. Overview 571
23.2. Characteristics 571
23.3. Function overview 572
23.3.1. USART frame format 573
23.3.2. Baud rate generation 574
23.3.3. USART transmitter 574
23.3.4. USART receiver 575
23.3.5. Use DMA for data buffer access 577
23.3.6. Hardware flow control 579
23.3.7. Multi-processor communication 580
23.3.8. LIN mode 581
23.3.9. Synchronous mode 582
23.3.10. IrDA SIR ENDEC mode 582
23.3.11. Half-duplex communication mode 584
23.3.12. Smartcard (ISO7816-3) mode 584
23.3.13. USART interrupts 586
23.4. Register definition 588
23.4.1. Status register 0 (USART_STAT0) 588
23.4.2. Data register (USART_DATA) 590
23.4.3. Baud rate register (USART_BAUD) 590
23.4.4. Control register 0 (USART_CTL0) 591
23.4.5. Control register 1 (USART_CTL1) 593
23.4.6. Control register 2 (USART_CTL2) 594
23.4.7. Guard time and prescaler register (USART_GP) 596

23.4.8. Control register 3 (USART_CTL3) 597
23.4.9. Receiver timeout register (USART_RT) 598
23.4.10. Status register 1 (USART_STAT1) 599
23.4.11. Coherence control register (USART_CHC) 600
24. Serial peripheral interface/Inter-IC sound (SPI/I2S) 602
24.1. Overview 602
24.2. Characteristics 602
24.2.1. SPI characteristics 602
24.2.2. I2S characteristics 602
24.3. SPI function overview 603
24.3.1. SPI block diagram 603
24.3.2. SPI signal description 603
24.3.3. SPI clock timing and data format 604
24.3.4. NSS function 605
24.3.5. SPI operation modes 606
24.3.6. DMA function 614
24.3.7. CRC function 614
24.3.8. SPI interrupts 615
24.4. I2S function overview 617
24.4.1. I2S block diagram 617
24.4.2. I2S signal description 617
24.4.3. I2S audio standards 617
24.4.4. I2S clock 626
24.4.5. Operation 627
24.4.6. DMA function 631
24.4.7. I2S interrupts 631
24.5. Register definition 633
24.5.1. Control register 0 (SPI_CTL0) 633
24.5.2. Control register 1 (SPI_CTL1) 635
24.5.3. Status register (SPI_STAT) 636
24.5.4. Data register (SPI_DATA) 637
24.5.5. CRC polynomial register (SPI_CRCPOLY) 638
24.5.6. RX CRC register (SPI_RCRC) 638
24.5.7. TX CRC register (SPI_TCRC) 639
24.5.8. I2S control register (SPI_I2SCTL) 640
24.5.9. I2S clock prescaler register (SPI_I2SPSC) 641
24.5.10. Quad-SPI mode control register (SPI_QCTL) of SPI0 642
25. Inter-integrated circuit interface (I2C) 643
25.1. Overview 643
25.2. Characteristics 643

25.3. Function overview 643
25.3.1. Clock requirements 644
25.3.2. I2C communication flow 645
25.3.3. Noise filter 647
25.3.4. I2C timings configuration 648
25.3.5. I2C reset 650
25.3.6. Data transfer 650
25.3.7. I2C slave mode 652
25.3.8. I2C master mode 657
25.3.9. SMBus support 662
25.3.10. SMBus mode 665
25.3.11. Use DMA for data transfer 667
25.3.12. I2C error and interrupts 667
25.3.13. I2C debug mode 668
25.4. Register definition 669
25.4.1. Control register 0 (I2C_CTL0) 669
25.4.2. Control register 1 (I2C_CTL1) 671
25.4.3. Slave address register 0 (I2C_SADDR0) 673
25.4.4. Slave address register 1 (I2C_SADDR1) 674
25.4.5. Timing register (I2C_TIMING) 675
25.4.6. Timeout register (I2C_TIMEOUT) 676
25.4.7. Status register (I2C_STAT) 677
25.4.8. Status clear register (I2C_STATC) 680
25.4.9. PEC register (I2C_PEC) 681
25.4.10. Receive data register (I2C_RDATA) 681
25.4.11. Transmit data register (I2C_TDATA) 681
25.4.12. Control register 2 (I2C_CTL2) 682
26. Controller area network (CAN) 683
26.1. Overview 683
26.2. Characteristics 683
26.3. Function overview 684
26.3.1. Working mode 684
26.3.2. Communication modes 685
26.3.3. Data transmission 686
26.3.4. Data reception 688
26.3.5. Filtering function 690
26.3.6. Time-triggered communication 693
26.3.7. Communication parameters 693
26.3.8. CAN FD operation 695
26.3.9. Transmitter Delay Compensation 696
26.3.10. Error flags 697
26.3.11. CAN interrupts 697

26.4. Register definition 700
26.4.1. Control register (CAN_CTL) 700
26.4.2. Status register (CAN_STAT) 701
26.4.3. Transmit status register (CAN_TSTAT) 703
26.4.4. Receive message FIFO0 register (CAN_RFIFO0) 705
26.4.5. Receive message FIFO1 register (CAN_RFIFO1) 706
26.4.6. Interrupt enable register (CAN_INTEN) 707
26.4.7. Error register (CAN_ERR) 709
26.4.8. Bit timing register (CAN_BT) 710
26.4.9. FD control register (CAN_FDCTL) 711
26.4.10. FD status register (CAN_FDSTAT) 712
26.4.11. FD transmitter delay compensation register (CAN_FDTDC) 712
26.4.12. Date Bit timing register (CAN_DBT) 713
26.4.13. Transmit mailbox identifier register (CAN_TMIx) (x = 0...2) 713
26.4.14. Transmit mailbox property register (CAN_TMPx) (x = 0…2) 714
26.4.15. Transmit mailbox data0 register (CAN_TMDATA0x) (x = 0…2) 715
26.4.16. Transmit mailbox data1 register (CAN_TMDATA1x) (x = 0...2) 716
26.4.17. Rx FIFO mailbox identifier register (CAN_RFIFOMIx) (x = 0,1) 716
26.4.18. Rx FIFO mailbox property register (CAN_RFIFOMPx) (x = 0,1) 717
26.4.19. Rx FIFO mailbox data0 register (CAN_RFIFOMDATA0x) (x = 0,1) 718
26.4.20. Rx FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x=0,1) 718
26.4.21. Filter control register (CAN_FCTL) 719
26.4.22. Filter mode configuration register (CAN_FMCFG) 719
26.4.23. Filter scale configuration register (CAN_FSCFG) 720
26.4.24. Filter associated FIFO register (CAN_FAFIFO) 720
26.4.25. Filter working register (CAN_FW) 721
26.4.26. Filter x data y register (CAN_FxDATAy) (x=0..27, y=0,1) 721

27. External memory controller (EXMC) 722

27.1. Overview 722

27.2. Characteristics 722

27.3. Function overview 722
27.3.1. Block diagram 722
27.3.2. Basic regulation of EXMC access 723
27.3.3. External device address mapping 724
27.3.4. NOR/PSRAM controller 726
27.3.5. NAND Flash controller 735

27.4. Registers definition 739
27.4.1. NOR/PSRAM controller registers 739
27.4.2. NAND Flash controller registers 742

28. Universal serial bus full-speed interface (USBFS) 747

28.1. Overview 747

28.2. Characteristics 747
28.3. Block diagram 748
28.4. Signal description 748
28.5. Function overview 748
28.5.1. USBFS clocks and working modes 748
28.5.2. USB host function 749
28.5.3. USB device function 751
28.5.4. Data FIFO 752
28.5.5. Operation guide 755
28.6. Interrupts 759
28.7. Register definition 761
28.7.1. Global control and status registers 761
28.7.2. Host control and status registers 777
28.7.3. Device control and status registers 789
28.7.4. Power and clock control register (USBFS_PWRCLKCTL) 811
29. Appendix 813
29.1. List of abbreviations used in register 813
29.2. List of terms 814
29.3. Available peripherals 814
30. Revision history 815

List of Figures

Figure 1-1. The structure of the Cortex®-M33 processor 30
Figure 1-2. GD32F50x series system architecture 32
Figure 1-3. ECC decoder 37
Figure 2-1. Block diagram of EXTI 64
Figure 3-1. Block diagram of DMA 71
Figure 3-2. Handshake mechanism 73
Figure 3-3. DMA interrupt logic 75
Figure 4-1. Block diagram of DMAMUX 84
Figure 4-2. Synchronization mode 86
Figure 4-3. Event generation 87
Figure 5-1. Process of page erase operation 102
Figure 5-2. Process of mass erase operation 103
Figure 5-3. Process of word program operation 105
Figure 6-1. Power supply overview 124
Figure 6-2. Waveform of the POR/PDR 126
Figure 6-3. Waveform of the LVD threshold 126
Figure 6-4. Waveform of the VAVD threshold 127
Figure 6-5. waveform of VOVD 128
Figure 6-6. waveform of VUVD 129
Figure 8-1. The system reset circuit 145
Figure 8-2. Clock tree (for GD32F502xx) 146
Figure 8-3. Clock tree (for GD32F503xx) 147
Figure 8-4. Clock tree (for GD32F505xx) 148
Figure 8-5. HXTAL clock source 149
Figure 8-6. HXTAL clock source in bypass mode 150
Figure 9-1. CTC overview 191
Figure 9-2. CTC trim counter 192
Figure 10-1. Basic structure of a standard I/O port bit 202
Figure 10-2. Basic structure of Input configuration 204
Figure 10-3. Basic structure of Output configuration 205
Figure 10-4. Basic structure of Analog configuration 205
Figure 10-5. Basic structure of Alternate function configuration 206
Figure 11-1. TRIGSEL main composition example 224
Figure 12-1. Advanced timer block diagram 248
Figure 12-2. Normal mode, internal clock divided by 1 249
Figure 12-3. Counter timing diagram with prescaler division change from 1 to 2 250
Figure 12-4. Timing diagram of up counting mode, PSC=0/2 251
Figure 12-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing 251
Figure 12-6. Timing diagram of down counting mode, PSC=0/2 252
Figure 12-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing 253
Figure 12-8. Timing diagram of center-aligned counting mode 254

Figure 12-9. Repetition counter timing diagram of center-aligned counting mode 255
Figure 12-10. Repetition counter timing diagram of up counting mode 255
Figure 12-11. Repetition counter timing diagram of down counting mode 256
Figure 12-12. Input capture logic for channel 0 256
Figure 12-13. Output compare logic 258
Figure 12-14. Output-compare in three modes 259
Figure 12-15. Timing diagram of EAPWM 260
Figure 12-16. Timing diagram of CAPWM 260
Figure 12-17. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 262
Figure 12-18. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 262
Figure 12-19. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 263
Figure 12-20. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 263
Figure 12-21. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 264
Figure 12-22. Four Channels outputs in Composite PWM mode 264
Figure 12-23. Channel x output PWM with（CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL） 266
Figure 12-24. Channel x output PWM with（CHxVAL = CHxCOMVAL_ADD） 267
Figure 12-25. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 267
Figure 12-26. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00) 269
Figure 12-27. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00) 269
Figure 12-28. Complementary output with dead time insertion 272
Figure 12-29. Complementary output with different dead time(DTDIFEN=1) 273
Figure 12-30. BREAK function logic diagram 274
Figure 12-31. Output behavior of the channel in response to BREAK (the break input high active and IOS=1) 274
Figure 12-32. Channel break function logic diagram 275
Figure 12-33. Example of counter operation in quadrature decoder interface mode 276
Figure 12-34. Example of quadrature decoder interface mode with CI0FE0 polarity inverted 276
Figure 12-35. Hall sensor is used for BLDC motor 278
Figure 12-36. Hall sensor timing between two timers 279
Figure 12-37. Restart mode 280
Figure 12-38. Pause mode 280
Figure 12-39. Event mode 281
Figure 12-40. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60 282
Figure 12-41. Trigger mode of TIMER0 controlled by enable signal of TIMER2 283
Figure 12-42. Trigger mode of TIMER0 controlled by update signal of TIMER2 283
Figure 12-43. Pause mode of TIMER0 controlled by enable signal of TIMER2 284
Figure 12-44. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2 285
Figure 12-45. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2 286
Figure 12-46. Triggering four timers by trigger out of basic timer 287
Figure 12-47. Counting direction in CINITDIR is 0 or 1 287
Figure 12-48. General Level 0 timer block diagram 334
Figure 12-49. Normal mode, internal clock divided by 1 335
Figure 12-50. Counter timing diagram with prescaler division change from 1 to 2 336

Figure 12-51. Timing chart of up counting mode, PSC=0/2 337
Figure 12-52. Timing chart of up counting, change TIMERx_CAR ongoing 337
Figure 12-53. Timing chart of down counting mode, PSC=0/2 338
Figure 12-54. Timing chart of down counting mode, change TIMERx_CAR ongoing 339
Figure 12-55. Timing chart of center-aligned counting mode 340
Figure 12-56. Input capture logic 341
Figure 12-57. Output compare logic (x=0,1,2,3) 342
Figure 12-58. Output-compare under three modes 343
Figure 12-59. Timing chart of EAPWM 344
Figure 12-60. Timing chart of CAPWM 344
Figure 12-61. Example of counter operation in decoder interface mode 346
Figure 12-62. Example of decoder interface mode with CI0FE0 polarity inverted 346
Figure 12-63. Quadrature decoder signal disconnection detection block diagram 347
Figure 12-64. Example of counter operation in non-quadrature mode 0 / 1 with CH1P=0 348
Figure 12-65. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0) 349
Figure 12-66. Restart mode 351
Figure 12-67. Pause mode 351
Figure 12-68. Event mode 352
Figure 12-69. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60 352
Figure 12-70. General level3 timer block diagram 380
Figure 12-71. Normal mode, internal clock divided by 1 381
Figure 12-72. Counter timing diagram with prescaler division change from 1 to 2 382
Figure 12-73. Timing diagram of up counting mode, PSC=0/2 383
Figure 12-74. Timing diagram of up counting mode, change TIMERx_CAR on the go 383
Figure 12-75. Timing diagram of down counting mode, PSC=0/2 384
Figure 12-76. Timing diagram of down counting mode, change TIMERx_CAR ongoing 385
Figure 12-77. Timing diagram of center-aligned counting mode 386
Figure 12-78. Repetition counter timing diagram of center-aligned counting mode 387
Figure 12-79. Repetition counter timing diagram of up counting mode 387
Figure 12-80. Repetition counter timing diagram of down counting mode 388
Figure 12-81. Input capture logic for channel 0 389
Figure 12-82. Input capture logic for multi mode channel 0 389
Figure 12-83. Output compare logic (when MCHxMSEL = 2'00, x=0) 391
Figure 12-84. Output compare logic (when MCHxMSEL = 2'11, x=0) 391
Figure 12-85. Output compare logic (x=1) 391
Figure 12-86. Output-compare in three modes 393
Figure 12-87. Timing diagram of EAPWM 394
Figure 12-88. Timing diagram of CAPWM 394
Figure 12-89. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 396
Figure 12-90. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 396
Figure 12-91. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 397
Figure 12-92. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 397
Figure 12-93. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 398

Figure 12-94. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00) 399
Figure 12-95. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00) 399
Figure 12-96. Complementary output with dead-time insertion 402
Figure 12-97. BREAK function logic diagram 403
Figure 12-98. Output behavior of the channel in response to BREAK (the break input high active and IOS=1) 403
Figure 12-99. Restart mode 405
Figure 12-100. Pause mode 405
Figure 12-101. Event mode 405
Figure 12-102. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 406
Figure 12-103. Basic timer block diagram 441
Figure 12-104. Normal mode, internal clock divided by 1 442
Figure 12-105. Counter timing diagram with prescaler division change from 1 to 2 442
Figure 12-106. Timing chart of up counting mode, PSC=0/2 443
Figure 12-107. Timing chart of up counting mode, change TIMERx_CAR ongoing 444
Figure 13-1. Block diagram of RTC 451
Figure 14-1. Free watchdog timer block diagram 460
Figure 14-2. Window watchdog timer block diagram 465
Figure 14-3. Window watchdog timing diagram 466
Figure 16-1. Block diagram of CRC calculation unit 477
Figure 17-1. TRNG block diagram 482
Figure 18-1. DATAM No swapping and Half-word swapping 488
Figure 18-2. DATAM Byte swapping and Bit swapping 488
Figure 18-3. CAU diagram 489
Figure 18-4. AES ECB encryption 490
Figure 18-5. AES ECB decryption 491
Figure 19-1. DATAM No swapping and Half-word swapping 503
Figure 19-2. DATAM Byte swapping and Bit swapping 503
Figure 19-3. HAU block diagram 504
Figure 20-1. ADC module block diagram 516
Figure 20-2. Single operation mode 517
Figure 20-3. Continuous operation mode 518
Figure 20-4. Scan operation mode, continuous disable 519
Figure 20-5. Scan operation mode, continuous enable 520
Figure 20-6. Discontinuous operation mode 520
Figure 20-7. Auto-insertion, CNT = 1 521
Figure 20-8. Triggered insertion 522
Figure 20-9. Data latch 522
Figure 20-10. Sequence data from ADC_LDATAx 523
Figure 20-11. Data storage mode of 12-bit resolution 524
Figure 20-12. Data storage mode of 6-bit resolution 524
Figure 20-13. 20-bit to 16-bit result truncation 527
Figure 20-14. Numerical example with 5-bits shift and rounding 528
Figure 20-15. ADC sync block diagram 529

Figure 20-16. Routine parallel mode on 10 channels 530
Figure 20-17. Inserted parallel mode on 4 channels 531
Figure 20-18. Routine follow-up fast mode (the CTN bit of ADCs are set) 532
Figure 20-19. Routine follow-up slow mode 532
Figure 20-20. Inserted trigger rotation: DISIC=0, IL=1 533
Figure 20-21. Inserted trigger rotation: DISIC=1, IL=1 533
Figure 20-22. Routine parallel & inserted trigger rotation mode: SYNCM = 4'b0010 534
Figure 20-23. Trigger occurs during inserted conversion: SYNCM = 4'b0010 535
Figure 20-24. Follow-up single channel with inserted sequence CH1, CH2 535
Figure 21-1. DAC block diagram 553
Figure 21-2. DAC LFSR algorithm 555
Figure 21-3. DAC triangle noise wave 555
Figure 22-1. CMP block diagram 563
Figure 22-2. CMP hysteresis 564
Figure 22-3. The CMP outputs signal blanking 565
Figure 22-4. CMP noise filter and interrupt operation 566
Figure 23-1. USART module block diagram 573
Figure 23-2. USART character frame (8 bits data and 1 stop bit) 573
Figure 23-3. USART transmit procedure 575
Figure 23-4. Receiving a frame bit by oversampling method (OSB=0) 576
Figure 23-5. Configuration step when using DMA for USART transmission 578
Figure 23-6. Configuration steps when using DMA for USART reception 579
Figure 23-7. Hardware flow control between two USARTs 579
Figure 23-8. Hardware flow control 580
Figure 23-9. Break frame occurs during idle state 581
Figure 23-10. Break frame occurs during a frame 581
Figure 23-11. Example of USART in synchronous mode 582
Figure 23-12. 8-bit format USART synchronous waveform (CLEN=1) 582
Figure 23-13. IrDA SIR ENDEC module 583
Figure 23-14. IrDA data modulation 583
Figure 23-15. ISO7816-3 frame format 584
Figure 23-16. USART interrupt mapping diagram 587
Figure 24-1. Block diagram of SPI 603
Figure 24-2. SPI timing diagram in normal mode 604
Figure 24-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0) 605
Figure 24-4. A typical Full-duplex connection 607
Figure 24-5. A typical simplex connection (Master: Receive, Slave: Transmit) 608
Figure 24-6. A typical simplex connection (Master: Transmit only, Slave: Receive) 608
Figure 24-7. A typical bidirectional connection 608
Figure 24-8. Timing diagram of TI master mode with discontinuous transfer 610
Figure 24-9. Timing diagram of TI master mode with continuous transfer 610
Figure 24-10. Timing diagram of TI slave mode 611
Figure 24-11. Timing diagram of NSS pulse with continuous transmit 611
Figure 24-12. Timing diagram of quad write operation in Quad-SPI mode 612

Figure 24-13. Timing diagram of quad read operation in Quad-SPI mode 613
Figure 24-14. Block diagram of I2S 617
Figure 24-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 618
Figure 24-16. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 618
Figure 24-17. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 619
Figure 24-18. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 619
Figure 24-19. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 619
Figure 24-20. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 619
Figure 24-21. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 620
Figure 24-22. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 620
Figure 24-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 620
Figure 24-24. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 620
Figure 24-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 620
Figure 24-26. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 621
Figure 24-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 621
Figure 24-28. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 621
Figure 24-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 621
Figure 24-30. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 621
Figure 24-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 622
Figure 24-32. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 622
Figure 24-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 622
Figure 24-34. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 622
Figure 24-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 623
Figure 24-36. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 623
Figure 24-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 623
Figure 24-38. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 623
Figure 24-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 623
Figure24-40. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 623
Figure 24-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 624
Figure 24-42. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 624
Figure 24-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 624
Figure24-44. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 624
Figure 24-45. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 624

Figure 24-46. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 625
Figure 24-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 625
Figure 24-48. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 625
Figure 24-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 625
Figure 24-50. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 625
Figure 24-51. Block diagram of I2S clock generator 626
Figure 24-52. I2S initialization sequence 628
Figure 24-53. I2S master reception disabling sequence 630
Figure 25-1. I2C module block diagram 644
Figure 25-2. Data validation 645
Figure 25-3. START and STOP signal 646
Figure 25-4. I2C communication flow with 10-bit address (Master Transmit) 646
Figure 25-5. I2C communication flow with 7-bit address (Master Transmit) 647
Figure 25-6. I2C communication flow with 7-bit address (Master Receive) 647
Figure 25-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0) 647
Figure 25-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1) 647
Figure 25-9. Data hold time 648
Figure 25-10. Data setup time 648
Figure 25-11. Data transmission 650
Figure 25-12. Data reception 651
Figure 25-13. I2C initialization in slave mode 654
Figure 25-14. Programming model for slave transmitting when SS=0 655
Figure 25-15. Programming model for slave transmitting when SS=1 656
Figure 25-16. Programming model for slave receiving 657
Figure 25-17. I2C initialization in master mode 658
Figure 25-18. Programming model for master transmitting (N<=255) 659
Figure 25-19. Programming model for master transmitting (N>255) 660
Figure 25-20. Programming model for master receiving (N<=255) 661
Figure 25-21. Programming model for master receiving (N>255) 662
Figure 25-22. SMBus Master Transmitter and Slave Receiver communication flow 666
Figure 25-23. SMBus Master Receiver and Slave Transmitter communication flow 666
Figure 26-1. CAN module block diagram 684
Figure 26-2. Transmission register 686
Figure 26-3. State of transmit mailbox 687
Figure 26-4. Reception register 689
Figure 26-5. 32-bit filter 690
Figure 26-6. 16-bit filter 690

Figure 26-7. 32-bit mask mode filter 690
Figure 26-8. 16-bit mask mode filter 691
Figure 26-9. 32-bit list mode filter 691
Figure 26-10. 16-bit list mode filter 691
Figure 26-11. The bit time 694
Figure 26-12. Transmitter Delay Measurement 696
Figure 27-1. The EXMC block diagram 723
Figure 27-2. EXMC memory banks 724
Figure 27-3. Four regions of bank0 address mapping 724
Figure 27-4. NAND address mapping 725
Figure 27-5. Diagram of bank1 common space 726
Figure 27-6. Multiplex mode read access 729
Figure 27-7. Multiplex mode write access 729
Figure 27-8. Read access timing diagram under async-wait signal assertion 731
Figure 27-9. Write access timing diagram under async-wait signal assertion 731
Figure 27-10. Read timing of synchronous multiplexed burst mode 733
Figure 27-11. Write timing of synchronous multiplexed burst mode 734
Figure 27-12. Access timing of common memory space of NAND flash Controller 737
Figure 27-13. Access to none "NCE don’t care" NAND Flash 738
Figure 28-1. USBFS block diagram 748
Figure 28-2. Connection with host or device mode 749
Figure 28-3. State transition diagram of host port 750
Figure 28-4. HOST mode FIFO space in SRAM 753
Figure 28-5. Host mode FIFO access register map 753
Figure 28-6. Device mode FIFO space in SRAM 754
Figure 28-7. Device mode FIFO access register map 755

List of Tables

Table 1-1. The interconnection relationship of the AHB interconnect matrix 30
Table 1-2. Memory map of GD32F50x devices 33
Table 1-3. Boot modes 38
Table 2-1. NVIC exception types in Cortex®-M33 61
Table 2-2. Interrupt vector table 61
Table 2-3. EXTI source 65
Table 3-1. DMA transfer operation 72
Table 3-2. interrupt events 75
Table 4-1. DMAMUX signals 85
Table 4-2. Interrupt events 89
Table 4-3. Request multiplexer input mapping 90
Table 4-4. Trigger input mapping 92
Table 4-5. Synchronization input mapping 93
Table 5-1. GD32F50x base address and size for flash memory 99
Table 5-2. Option byte 106
Table 5-3. WP bit for pages protected 107
Table 5-4. OTP available programming width 108
Table 5-5. OTP0 lock 108
Table 5-6. OTP1 lock and data 109
Table 5-7. OTP2 lock and data 109
Table 5-8. OTP3 lock and data 110
Table 5-9. Security protection 110
Table 6-1. Power saving mode summary 131
Table 8-1. Clock output source select 153
Table 8-2. V_{CORE} domain voltage selected in deep-sleep mode 153
Table 10-1. GPIO configuration table 202
Table 11-1. Trigger input bit fields selection 224
Table 11-2. TRIGSEL input and output mapping 226
Table 12-1. Timers (TIMERx) are divided into four sorts 246
Table 12-2.The composite PWM pulse width 261
Table 12-3.The asymmetric PWM pulse width 265
Table 12-4. Complementary outputs controlled by parameters 271
Table 12-5. Output behavior of the channel in response to a BREAK (the break input is high active) 274
Table 12-6. Counting direction in different quadrature decoder signals 276
Table 12-7. Examples of slave mode 279
Table 12-8. Counting direction in different quadrature decoder signals 346
Table 12-9. the counter operation in non-quadrature mode 1 348
Table 12-10. the counter operation in non-quadrature decoder mode 2 / 3 349
Table 12-11. Examples of slave mode 350
Table 12-12.The Composite PWM pulse width 395

Table 12-13. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 401
Table 12-14. Slave mode example table 404
Table 14-1. Min/max FWDGT timeout period at 40 kHz (IRC40K) 460
Table 14-2. Min / max timeout value at 110 MHz (f_{PCLK1}) 467
Table 15-1. Pin assignment 471
Table 20-1. ADC internal input channels 515
Table 20-2. ADC input pins definition 515
Table 20-3. External trigger mode and type 525
Table 20-4. t_{CONV} timings depending on resolution 526
Table 20-5. Maximum output results for N and M (Grayed values indicates truncation) 528
Table 20-6. ADC sync mode table 529
Table 21-1. DAC I/O description 553
Table 21-2. DAC triggers and outputs summary 553
Table 21-3. Triggers of DAC 554
Table 22-1. CMP inputs and outputs summary 563
Table 23-1. Description of USART important pins 572
Table 23-2. Configuration of stop bits 573
Table 23-3. USART interrupt requests 586
Table 24-1. SPI signal description 603
Table 24-2. Quad-SPI signal description 604
Table 24-3. NSS function in slave mode 605
Table 24-4. NSS function in master mode 606
Table 24-5. SPI operation modes 606
Table 24-6. SPI interrupt requests 616
Table 24-7. I2S bitrate calculation formulas 626
Table 24-8. Audio sampling frequency calculation formulas 626
Table 24-9. Direction of I2S interface signals for each operation mode 627
Table 24-10. I2S interrupt 632
Table 25-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors) 644
Table 25-2. Data setup time and data hold time 649
Table 25-3. Communication modes to be shut down 651
Table 25-4. SMBus with PEC configuration 664
Table 25-5. I2C error flags 667
Table 25-6. I2C interrupt events 667
Table 26-1. 32-bit filter number 691
Table 26-2. Filtering index 692
Table 26-3. CAN Event / Interrupt flags 698
Table 27-1. NOR Flash interface signals description 727
Table 27-2. PSRAM muxed signal description 727
Table 27-3. EXMC bank 0 supports all transactions 727
Table 27-4. NOR / PSRAM controller timing parameters 728
Table 27-5. EXMC_timing models 728
Table 27-6. Multiplex mode related registers configuration 730

Table 27-7. Timing configurations of synchronous multiplexed read mode 733
Table 27-8. Timing configurations of synchronous multiplexed write mode 734
Table 27-9. 8-bit or 16-bit NAND interface signal 735
Table 27-10. Bank1 of EXMC support the memory and access mode 736
Table 27-11. NAND Flash programmable parameters 736
Table 28-1. USBFS signal description 748
Table 28-2. USBFS global interrupt 759
Table 29-1. List of abbreviations used in register 813
Table 29-2. List of terms 814
Table 30-1. Revision history 815

1. System and memory architecture

The GD32F50x series are 32-bit general-purpose microcontrollers based on the Arm® Cortex®-M33 processor. The Arm® Cortex®-M33 processor includes two AHB buses known as Code and System buses. All memory accesses of the Arm® Cortex®-M33 processor are executed on these two buses according to the different purposes and the target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

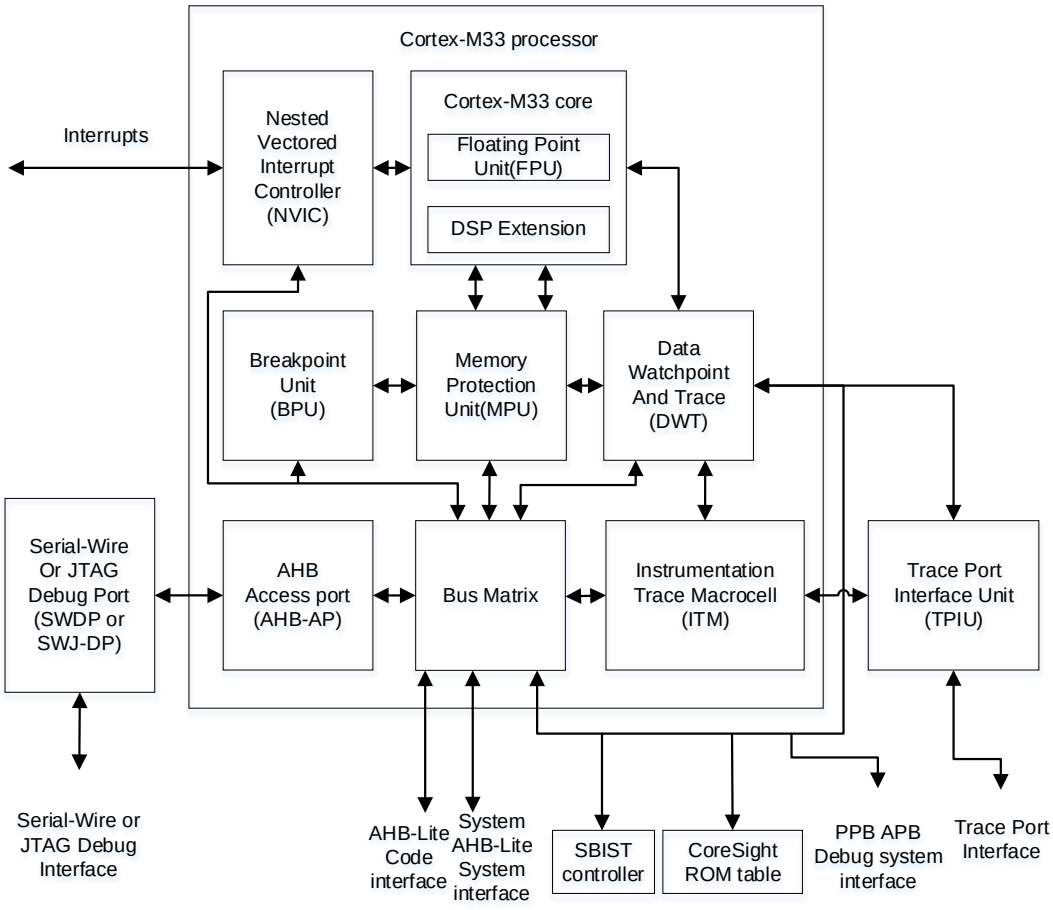
1.1. Arm® Cortex®-M33 processor

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

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

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

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



1.2. System architecture

A 32-bit multilayer bus is implemented in the GD32F50x devices, which enables parallel access paths between multiple masters and slaves in the system. The multilayer bus consists of an AHB interconnect matrix, one AHB bus and two APB buses. The interconnection relationship of the AHB interconnect matrix is shown below. In the following table, “1” indicates the corresponding master is able to access the corresponding slave through the AHB interconnect matrix, while the blank means the corresponding master cannot access the corresponding slave through the AHB interconnect matrix.

Table 1-1. The interconnection relationship of the AHB interconnect matrix

	CBUS	SBUS	DMA0	DMA1
FMC	1		1	1
SRAM	1	1	1	1
AHB		1	1	1
EXMC	1	1	1	1

As is shown ***Table 1-1. The interconnection relationship of the AHB interconnect matrix***,

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

The GD32F50x devices support a master port timeout monitoring mechanism to detect system bus transmission timeouts.

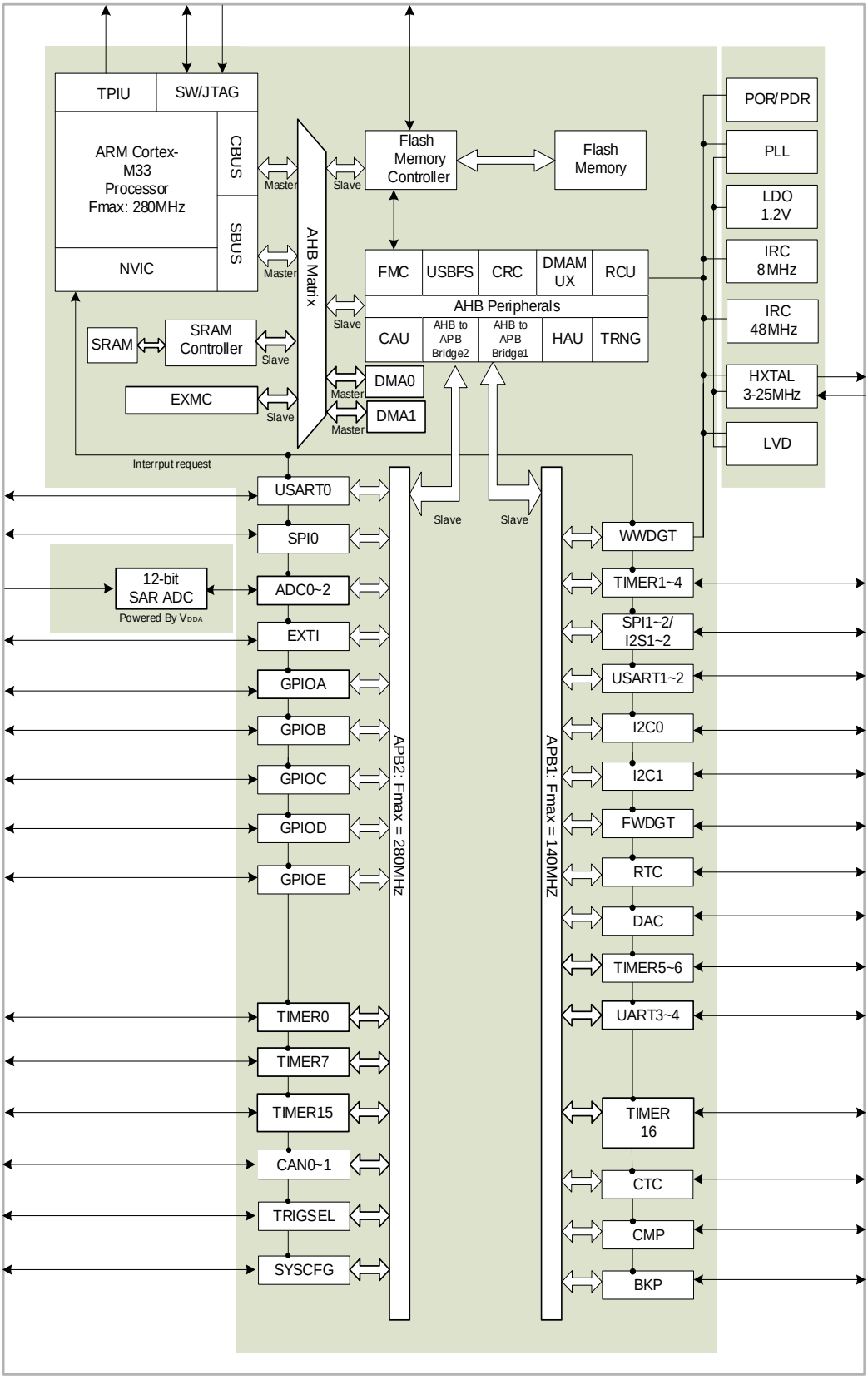
The system monitors timeout conditions for 4 AHB master ports (CPU CBUS, CPU SBUS, DMA0, DMA1). When access anomalies cause prolonged bus access blocking, the hardware automatically generates bus error feedback to the corresponding master, while the SYSCFG module sets the corresponding timeout status flag bits and triggers an NMI interrupt for exception handling.

The bus timeout register (SYSCFG_BUSTO) is used to enable the timeout monitoring function for corresponding master ports, and the bus timeout status register (SYSCFG_BUSTOSTAT) is used to check the timeout status flag bits of each master port.

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM, EXMC, AHB, APB1 and APB2. FMC is the flash memory controller. SRAM are on-chip static random access memories. EXMC is the external memory controller. AHB is the AHB bus connected with all of the AHB slaves, while APB1 and APB2 are the two APB buses connected with all of the APB slaves. The two APB buses connect with all the APB peripherals. APB1 is limited to 140 MHz, APB2 operates at full speed (up to 280MHz depending on the device).

These are interconnected using a multilayer AHB bus architecture as shown in ***Figure 1-2. GD32F50x series system architecture***:

Figure 1-2. GD32F50x series system architecture



Note: The maximum operating clock frequency of the GD32F502xx system clock can reach

up to 200MHz, the maximum operating clock frequency of the GD32F503xx system clock can reach up to 252MHz, while the maximum operating clock frequency of the GD32F505xx system clock can reach up to 280MHz.

1.3. Memory map

The Arm® Cortex®-M33 processor is structured in Harvard architecture which can use separate buses to fetch instructions and load/store data. 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®-M33 since the bus address width is 32-bit. Additionally, a pre-defined memory map is provided by the Cortex®-M33 processor to reduce the software complexity of repeated implementation of different device vendors. In the map, some regions are used by the Arm® Cortex®-M33 system peripherals which can not be modified. However, the other regions are available to the vendors. ***Table 1-2. Memory map of GD32F50x devices*** shows the memory map of the GD32F50x series devices, including Code, SRAM, peripheral, and other pre-defined regions. Almost each peripheral is allocated 1KB of space. This allows simplifying the address decoding for each peripheral.

Table 1-2. Memory map of GD32F50x devices

Pre-defined Regions	Bus	Address	Peripherals
External device	AHB3	0xA000 0000 - 0xA000 0FFF	EXMC - SWREG
External RAM		0x9000 0000 - 0x9FFF FFFF	Reserved
		0x7000 0000 - 0x8FFF FFFF	EXMC - NAND
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM
Peripheral	AHB1	0x5000 0000 - 0x5003 FFFF	USBFS
		0x4008 0000 - 0x4FFF FFFF	Reserved
		0x4004 0000 - 0x4007 FFFF	Reserved
		0x4002 BC00 - 0x4003 FFFF	Reserved
		0x4002 B000 - 0x4002 BBFF	Reserved
		0x4002 A000 - 0x4002 AFFF	Reserved
		0x4002 8000 - 0x4002 9FFF	Reserved
		0x4002 6800 - 0x4002 7FFF	Reserved
		0x4002 6400 - 0x4002 67FF	Reserved
		0x4002 6000 - 0x4002 63FF	Reserved
		0x4002 5000 - 0x4002 5FFF	Reserved
		0x4002 4000 - 0x4002 4FFF	Reserved
		0x4002 3C00 - 0x4002 3FFF	TRNG
		0x4002 3800 - 0x4002 3BFF	HAU

Pre-defined Regions	Bus	Address	Peripherals
		0x4002 3400 - 0x4002 37FF	CAU
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2C00 - 0x4002 2FFF	Reserved
		0x4002 2800 - 0x4002 2BFF	Reserved
		0x4002 2400 - 0x4002 27FF	Reserved
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	Reserved
		0x4002 1800 - 0x4002 1BFF	Reserved
		0x4002 1400 - 0x4002 17FF	Reserved
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	Reserved
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
		0x4001 8400 - 0x4001 FFFF	Reserved
		0x4001 8000 - 0x4001 83FF	Reserved
	APB2	0x4001 7C00 - 0x4001 7FFF	Reserved
		0x4001 7800 - 0x4001 7BFF	Reserved
		0x4001 7400 - 0x4001 77FF	Reserved
		0x4001 7000 - 0x4001 73FF	Reserved
		0x4001 6C00 - 0x4001 6FFF	Reserved
		0x4001 6800 - 0x4001 6BFF	Reserved
		0x4001 6500 - 0x4001 67FF	Reserved
		0x4001 6400 - 0x4001 64FF	Shared CAN/APB SRAM1
		0x4001 6000 - 0x4001 63FF	Shared CAN/APB SRAM0
		0x4001 5C00 - 0x4001 5FFF	CAN1
		0x4001 5800 - 0x4001 5BFF	CAN0
		0x4001 5400 - 0x4001 57FF	Reserved
		0x4001 5000 - 0x4001 53FF	TIMER15
		0x4001 4C00 - 0x4001 4FFF	Reserved
		0x4001 4800 - 0x4001 4BFF	Reserved
		0x4001 4400 - 0x4001 47FF	TRIGSEL
		0x4001 4000 - 0x4001 43FF	SYSCFG
		0x4001 3C00 - 0x4001 3FFF	ADC2
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	ADC1

Pre-defined Regions	Bus	Address	Peripherals
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	Reserved
		0x4001 1C00 - 0x4001 1FFF	Reserved
		0x4001 1800 - 0x4001 1BFF	GPIOE
		0x4001 1400 - 0x4001 17FF	GPIOD
		0x4001 1000 - 0x4001 13FF	GPIOC
		0x4001 0C00 - 0x4001 0FFF	GPIOB
		0x4001 0800 - 0x4001 0BFF	GPIOA
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	AFIO
	APB1	0x4000 CC00 - 0x4000 FFFF	Reserved
		0x4000 C800 - 0x4000 CBFF	CTC
		0x4000 C400 - 0x4000 C7FF	Reserved
		0x4000 C000 - 0x4000 C3FF	Reserved
		0x4000 8000 - 0x4000 BFFF	Reserved
		0x4000 7C00 - 0x4000 7FFF	Reserved
		0x4000 7800 - 0x4000 7BFF	CMP
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	BKP
		0x4000 6800 - 0x4000 6BFF	Reserved
		0x4000 6400 - 0x4000 67FF	Reserved
		0x4000 6000 - 0x4000 63FF	Reserved
		0x4000 5C00 - 0x4000 5FFF	Reserved
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	Reserved
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	Reserved
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	Reserved
		0x4000 2000 - 0x4000 23FF	Reserved
		0x4000 1C00 - 0x4000 1FFF	Reserved
		0x4000 1800 - 0x4000 1BFF	TIMER16

Pre-defined Regions	Bus	Address	Peripherals
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	TIMER4
		0x4000 0800 - 0x4000 0BFF	TIMER3
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM	AHB	0x2007 0000 - 0x3FFF FFFF	Reserved
		0x2006 0000 - 0x2006 FFFF	Reserved
		0x2003 0000 - 0x2005 FFFF	Reserved
		0x2000 0000 - 0x2002 FFFF	SRAM
Code	AHB	0x1FFF F810 - 0x1FFF FFFF	Reserved
		0x1FFF F800 - 0x1FFF F80F	Option Bytes
		0x1FFF B000 - 0x1FFF F7FF	Boot loader
		0x1FFF 7A10 - 0x1FFF AFFF	Reserved
		0x1FFF 7940 - 0x1FFF 7A0F	Reserved
		0x1FFF 7930 - 0x1FFF 793F	OTP3(lock area)
		0x1FFF 7900 - 0x1FFF 792F	OTP3(data area)
		0x1FFF 7880 - 0x1FFF 788F	Reserved
		0x1FFF 7840 - 0x1FFF 787F	OTP0(lock area)
		0x1FFF 7800 - 0x1FFF 783F	OTP0(data area)
		0x1FFF 0000 - 0x1FFF 77FF	Reserved
		0x1FFE C010 - 0x1FFE FFFF	Reserved
		0x1FFE C000 - 0x1FFE C00F	Reserved
		0x1FF2 0190 - 0x1FFE BFFF	Reserved
		0x1FF2 0110 - 0x1FF2 018F	OTP2 (lock area)
		0x1FF2 0100 - 0x1FF2 010F	OTP1 (lock area)
		0x1FF2 0000 - 0x1FF2 00FF	OTP2 (data area)
		0x1FF0 0000 - 0x1FF1 FFFF	OTP1 (data area)
		0x1001 0000 - 0x1FEF FFFF	Reserved
		0x1000 0000 - 0x1000 FFFF	Reserved
		0x083C 0000 - 0x0FFF FFFF	Reserved
		0x0830 0000 - 0x083B FFFF	Reserved
		0x0810 0000 - 0x082F FFFF	Reserved
		0x0800 0000 - 0x080F FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	Reserved
		0x0010 0000 - 0x002F FFFF	Aliased to Main Flash or Boot loader
		0x0002 0000 - 0x000F FFFF	
		0x0000 0000 - 0x0001 FFFF	

1.3.1. On-chip SRAM memory

The GD32F50x series of devices contain up to 192 KB of on-chip SRAM which starts at the address 0x2000 0000. It supports byte, half-word (16 bits), and word (32 bits) accesses.

ECC

When reading and writing first 32KB SRAM, it supports 7-bit ECC function. It can correct 1-bit error and detect multiple bits (two bits) error.

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

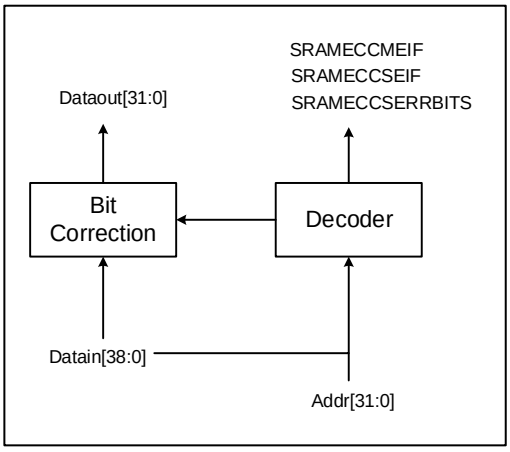
The ECC module is composed of an encoder and a decoder.

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

Decoder: When performing a SRAM read operation, it uses the same algorithm as the encoder to decode and generate a 7-bit ECC code. The ECC code includes ECC error status and information which specific bit of the 32-bit data has single bit error.

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

Figure 1-3. ECC decoder



EEIC

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

Single bit correction error event

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

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

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

When a multi-bits non-correction error event is detected, EEIC:

(1) The SRAMECCMEIF bit in SYSCFG_SRAMECCSTAT register will be set. Software can clear it by writing 1.

(2) The SYSCFG_SRAMECCCS records the address where the multi-bits non-correction error event occurred.

Single bit correction error interrupt

Set the SRAMECCSEIE bit in SYSCFG_SRAMECCCS register. When a single-bit error correctable event is detected, NMI and SRAMECC interrupt will be generated.

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

Set the SRAMECCMEIE bit in SYSCFG_SRAMECCCS register. When a multi-bits error non-correction event is detected, NMI and SRAMECC interrupt will be generated.

1.3.2. On-chip flash memory overview

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

- Up to 1024KB of main flash memory.
- Up to 18KB of information blocks for the boot loader.
- Option bytes to configure the device.

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

1.4. Boot configuration

The GD32F50x devices provide four kinds of boot sources. The boot mode is influenced by Security Protection, NBTSB and BTFOSEL bits in OTP3, and Boot pins. The details are shown in the following table ***Table 1-3. Boot modes***.

The value on the BOOT0 and BOOT1 pins is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT0 and BOOT1 pins after a power-on reset or a system reset to select the required boot source. Once the two pins have been sampled, they are free and can be used for other purposes.

Table 1-3. Boot modes

Security	OTP3	Boot pin	BOOT_MODE[2	Boot Select

protection	NBTSB	BTFOSEL	BOOT0	BOOT1	:0]	
No protection/ Protection level low	0	x	1	1	011	SRAM
No protection/ Protection level low	0	x	1	0	001	BootLoader
No protection/ Protection level low	0	0	0	x	000	Main Flash
No protection/ Protection level low	0	1	0	x	101	OTP1
x	1	0	x	x	000	Main Flash
x	1	1	x	x	101	OTP1
Protection level high	x	0	x	x	000	Main Flash
Protection level high	x	1	x	x	101	OTP1

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

The corresponding memory space of the selected boot source is aliased in the boot memory space which begins at the address 0x0000 0000. When the on-chip SRAM 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 the offset register.

When the main flash memory is selected to be the boot source, the memory space beginning at the address 0x0800 0000 is aliased in the boot memory space. When OTP1 is selected to be the boot source, the memory space beginning at the address 0x1FF0 0000 is aliased in the boot memory space.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory.GD32F50x MCU embedded bootloader supports multi interfaces to update the Flash memory. There will be USART port, and standard USB port can be used on GD32F50x line products.

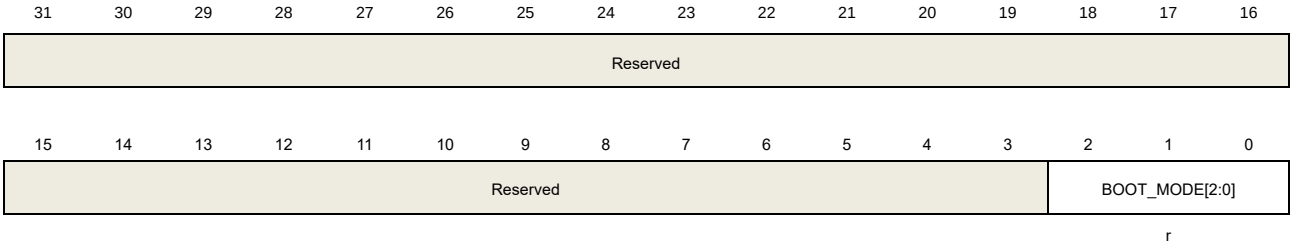
1.5. System configuration registers

SYSCFG base address: 0x4001 4000

1.5.1. Configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00
Reset value: 0x0000 000X (X indicates BOOT_MODE[2:0] may be any value according to the BOOT0 and BOOT1 pins)

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

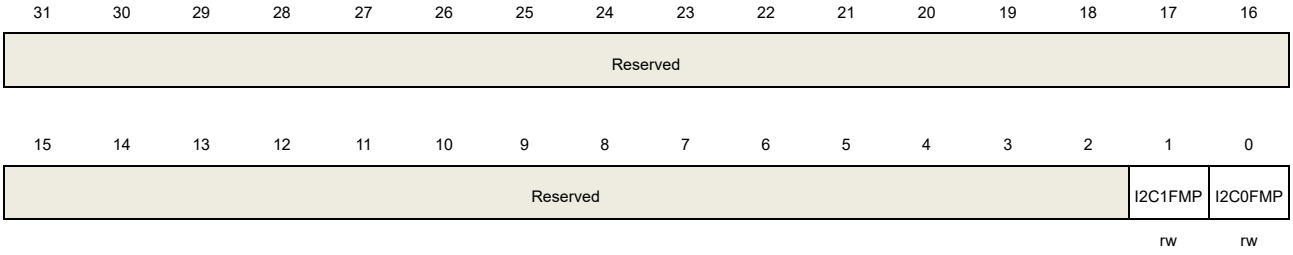


Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	BOOT_MODE[2:0]	Boot mode. Refer to ***Table 1-3. Boot modes***

1.5.2. Configuration register 1 (SYSCFG_CFG1)

Address offset: 0x04
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	I2C1FMP	I2C1 Fm+ This bit enables Fm+ on I2C1. 0: Fm+ disabled 1: Fm+ enabled
0	I2C0FMP	I2C0 Fm+

This bit enables Fm+ on I2C0.

0: Fm+ disabled

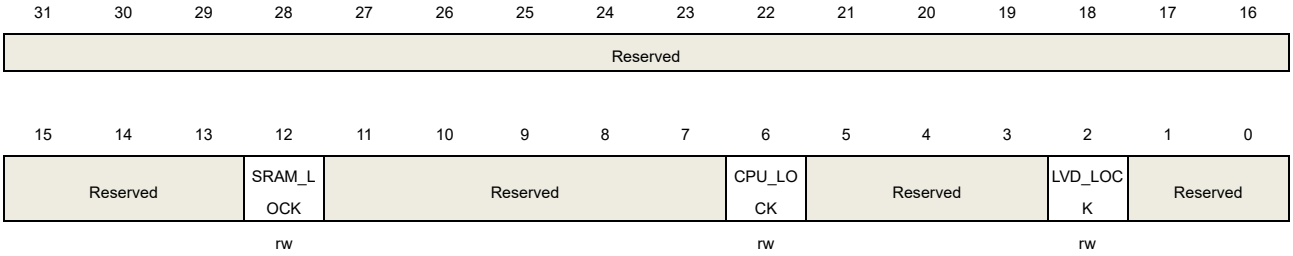
1: Fm+ enabled

1.5.3. Lockup control register (SYSCFG_LKCTL)

Address offset: 0x018

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	SRAM_LOCK	SRAM ECC double error lockup bit This bit is set by software and cleared by a system reset. 0: SRAM ECC double error signal is disconnected from the break input of TIMER0/7/15/16 1: SRAM ECC double error signal is connected from the break input of TIMER0/7/15/16
11:7	Reserved	Must be kept at reset value.
6	CPU_LOCK	CPU lockup bit This bit is set by software and cleared by a system reset. 0: CPU lockup signal is disconnected from the break input of TIMER0/7/15/16 1: CPU lockup signal is connected from the break input of TIMER0/7/15/16
5:3	Reserved	Must be kept at reset value.
2	LVD_LOCK	Low voltage detector lockup bit This bit is set by software and cleared by a system reset. 0: LVD signal is disconnected from the break input of TIMER0/7/15/16 1: LVD signal is connected from the break input of TIMER0/7/15/16
1:0	Reserved	Must be kept at reset value.

1.5.4. Bus timeout register (SYSCFG_BUSTO)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DMA1BUSTO	DMA0BUSTO	CPUSBUSTO	CPUCBUSTO
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	DMA1BUSTO	DMA1 bus timeout enable bit. 0: Disable 1: Enable
2	DMA0BUSTO	DMA0 bus timeout enable bit. 0: Disable 1: Enable
1	CPUSBUSTO	CPU Sbus timeout enable bit. 0: Disable 1: Enable
0	CPUCBUSTO	CPU Cbus timeout enable bit. 0: Disable 1: Enable

1.5.5. Timer input selection register (SYSCFG_TIMERCISEL)

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										TIMER16_CI0_SEL[1:0]		Reserved		TIMER15_CI0_SEL[1:0]	
										rw				rw	

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:4	TIMER16_CI0_SEL[1:0]	Selects TIMER16_CI0 input selection This bit selects the TIMER input source. 00: TIMER16_CH0 input

		01: Reserved 10: Reserved 11: CKOUT
3:2	Reserved	Must be kept at reset value.
1:0	TIMER15_CI0_SEL[1:0]	Selects TIMER15_CI0 input selection This bit selects the TIMER input source. 00: TIMER15_CH0 input 01: IRC40K 10: LXTAL 11: Reserved

1.5.6. FPU interrupt enable register (SYSCFG_FPUINTEN)

Address offset: 0x24

Reset value: 0x0000 001F

This register 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										IXIE	IAIE	OVFIE	UFIE	DZIE	IOPIE
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	IXIE	Inexact interrupt enable bit 0: Disable 1: Enable
4	IAIE	Input abnormal interrupt enable bit 0: Disable 1: Enable
3	OVFIE	Overflow interrupt enable bit 0: Disable 1: Enable
2	UFIE	Underflow interrupt enable bit 0: Disable 1: Enable
1	DZIE	Divide by 0 interrupt enable bit 0: Disable 1: Enable

0	IOPIE	Invalid operation interrupt enable bit 0: Disable 1: Enable

1.5.7. SRAM write protection register (SYSCFG_SRAMWP)

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
SRAM_P31WP	SRAM_P30WP	SRAM_P29WP	SRAM_P28WP	SRAM_P27WP	SRAM_P26WP	SRAM_P25WP	SRAM_P24WP	SRAM_P23WP	SRAM_P22WP	SRAM_P21WP	SRAM_P20WP	SRAM_P19WP	SRAM_P18WP	SRAM_P17WP	SRAM_P16WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAM_P15WP	SRAM_P14WP	SRAM_P13WP	SRAM_P12WP	SRAM_P11WP	SRAM_P10WP	SRAM_P9WP	SRAM_P8WP	SRAM_P7WP	SRAM_P6WP	SRAM_P5WP	SRAM_P4WP	SRAM_P3WP	SRAM_P2WP	SRAM_P1WP	SRAM_P0WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

Bits	Fields	Descriptions
31:0	SRAM_PxWP (x=0 to 31)	SRAM page x write protection These bits are set by software and cleared only by system reset. 0: Write protection of SRAM page x is disabled. 1: Write protection of SRAM page x is enabled.

1.5.8. SRAM ECC status register (SYSCFG_SRAMECCSTAT)

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
Reserved														SRAMECCSEIF	SRAMECCMEIF
														rc_w1	rc_w1

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SRAMECCSEIF	Indicates the single-bit error SRAM single bit correction event The software can clear it by writing 1. 0: no SRAM single bit correction event is detected.

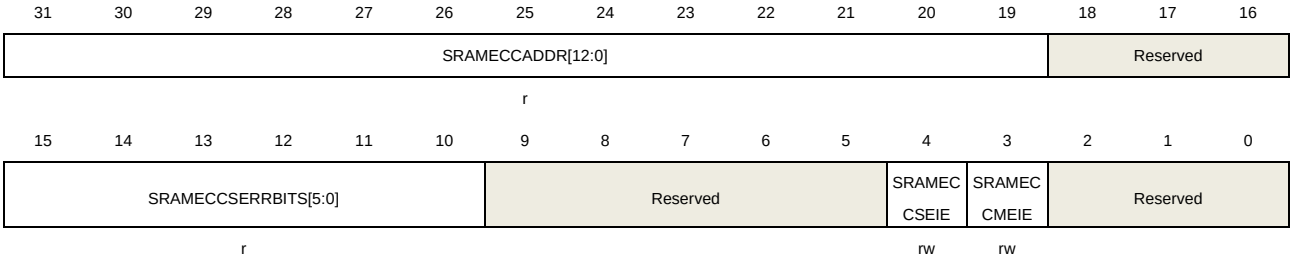
		1: SRAM single bit correction event is detected.
0	SRAMECCMEIF	Indicates the multi-bit error SRAM non-correction event The software can clear it by writing 1. 0: no SRAM non-correction event is detected. 1: SRAM non-correction event is detected.

1.5.9. SRAM ECC control and status register (SYSCFG_SRAMECCCS)

Address offset: 0x30

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:19	SRAMECCADDR[12:0]	Record the faulting system address where the last ECC event on SRAM occurred.
18:16	Reserved	Must be kept at reset value.
15:10	SRAMECCSERRBITS[5:0]	Indicates the error bit Which one bit has an SRAM ECC single-bit correctable error 000000: no error 000001: bit 0 … 111001: bit 38
9:5	Reserved	Must be kept at reset value.
4	SRAMECCSEIE	SRAM single bit correction interrupt enable 0: Disable. 1: Enable.
3	SRAMECCMEIE	SRAM multi-bit non-correction interrupt enable 0: Disable. 1: Enable.
2:0	Reserved	Must be kept at reset value.

1.5.10. Bus timeout status register (SYSCFG_BUSTOSTAT)

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
Reserved												DMA1BUSTOF	DMA0BUSTOF	CPUSBUSTOF	CPUCBUSTOF
												rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	DMA1BUSTOF	DMA1 bus timeout flag. 0: DMA1 bus not timeout 1: DMA1 bus timeout
2	DMA0BUSTOF	DMA0 bus timeout flag. 0: DMA0 bus not timeout 1: DMA0 bus timeout
1	CPUSBUSTOF	CPU Sbus timeout flag. 0: CPU Sbus not timeout 1: CPU Sbus timeout
0	CPUCBUSTOF	CPU Cbus timeout flag. 0: CPU Cbus not timeout 1: CPU Cbus timeout

1.5.11. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 7)

Address offset: 0x100 for TIMER0

Address offset: 0x13C for TIMER7

Reset value: 0x0000 0000

TSCFG0[4:0]…TSCFG6[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				

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

Reserved	TSCFG2[4:0]	TSCFG1[4:0]	TSCFG0[4:0]
	rw	rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: Reserved 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) Others: Reserved
20:16	TSCFG3[4:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3)

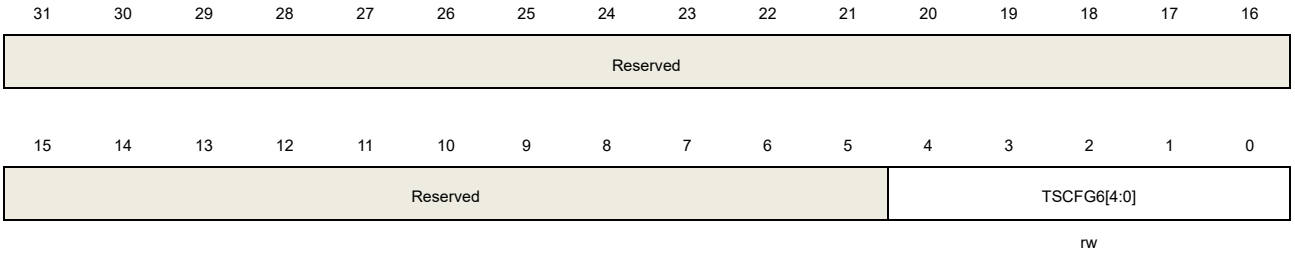
		00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) Others: Reserved
15	Reserved	Must be kept at reset value.
14:10	TSCFG2[4:0]	Quadrature decoder mode 2 configuration 00000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on each other
9:5	TSCFG1[4:0]	Quadrature decoder mode 1 configuration 00000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level
4:0	TSCFG0[4:0]	Quadrature decoder mode 0 configuration 00000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

1.5.12. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 7)

Address offset: 0x104 for TIMER0
Address offset: 0x140 for TIMER7
Reset value: 0x0000 0000

TSCFG0[4:0]…TSCFG6[4:0] are mutually exclusive and cannot be configured at the same time.

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



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	TSCFG6[4:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger when these bits are

not 0.
00000: External clock mode 0 disable
00001: Internal trigger input 0 (ITI0)
00010: Internal trigger input 1 (ITI1)
00011: Internal trigger input 2 (ITI2)
00100: Internal trigger input 3 (ITI3)
00101: CI0 edge flag (CI0F_ED)
00110: The filtered output of channel 0 input (CI0FE0)
00111: The filtered output of channel 1 input (CI1FE1)
01000: The filtered output of external trigger input (ETIFP)
01001: The filtered output of channel 2 input (CI2FE2)
01010: The filtered output of channel 3 input (CI3FE3)
Others: Reserved

1.5.13. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=0, 7)

Address offset: 0x108 for TIMER0
Address offset: 0x144 for TIMER7
Reset value: 0x0000 0000

TSCFG0[4:0]…TSCFG6[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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											TSCFG15[4:0]				
													rw		

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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Internal trigger input 0 (ITI0) 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) Others: Reserved **Note:** When TSCFG15[4:0] is used, TSCFGy[4:0](y=0…6) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on

		TSCFGy[4:0].
15:0	Reserved	Must be kept at reset value.

1.5.14. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=1, 2, 3, 4)

Address offset: 0x10C for TIMER1
Address offset: 0x118 for TIMER2
Address offset: 0x124 for TIMER3
Address offset: 0x130 for TIMER4
Reset value: 0x0000 0000

TSCFG0[4:0]…TSCFG6[4:0], TSCFG9[4:0]…TSCFG15[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TSCFG2[4:0]					TSCFG1[4:0]					TSCFG0[4:0]				
	rw					rw					rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0)

		00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: Reserved 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) Others: Reserved
20:16	TSCFG3[4:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01000: The filtered output of external trigger input (ETIFP) 01001: The filtered output of channel 2 input (CI2FE2) 01010: The filtered output of channel 3 input (CI3FE3) Others: Reserved
15	Reserved	Must be kept at reset value.
14:10	TSCFG2[4:0]	Quadrature decoder mode 2 configuration 00000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on each other
9:5	TSCFG1[4:0]	Quadrature decoder mode 1 configuration 00000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level
4:0	TSCFG0[4:0]	Quadrature decoder mode 0 configuration 00000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

1.5.15. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=1, 2, 3, 4)

Address offset: 0x110 for TIMER1

Address offset: 0x11C for TIMER2
Address offset: 0x128 for TIMER3
Address offset: 0x134 for TIMER4
Reset value: 0x0000 0000

TSCFG0[4:0]…TSCFG6[4:0], TSCFG9[4:0]…TSCFG15[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
	rw					rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TSCFG6[4:0]				
											rw				

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG11[4:0]	Non-quadrature decoder mode 2 configuration 00000: disable Others: The counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.
25:21	TSCFG10[4:0]	Non-quadrature decoder mode 1 configuration 00000: disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. And the counter counts on the edge of CI1FE1, which is set by the CH1P.
20:16	TSCFG9[4:0]	Non-quadrature decoder mode 0 configuration 00000: Disable Others: The CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse. The counter will count on both rising and falling edges of the CI1FE1 signal.
15:5	Reserved	Must be kept at reset value.
4:0	TSCFG6[4:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger when these bits are not 0. 00000: External clock mode 0 disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED)

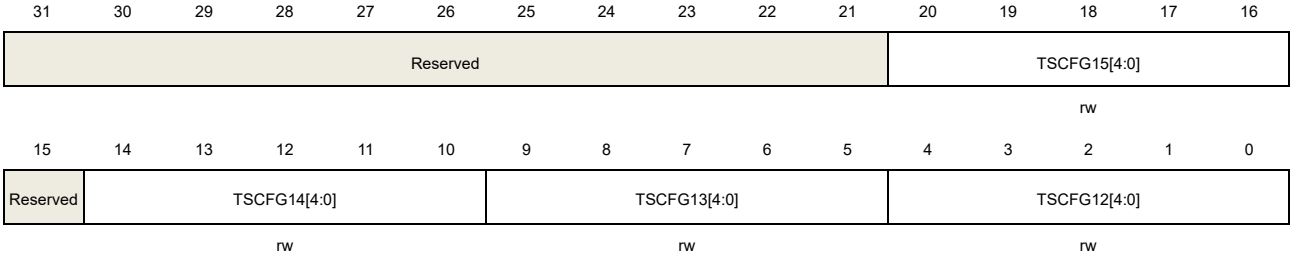
00110: The filtered output of channel 0 input (CI0FE0)
00111: The filtered output of channel 1 input (CI1FE1)
01000: The filtered output of external trigger input (ETIFP)
01001: The filtered output of channel 2 input (CI2FE2)
01010: The filtered output of channel 3 input (CI3FE3)
Others: Reserved

1.5.16. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=1, 2, 3, 4)

Address offset: 0x114 for TIMER1
Address offset: 0x120 for TIMER2
Address offset: 0x12C for TIMER3
Address offset: 0x138 for TIMER4
Reset value: 0x0000 0000

TSCFG0[4:0]…TSCFG6[4:0], TSCFG9[4:0]…TSCFG15[4:0] are mutually exclusive and cannot be configured at the same time.

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Internal trigger input 0 (ITI0) 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) Others: Reserved **Note:** When TSCFG15[4:0] is used, TSCFGy[4:0](y=0…6, 9…15) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on TSCFGy[4:0].
15	Reserved	Must be kept at reset value.
14:10	TSCFG14[4:0]	Quadrature decoder mode 4 configuration 00000: Disable Others: The counter counts on CI1FE1 edge only, while the direction depends on

		CI0FE0 level.
9:5	TSCFG13[4:0]	Quadrature decoder mode 3 configuration 00000: Disable Others: The counter counts on CI0FE0 edge only, while the direction depends on CI1FE1 level.
4:0	TSCFG12[4:0]	Non-quadrature decoder mode 3 configuration 00000: disable Others: the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the CIxFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the CIxFEx signal.

1.5.17. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=15, 16)

Address offset: 0x148 for TIMER15
Address offset: 0x154 for TIMER16
Reset value: 0x0000 0000

TSCFG3[4:0]...TSCFG6[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				

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

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01011: MCI0FEM0 10011: Internal trigger input 14 (ITI14)

		Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: Reserved 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01011: MCI0FEM0 10011: Internal trigger input 14 (ITI14) Others: Reserved
20:16	TSCFG3[4:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 00000: Restart mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01011: MCI0FEM0 10011: Internal trigger input 14 (ITI14) Others: Reserved
15:0	Reserved	Must be kept at reset value.

1.5.18. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=15, 16)

Address offset: 0x14C for TIMER15
Address offset: 0x158 for TIMER16
Reset value: 0x0000 0000

TSCFG3[4:0]...TSCFG6[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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											TSCFG6[4:0]				
											rw				

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	TSCFG6[4:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger when these bits are not 0. 00000: External clock mode 0 disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01011: MCI0FEM0 10011: Internal trigger input 14 (ITI14) Others: Reserved

1.5.19. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2, x=15, 16)

Address offset: 0x150 for TIMER15
Address offset: 0x15C for TIMER16
Reset value: 0x0000 0000

TSCFG3[4:0]...TSCFG6[4:0] are mutually exclusive and cannot be configured at the same time.

This register 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											TSCFG15[4:0]				
											rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Internal trigger input 0 (ITI0)

00001: Internal trigger input 0 (ITI0)
00010: Internal trigger input 1 (ITI1)
00011: Internal trigger input 2 (ITI2)
00100: Internal trigger input 3 (ITI3)
00101: CI0 edge flag (CI0F_ED)
10011: Internal trigger input 14 (ITI14)
Others: Reserved
Note: When TSCFG15[4:0] is used, TSCFGy[4:0](y = 3..6) should be zero, otherwise, the ITS trigger cooperate with TSCFGy[4:0], and input source depend on TSCFGy[4:0].

15:0	Reserved	Must be kept at reset value.

1.6. Device electronic signature

The device electronic signature contains memory size 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.

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

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

Bits	Fields	Descriptions
31:16	SRAM_DENSITY [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 F7E8

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

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

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

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

Base address: 0x1FFF F7EC

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

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

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

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

Base address: 0x1FFF F7F0

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

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. Interrupt / event controller (EXTI)

2.1. Overview

Cortex®-M33 integrates the Nested Vectored Interrupt Controller (NVIC) for efficient exception and interrupts processing. NVIC facilitates low-latency exception and interrupt handling and power management controls. It's tightly coupled to the processer core. More details about NVIC could referred to the technical reference manual of cortex®-M33.

EXTI (interrupt / event controller) contains up to 20 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.

2.2. Characteristics

- Cortex®-M33 system exception.
- Up to 79 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 20 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

2.3. Interrupts function overview

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

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

Table 2-1. NVIC exception types in Cortex®-M33

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

Table 2-2. Interrupt vector table

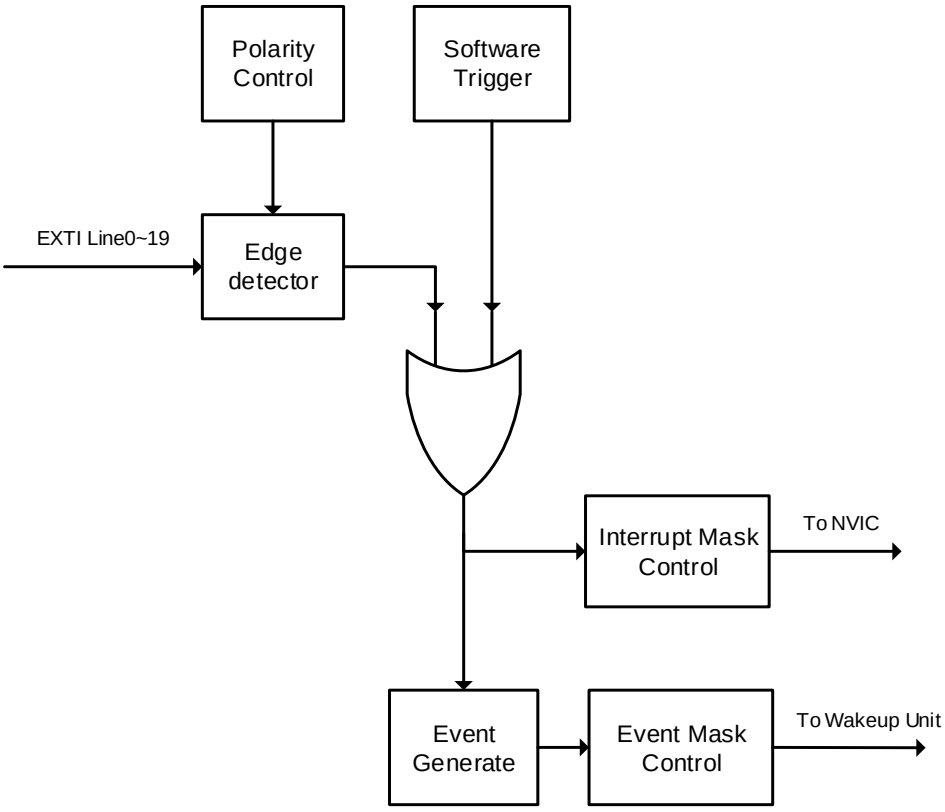
Interrupt Number	Vector Number	Devices Interrupt Description	Vector Address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	LVD / VAVD from EXTI interrupt	0x0000_0044
IRQ 2	18	Tamper interrupt	0x0000_0048
IRQ 3	19	RTC global interrupt	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU and CTC interrupt	0x0000_0054
IRQ 6	22	EXTI line0 interrupt	0x0000_0058
IRQ 7	23	EXTI line1 interrupt	0x0000_005C
IRQ 8	24	EXTI line2 interrupt	0x0000_0060
IRQ 9	25	EXTI line3 interrupt	0x0000_0064
IRQ 10	26	EXTI line4 interrupt	0x0000_0068
IRQ 11	27	DMA0 channel0 global interrupt	0x0000_006C
IRQ 12	28	DMA0 channel1 global interrupt	0x0000_0070
IRQ 13	29	DMA0 channel2 global interrupt	0x0000_0074
IRQ 14	30	DMA0 channel3 global interrupt	0x0000_0078
IRQ 15	31	DMA0 channel4 global interrupt	0x0000_007C
IRQ 16	32	DMA0 channel5 global interrupt	0x0000_0080
IRQ 17	33	DMA0 channel6 global interrupt	0x0000_0084

Interrupt Number	Vector Number	Devices Interrupt Description	Vector Address
IRQ 18	34	ADC0 and ADC1 global interrupt	0x0000_0088
IRQ 19	35	CAN0 TX interrupts	0x0000_008C
IRQ 20	36	CAN0 RX0 interrupts	0x0000_0090
IRQ 21	37	CAN0 RX1 interrupts	0x0000_0094
IRQ 22	38	CAN0 EWMC interrupts	0x0000_0098
IRQ 23	39	EXTI line[9:5] interrupts	0x0000_009C
IRQ 24	40	TIMER0 break interrupt	0x0000_00A0
IRQ 25	41	TIMER0 update interrupt	0x0000_00A4
IRQ 26	42	TIMER0 trigger and Channel commutation interrupts	0x0000_00A8
IRQ 27	43	TIMER0 channel capture compare interrupt	0x0000_00AC
IRQ 28	44	TIMER1 global interrupt	0x0000_00B0
IRQ 29	45	TIMER2 global interrupt	0x0000_00B4
IRQ 30	46	TIMER3 global interrupt	0x0000_00B8
IRQ 31	47	I2C0 event interrupt	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1 global interrupt	0x0000_00D0
IRQ 37	53	USART0 global interrupt	0x0000_00D4
IRQ 38	54	USART1 global interrupt	0x0000_00D8
IRQ 39	55	USART2 global interrupt	0x0000_00DC
IRQ 40	56	EXTI line[15:10] interrupts	0x0000_00E0
IRQ 41	57	RTC alarm from EXTI interrupt	0x0000_00E4
IRQ 42	58	USBFS wakeup from EXTI interrupt	0x0000_00E8
IRQ 43	59	TIMER7 break interrupt	0x0000_00EC
IRQ 44	60	TIMER7 update interrupt	0x0000_00F0
IRQ 45	61	TIMER7 trigger and Channel commutation interrupts	0x0000_00F4
IRQ 46	62	TIMER7 channel capture compare interrupt	0x0000_00F8
IRQ 47	63	ADC2 global interrupt	0x0000_00FC
IRQ 48	64	reserved	0x0000_0100
IRQ 49	65	RCU clock frequency monitor interrupt	0x0000_0104
IRQ50	66	CMP from EXTI interrupt	0x0000_0108
IRQ51	67	SPI2 global interrupt	0x0000_010C
IRQ52	68	UART3 global interrupt	0x0000_0110
IRQ53	69	UART4 global interrupt	0x0000_0114
IRQ54	70	TIMER5 global interrupt	0x0000_0118

Interrupt Number	Vector Number	Devices Interrupt Description	Vector Address
IRQ55	71	TIMER6 global interrupt	0x0000_011C
IRQ56	72	DMA1 channel0 global interrupt	0x0000_0120
IRQ57	73	DMA1 channel1 global interrupt	0x0000_0124
IRQ58	74	DMA1 channel2 global interrupt	0x0000_0128
IRQ59	75	DMA1 channel3 global interrupt	0x0000_012C
IRQ60	76	DMA1 channel4 global interrupt	0x0000_0130
IRQ61	77	DAC global interrupt	0x0000_0134
IRQ62	78	VUVD or VOVD interrupt	0x0000_0138
IRQ63	79	CAN1 TX interrupt	0x0000_013C
IRQ64	80	CAN1 RX0 interrupt	0x0000_0140
IRQ65	81	CAN1 RX1 interrupt	0x0000_0144
IRQ66	82	CAN1 EWMC interrupt	0x0000_0148
IRQ67	83	SRAM ECC interrupt	0x0000_014C
IRQ68	84	FPU interrupt	0x0000_0150
IRQ69	85	CMP interrupt	0x0000_0154
IRQ70	86	DMAMUX interrupt	0x0000_0158
IRQ71	87	CAU interrupt	0x0000_015C
IRQ72	88	HAU interrupt	0x0000_0160
IRQ73	89	TRNG interrupt	0x0000_0164
IRQ74	90	USBFS interrupt	0x0000_0168
IRQ75	91	TIMER4 interrupt	0x0000_016C
IRQ76	92	TIMER15 interrupt	0x0000_0170
IRQ77	93	TIMER16 interrupt	0x0000_0174
IRQ78	94	TIMER0 channel break interrupt	0x0000_0178
IRQ79	95	TIMER7 channel break interrupt	0x0000_017C

2.4. External interrupt and event (EXTI) block diagram

Figure 2-1. Block diagram of EXTI



2.5. External Interrupt and Event function overview

The EXTI contains up to 20 independent edge detectors and generates interrupts request or event 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 4 lines from internal modules which refers to ***Table 2-3. EXTI source***. All GPIO pins can be selected as an EXTI trigger source by configuring AFIO_EXTISSx registers in GPIO module (please refer to ***General-purpose and alternate-function I/Os (GPIO and AFIO)*** section for detail).

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

Hardware trigger

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

1. Configure EXTI sources in AFIO 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 2-3. EXTI source

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

EXTI Line Number	Source
18	CMP Wakeup
19	USBFS Wakeup

2.6. EXTI Register

EXTI base address: 0x4001 0400

2.6.1. Interrupt enable register (EXTI_INTEN)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												INTEN19	INTEN18	INTEN17	INTEN16
												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
31:20	Reserved	Must be kept at reset value.
19: 0	INTENx	Interrupt enablebit x (x = 0…19) 0: Interrupt from linex is disabled. 1: Interrupt from linex is enabled.

2.6.2. 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												EVEN19	EVEN18	EVEN17	EVEN16
												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
31:20	Reserved	Must be kept at reset value.
19: 0	EVENx	Event enable bit x (x = 0…19) 0: Event from linex is disabled. 1: Event from linex is enabled.

2.6.3. 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												RTEN19	RTEN18	RTEN17	RTEN16
												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
31:20	Reserved	Must be kept at reset value.
19:0	RTENx	Rising edge trigger enable x (x = 0…19) 0: Rising edge of linex is invalid 1: Rising edge of linex is valid as an interrupt / event request

2.6.4. 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												FTEN19	FTEN18	FTEN17	FTEN16
												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: 20	Reserved	Must be kept at reset value.
19: 0	FTENx	Falling edge trigger enable x (x = 0…19) 0: Falling edge of linex is invalid 1: Falling edge of linex is valid as an interrupt / event request

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

2.6.6. Pending register (EXTI_PD)

Address offset: 0x14

Reset value: 0xXXXX XXXX, where X is undefined.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												PD19	PD18	PD17	PD16
												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: 20	Reserved	Must be kept at reset value.
19: 0	PDx	Interrupt pending status x (x = 0…19) 0: EXTI linex is not triggered 1: EXTI linex is triggered This bit is cleared to 0 by writing 1 to it.

3. Direct memory access controller (DMA)

3.1. Overview

The direct memory access (DMA) controller provides a hardware method of transferring data between peripherals and/or memory without intervention from the CPU, thereby freeing up bandwidth for other system functions. Data can be quickly moved by DMA between peripherals and memory as well as memory and memory without any CPU actions. There are 12 channels in the DMA controller (7 for DMA0 and 5 for DMA1). Each channel is dedicated to manage memory access requests from one or more peripherals. An arbiter is implemented inside to handle the priority among DMA requests.

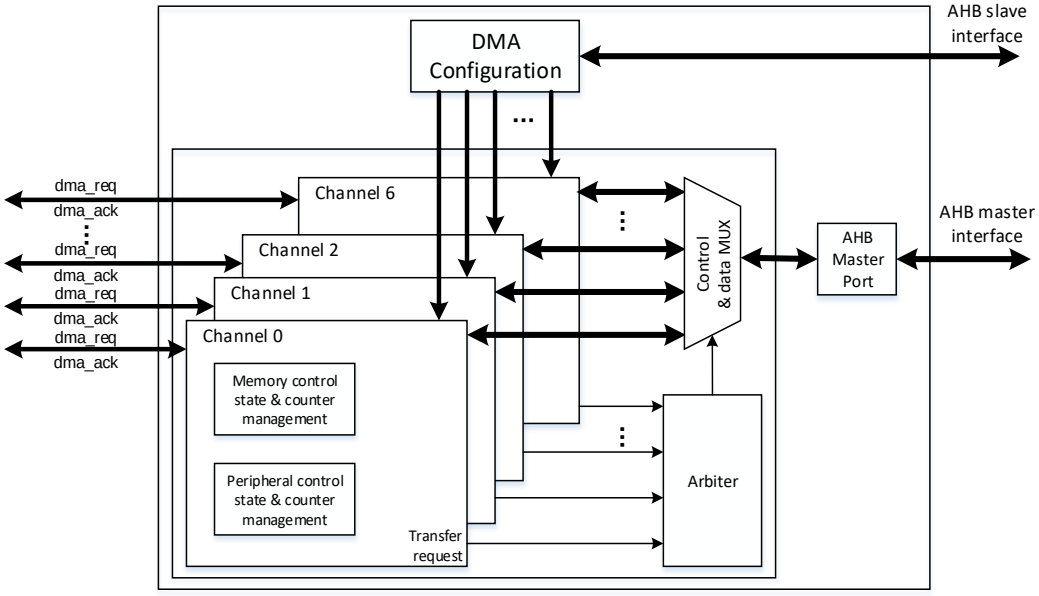
The system bus is shared by the DMA controller and the Cortex®-M33 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system bus bandwidth for the CPU.

3.2. Characteristics

- Programmable length of data to be transferred, max to 65536.
- 12 channels (7 for DMA0 and 5 for DMA1) and each channel are configurable.
- AHB and APB peripherals, FLASH, SRAM can be accessed as source and destination.
- Each channel is connected to fixed hardware DMA request.
- Software DMA channel priority (low, medium, high, ultra high) and hardware DMA channel priority (DMA channel 0 has the highest priority and DMA channel 6 has the lowest priority).
- Support independent 8, 16, 32-bit memory and peripheral transfer.
- Support independent fixed and increasing address generation algorithm of memory and peripheral.
- Support circular transfer mode.
- Support peripheral to memory, memory to peripheral, and memory to memory transfers.
- One separate interrupt per channel with three types of event flags.
- Support interrupt enable and clear.

3.3. Block diagram

Figure 3-1. Block diagram of DMA



As shown in ***Figure 3-1. 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.

3.4. Function overview

3.4.1. 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 3-1. DMA transfer operation***.

Table 3-1. DMA transfer operation

Transfer size		Transfer operations	
Source	**Destination**	**Source**	**Destination**
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

The CNT bits in the DMA_CHxCNT register control how many data to be transmitted on the channel and must be configured before enable the CHEN bit in the register. During the transmission, the CNT bits indicate the remaining number of data items to be transferred.

The DMA transmission is disabled by clearing the CHEN bit in the DMA_CHxCTL register.

- If the DMA transmission is not completed when the CHEN bit is cleared, two situations may be occurred when restart this DMA channel:
  - If no register configuration operations of the channel occurs before restart the DMA channel, the DMA will continue to complete the rest of the transmission.
  - If any register configuration operations to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel occur, the DMA will restart a new transmission.
- If the DMA transmission has been finished when clearing the CHEN bit, enable the DMA channel without any register configuration operation to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel will not launch any DMA transfer.

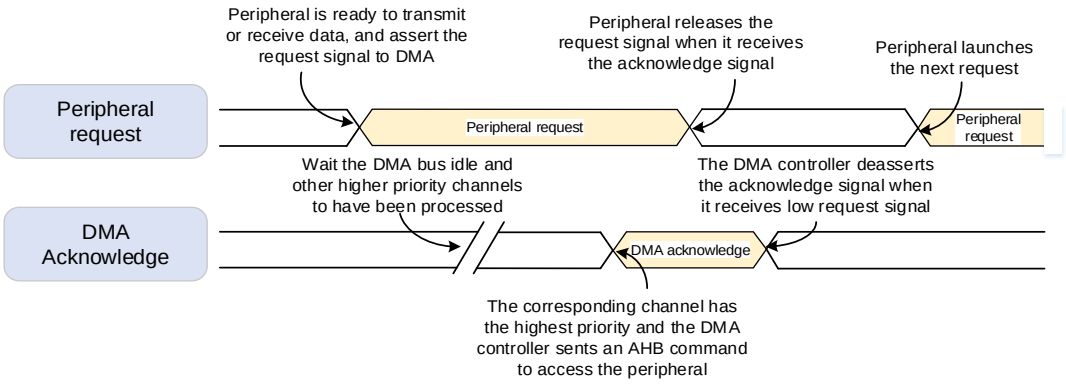
3.4.2. Peripheral handshake

To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and an acknowledge signal:

- Request signal asserted by peripheral to DMA controller, indicating that the peripheral is ready to transmit or receive data.
- Acknowledge signal responded by DMA to peripheral, indicating that the DMA controller has initiated an AHB command to access the peripheral.

Figure 3-2. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 3-2. Handshake mechanism



3.4.3. 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.

3.4.4. 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.

3.4.5. 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.

3.4.6. 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.

3.4.7. 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.

3.4.8. 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.

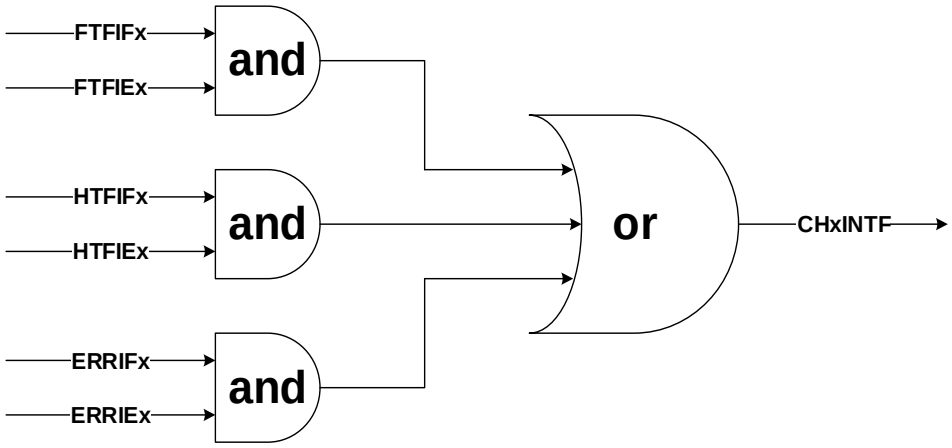
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following ***Table 3-2. interrupt events***.

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

Figure 3-3. DMA interrupt logic



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

3.4.9. DMA request mapping

The DMA requests of a channel are coming from the AHB/APB peripherals through the corresponding channel output of DMAMUX request multiplexer, refer to ***Table 4-3. Request***

multiplexer input mapping.

3.5. Register definition

DMA base address: 0x4002 0000

DMA1 base address: 0x4002 0400

Note: For DMA1 having 5 channels, all bits related to channel 5 and channel 6 in the following registers are not suitable for DMA1

3.5.1. 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

3.5.2. Interrupt flag clear register (DMA_INTC)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
27/23/19/15 /11/7/3	ERRIFCx	Clear bit for error flag of channel x (x=0…6) 0: No effect 1: Clear error flag
26/22/18/14 /10/6/2	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0…6) 0: No effect 1: Clear half transfer finish flag
25/21/17/13 /9/5/1	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=0…6) 0: No effect 1: Clear full transfer finish flag
24/20/16/12 /8/4/0	GIFCx	Clear global interrupt flag of channel x (x=0…6) 0: No effect 1: Clear GIFx, ERRIFx, HTFIFx and FTFIFx bits in the DMA_INTF register

3.5.3. 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

3.5.4. 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.

3.5.5. 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.

3.5.6. 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.

4. DMA request multiplexer (DMAMUX)

4.1. Overview

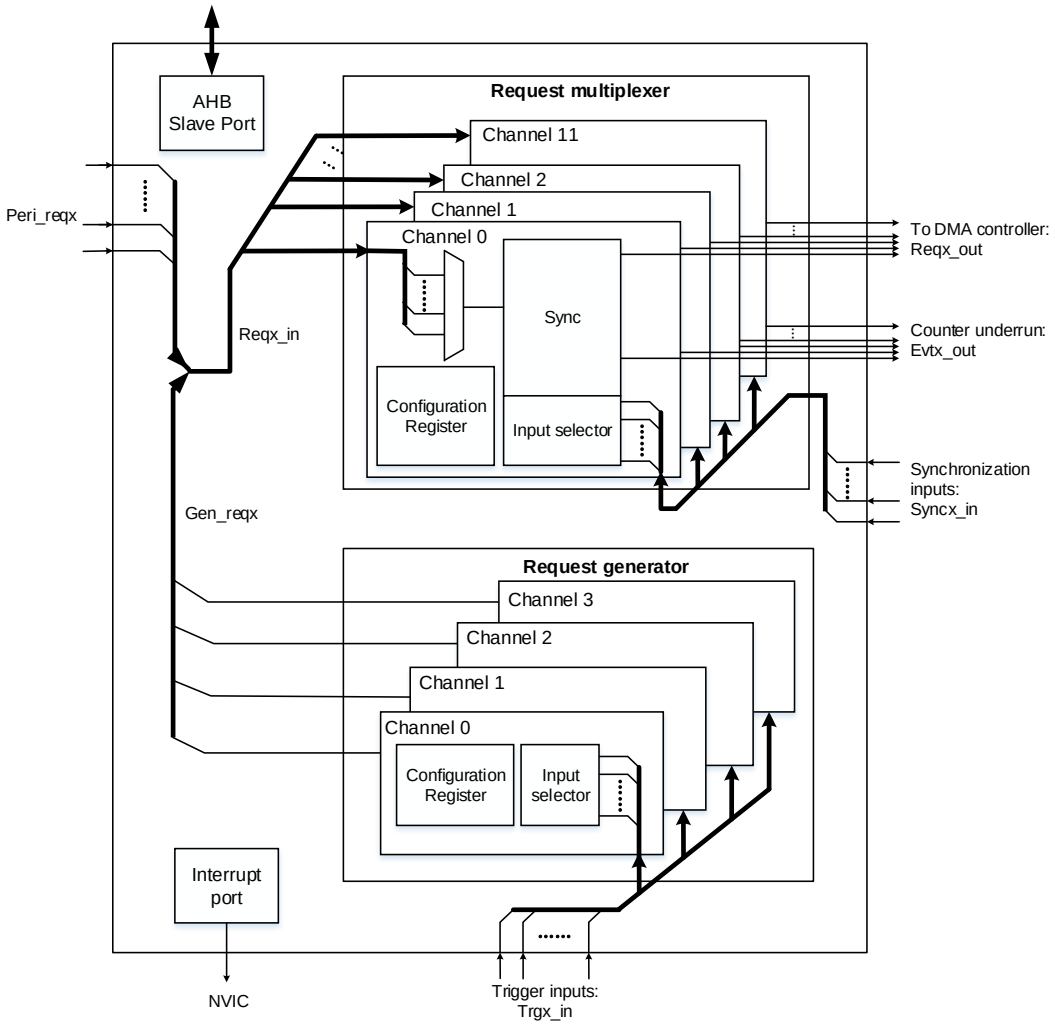
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, unconditionally or synchronously with events from its DMAMUX synchronization inputs. The DMA request is pending until it is served by the DMA controller which generates a DMA acknowledge signal (the DMA request signal is de-asserted).

4.2. Characteristics

- 12 channels for DMAMUX request multiplexer.
- 4 channels for DMAMUX request generator.
- Support 20 trigger inputs.
- Support 20 synchronization inputs.
- Each DMAMUX request generator channel:
  - DMA request trigger input selector
  - DMAMUX request generator counter
  - Trigger overrun flag
- Each DMAMUX request multiplexer channel:
  - 82 input DMA request lines from peripherals
  - Synchronization input selector
  - One DMA request line output
  - One channel event output, for DMA request chaining
  - DMAMUX request multiplexer counter
  - Synchronization overrun flag

4.3. Block diagram

Figure 4-1. Block diagram of DMAMUX



4.4. Function overview

As shown in ***Figure 4-1. Block diagram of DMAMUX***, DMAMUX includes two sub-blocks:

- DMAMUX request multiplexer.

DMAMUX request multiplexer inputs (Reqx_in) source from:

- Peripherals (Peri_reqx).
- DMAMUX request generator outputs (Gen_reqx).

DMAMUX request multiplexer outputs (Reqx_out) is connected to channels of DMA controller.

Synchronization inputs (Syncx_in) source from internal or external signals.

- DMAMUX request generator.

Trigger inputs (Trgx_in) source from internal or external signals.

4.4.1. DMAMUX signals

Table 4-1. DMAMUX signals

Signal name	Discription
Reqx_in	DMAMUX request multiplexer inputs (from peripheral requests and request generator channels)
Peri_reqx	DMAMUX DMA request line inputs from peripherals
Gen_reqx	DMAMUX generated DMA request from request generator
Reqx_out	DMAMUX requests outputs (to DMA controller)
Trgx_in	DMAMUX DMA request triggers inputs (to request generator)
Syncx_in	DMAMUX synchronization inputs (to request multiplexer)
Evtx_out	DMAMUX request multiplexer counter underrun event outputs

4.4.2. DMAMUX request multiplexer

The DMAMUX request multiplexer enables routing a DMA request line between the peripherals / generated DMA request and the DMA controllers of the product. Its component unit is the request multiplexer channels. DMA request lines are connected in parallel to all request multiplexer channels. There is a synchronization unit for each request multiplexer channel. The synchronization inputs are connected in parallel to all synchronization unit of request multiplexer channels. And there is a built-in DMAMUX request multiplexer counter for each request multiplexer channel.

Request multiplexer channel

A DMA request input for the DMAMUX request multiplexer channel x is configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 4-3. Request multiplexer input mapping***. A DMAMUX request multiplexer channel is connected and dedicated to one single channel of the DMA controller.

Note: The value 0 of MUXID[6:0] bits corresponds to no DMA request line is selected. It is not allowed to configure the same DMA request line (same non-null MUXID[6:0]) to two different request multiplexer channels.

When synchronization mode is disabled

Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register. If the channel event generation is enabled by setting EVGEN bit, the number of DMA requests before an output event generation is NBR[4:0] + 1.

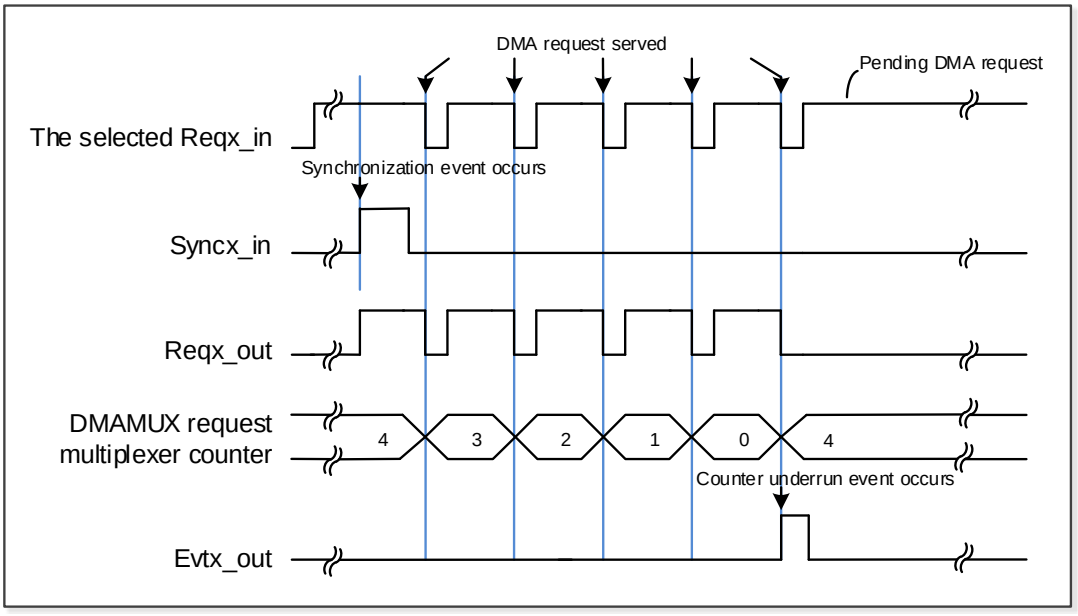
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

When synchronization mode is enabled

A channel x in synchronization mode, when a rising/falling edge on the selected synchronization input is detected, the pending selected input DMA request line is routed to the multiplexer channel x output. Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the input DMA request line is disconnected from the request multiplexer channel x output, and the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register. The number of DMA requests transferred to the request multiplexer channel x output following a detected synchronization event is NBR[4:0] + 1.

Figure 4-2. Synchronization mode shows an example when NBR[4:0]=4, SYNCEN=1, EVGEN=1, SYNCP[1:0]=01.

Figure 4-2. Synchronization mode



DMAMUX request multiplexer channel x can be synchronized by setting the synchronization enable bit SYNCEN in the DMAMUX_RM_CHxCFG register. The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to *Table 4-5. Synchronization input mapping*. The synchronization input valid edge is configured by the SYNCP[1:0] bits of the DMAMUX_RM_CHxCFG register.

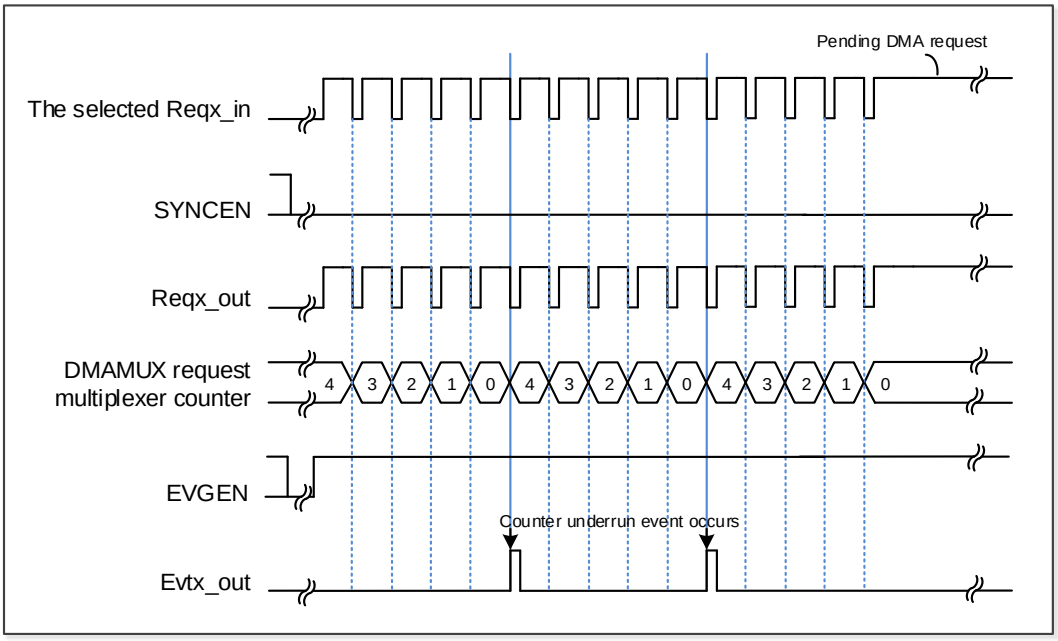
Note: If a synchronization input event occurs when there is no pending selected input DMA request line, the input event is discarded. The following asserted input request lines will not be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

Channel event generation

Each DMA request line multiplexer channel has an event output called Evtx_out, which is the DMA request multiplexer counter underrun event. Signals dmamux_evt0 ~ dmamux_evt3 can be used for DMA request chaining. If event generation bit EVGEN in the DMAMUX_RM_CHxCFG register is enabled on the channel x output, when its DMA request multiplexer counter is automatically reloaded with the value of the programmed NBR[4:0] field, the multiplexer channel generates a channel event, as a pulse of one AHB clock cycle.

Figure 4-3. Event generation shows an example when NBR[4:0]=4, SYNCEN=0, EVGEN=1.

Figure 4-3. Event generation



Note: If EVGEN = 1 and NBR[4:0] = 0, an event is generated after each served DMA request.

Synchronization overrun

If a new synchronization event occurs before the built-in DMAMUX request multiplexer counter underrun, the synchronization overrun flag bit SOIFx is set in the DMAMUX_RM_INTF register.

Note: The synchronization mode of request multiplexer channel x shall be disabled by resetting SYNCEN bit in DMAMUX_RM_CHxCFG register at the completion of the use of the related channel of the DMA controller. Otherwise, when a new synchronization event occurs, there will be a synchronization overrun due to the absence of a DMA acknowledge (that is, no served request) received from the DMA controller.

4.4.3. DMAMUX request generator

The DMAMUX request generator produces DMA requests upon trigger input event. Its component unit is the request generator channels. DMA request trigger inputs are connected

in parallel to all request generator channels. And there is a built-in DMAMUX request generator counter for each request generator channel.

The active edge of trigger input events is selected through the RGTP[1:0] bits in DMAMUX_RG_CHxCFG register. The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 4-4. Trigger input mapping***. DMAMUX request generator channel x can be enabled by setting RGEN to 1 in DMAMUX_RG_CHxCFG register.

Request generator channel

Upon the trigger input event, the corresponding request generator channel starts generating DMA requests on its output, and the output goes to the input of the DMAMUX request multiplexer. Each time the DMAMUX generated request is served by the connected DMA controller, the served request will be de-asserted, and the built-in DMAMUX request generator counter of the request generator channel is decremented. At the request generator counter underrun, the request generator channel stops generating DMA requests. The built-in DMAMUX request generator counter will be automatically reloaded to its programmed value upon the next trigger input event, the built-in counter is programmed by the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register.

Note: The number of generated DMA requests after the trigger input event is NBRG[4:0] + 1. The NBRG[4:0] value shall only be written by software when the RGEN bit of the corresponding generator channel x is disabled.

Trigger overrun

If a request generator channel x was enabled by RGEN bit, when a new DMA request trigger event for the request generator channel x occurs before the DMAMUX request generator counter underrun, then the request trigger overrun event flag bit TOIFx is set by hardware in the DMAMUX_RG_INTF register.

Note: The request generator channel x shall be disabled by resetting RGEN bit in DMAMUX_RG_CHxCFG register at the completion of the usage of the related channel of the DMA controller. Otherwise, when a new detected trigger event occurs, there will be a trigger overrun due to the absence of an acknowledge (that is, no served request) received from the DMA.

4.4.4. Channel configurations

The following sequence should be followed to configure a DMAMUX channel y and the related DMA channel x:

1. Set and configure the DMA channel x completely, except enabling the channel x.
2. Set and configure the related DMAMUX channel y completely.
3. Configure the CHEN bit with ‘1’ in the DMA_CHxCTL register to enable the DMA channel x.

4.4.5. Interrupt

There are two types of interrupt event, including synchronization overrun event on each DMAMUX request multiplexer channel, and trigger overrun event on each DMAMUX request generator channel.

Each interrupt event has a dedicated flag bit, a dedicated clear bit, and a dedicated enable bit. The relationship is described in the following ***Table 4-2. Interrupt events***.

Table 4-2. Interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
Synchronization overrun event on DMAMUX request multiplexer channel x	SOIFx	SOIFCx	SOIE
Trigger overrun event on DMAMUX request generator channel y	TOIFy	TOIFCy	TOIE

Trigger overrun interrupt

When the DMAMUX request trigger overrun flag TOIFx is set, and the trigger overrun interrupt is enabled by setting TOIE bit, a trigger overrun interrupt will be generated. The overrun flag TOIFx is reset by writing 1 to the corresponding clear bit of overrun flag TOIFCx in the DMAMUX_RG_INTC register.

Synchronization overrun interrupt

When the synchronization overrun flag SOIFx is set, and the synchronization overrun interrupt is enabled by setting SOIE bit, a synchronization overrun interrupt will be generated. The overrun flag SOIFx is reset by writing 1 to the corresponding clear bit of synchronization overrun flag bit SOIFCx in the DMAMUX_RM_INTC register.

4.4.6. DMAMUX mapping

The DMAMUX works in conjunction with the DMA. Channels 0 to 2 of the DMAMUX are connected to channels 0 to 2 of the DMA.

Request multiplexer input mapping

A DMA request is sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 4-3. Request multiplexer input mapping***, configured by the MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register for the DMAMUX request multiplexer channel x.

Table 4-3. Request multiplexer input mapping

Request multiplexer channel input identification MUXID[6:0]	Source
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC0_ROUTINE
6	ADC0_INSERTED
7	DAC0_CH0
8	TIMER5_UP
9	TIMER6_UP
10	SPI0_RX
11	SPI0_TX
12	SPI1_RX
13	SPI1_TX
14	SPI2_RX
15	SPI2_TX
16	I2C0_RX
17	I2C0_TX
18	I2C1_RX
19	I2C1_TX
20	USART0_RX
21	USART0_TX
22	USART1_RX
23	USART1_TX
24	USART2_RX
25	USART2_TX
26	UART3_RX
27	UART3_TX
28	TIMER0_CH0
29	TIMER0_CH1
30	TIMER0_CH2
31	TIMER0_CH3
32	TIMER0_UP
33	TIMER0_TRG
34	TIMER0_CMT
35	TIMER7_CH0
36	TIMER7_CH1
37	TIMER7_CH2
38	TIMER7_CH3
39	TIMER7_UP

Request multiplexer channel input identification MUXID[6:0]	Source
40	TIMER7_TRG
41	TIMER7_CMT
42	TIMER1_CH0
43	TIMER1_CH1
44	TIMER1_CH2
45	TIMER1_CH3
46	TIMER1_UP
47	TIMER1_TRG
48	TIMER2_CH0
49	TIMER2_CH1
50	TIMER2_CH2
51	TIMER2_CH3
52	TIMER2_UP
53	TIMER2_TRG
54	TIMER3_CH0
55	TIMER3_CH1
56	TIMER3_CH2
57	TIMER3_CH3
58	TIMER3_UP
59	TIMER3_TRG
60	TIMER4_CH0
61	TIMER4_CH1
62	TIMER4_CH2
63	TIMER4_CH3
64	TIMER4_UP
65	TIMER4_TRG
66	ADC1_ROUTINE
67	ADC1_INSERTED
68	ADC2_ROUTINE
69	ADC2_INSERTED
70	Reserved
71	Reserved
72	TIMER15_CH0
73	TIMER15_CH1
74	TIMER15_CHLN
75	TIMER15_UP
76	TIMER15_TRG
77	TIMER15_CMT
78	TIMER16_CH0
79	TIMER16_CH1

Request multiplexer channel input identification MUXID[6:0]	Source
80	TIMER16_CHLN
81	TIMER16_UP
82	TIMER16_TRG
83	TIMER16_CMT
84	CAU_IN
85	CAU_OUT
86	HAU_IN

Trigger input mapping

The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 4-4. Trigger input mapping***.

Table 4-4. Trigger input mapping

Trigger input identification TID[4:0]	Source
0	Evtx_out0
1	Evtx_out1
2	Evtx_out2
3	Evtx_out3
4	EXTI_0_INT
5	EXTI_1_INT
6	EXTI_2_INT
7	EXTI_3_INT
8	EXTI_4_INT
9	EXTI_5_INT
10	EXTI_6_INT
11	EXTI_7_INT
12	EXTI_0_EVT
13	EXTI_1_EVT
14	EXTI_2_EVT
15	EXTI_3_EVT
16	EXTI_4_EVT
17	EXTI_5_EVT
18	EXTI_6_EVT
19	EXTI_7_EVT
20	Reserved
21	Reserved
22	Reserved
23	Reserved

Trigger input identification TID[4:0]	Source
24	Reserved
25	Reserved
26	Reserved

Synchronization input mapping

The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to ***Table 4-5. Synchronization input mapping***.

Table 4-5. Synchronization input mapping

Synchronization input identification SYNCID[4:0]	Source
0	Evtx_out0
1	Evtx_out1
2	Evtx_out2
3	Evtx_out3
4	EXTI_0_INT
5	EXTI_1_INT
6	EXTI_2_INT
7	EXTI_3_INT
8	EXTI_4_INT
9	EXTI_5_INT
10	EXTI_6_INT
11	EXTI_7_INT
12	EXTI_0_EVT
13	EXTI_1_EVT
14	EXTI_2_EVT
15	EXTI_3_EVT
16	EXTI_4_EVT
17	EXTI_5_EVT
18	EXTI_6_EVT
19	EXTI_7_EVT
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved

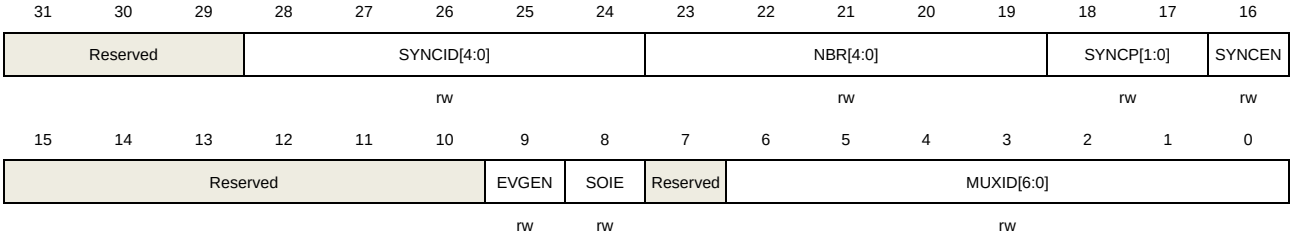
4.5. Register definition

DMAMUX base address: 0x4002 0800

4.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

x = 0...11, where x is a channel number
Address offset: 0x00 + 0x04 × x
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:24	SYNCID[4:0]	Synchronization input identification Selects the synchronization input source.
23:19	NBR[4:0]	Number of DMA requests to forward The number of DMA requests to forward to the DMA controller after a synchronization event / before an output event is generated equals to NBR[4:0] + 1. These bits shall only be written when both SYNCEN and EVGEN bits are disabled.
18:17	SYNCP[1:0]	Synchronization input polarity 00: No event detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	SYNCEN	Synchronization enable 0: Disable synchronization 1: Enable synchronization
15:10	Reserved	Must be kept at reset value.
9	EVGEN	Event generation enable 0: Disable event generation

		1: Enable event generation
8	SOIE	Synchronization overrun interrupt enable 0: Disable interrupt 1: Enable interrupt
7	Reserved	Must be kept at reset value.
6:0	MUXID[6:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

4.5.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF)

Address offset: 0x80
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SOIF11	SOIF10	SOIF9	SOIF8	SOIF7	SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
				r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	SOIFx	Synchronization overrun event flag of request multiplexer channel x (x=0..11) The flag is set when a synchronization event occurs on a DMA request line multiplexer channel x, while the DMA request counter value is lower than NBR[4:0]. The flag is cleared by writing 1 to the corresponding SOIFCx bit in DMAMUX_RM_INTC register.

4.5.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC)

Address offset: 0x084
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SOIFC11	SOIFC10	SOIFC9	SOIFC8	SOIFC7	SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
				w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	SOIFCx	Clear bit for synchronization overrun event flag of request multiplexer channel x (x=0..11) Writing 1 clears the corresponding overrun flag SOIFx in the DMAMUX_RM_INTF register.

4.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

x = 0...3, where x is a channel number

Address offset: 0x100 + 0x04 × x

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								NBRG[4:0]					RGTP[1:0]		RGEN
										rw			rw		rw

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:19	NBRG[4:0]	Number of DMA requests to be generated The number of DMA requests to be generated after a trigger event equals to NBRG[4:0] + 1. **Note:** These bits shall only be written when RGEN bit is disabled.
18:17	RGTP[1:0]	DMA request generator trigger polarity 00: No event trigger detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	RGEN	DMA request generator channel x enable 0: Disable DMA request generator channel x 1: Enable DMA request generator channel x
15:9	Reserved	Must be kept at reset value.
8	TOIE	Trigger overrun interrupt enable 0: Disable interrupt

		1: Enable interrupt
7:5	Reserved	Must be kept at reset value.
4:0	TID[4:0]	Trigger input identification Selects the DMA request trigger input source.

4.5.5. Request generator interrupt flag register (DMAMUX_RG_INTF)

Address offset: 0x140

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TOIF3	TOIF2	TOIF1	TOIF0
												r	r	r	r

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	TOIFx	Trigger overrun event flag of request generator channel x (x=0..3) The flag is set when a new trigger event occurs on DMA request generator channel x, before the request counter underrun (the internal request counter programmed via the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register). The flag is cleared by writing 1 to the corresponding TOIFCx bit in the DMAMUX_RG_INTC register.

4.5.6. Request generator interrupt flag clear register (DMAMUX_RG_INTC)

Address offset: 0x144

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.

3:0	TOIFCx	Clear bit for trigger overrun event flag of request generator channel x (x=0..3) Writing 1 in each bit clears the corresponding overrun flag TOIFx in the DMAMUX_RG_INTF register.

5. Flash memory controller (FMC)

5.1. Introduction

The flash memory controller, FMC, provides all the necessary functions for the on-chip flash memory. There is no waiting time while CPU executes instructions stored in the maximum first 256K bytes of the flash. It also provides page erase, mass erase, and 32bit word / 16bit half-word program operations for flash memory.

5.2. Characteristics

- Up to 1024KB of main flash memory for instruction and data;
- No waiting time within maximum first 256K bytes when CPU executes instructions. A long delay when CPU fetches the instructions out of the range;
- 2 banks adopted for GD32F50x with flash more than 512KB. Bank0 is used for the first 512KB and bank1 is for the rest capacity;
- The flash page size is 2KB for bank0, 4KB for bank1;
- Word/half-word programming, page erase and mass erase operation;
- 16B option bytes block for user application requirements;
- Option bytes are uploaded to the option byte control registers on every system reset;
- Flash security protection to prevent illegal code/data access;
- Page erase/program protection to prevent unexpected operation.
- 18K bytes information block for bootloader.
- 64Byte OTP0 (One-time program) block used for user data storage.
- 128K byte OTP1 used for BOOT entry or user data storage.
- 256Byte OTP2 with write lock and read lock, used for user data storage.
- 48Byte OTP3 used for critical security configuration(12 bytes user-programmable).

5.3. Function overview

5.3.1. Flash memory architecture

For the GD32F50x, the flash memory can have up to 256 pages of 2K bytes each and 128 pages of 4K bytes each. Each page can be erased individually.

Table 5-1. GD32F50x base address and size for flash memory shows the details of flash organization.

Table 5-1. GD32F50x base address and size for flash memory

Block		Name	Address Range	size (bytes)
Main Flash	Bank0	Page 0	0x0800 0000 - 0x0800 07FF	2KB

Block		Name	Address Range	size (bytes)
Block	512KB	Page 1	0x0800 0800 - 0x0800 0FFF	2KB
		...	...	...
		Page 255	0x0807 F800 - 0x0807 FFFF	2KB
	Bank1 512KB	Page 256	0x0808 0000 – 0x0808 0FFF	4KB
		Page 257	0x0808 1000 – 0x0808 1FFF	4KB
		….	….	…
		Page 383	0x080F F000 – 0x080F FFFF	4KB
Information Block		Bootloader	0x1FFF B000- 0x1FFF F7FF	18KB
Option bytes Block		Option bytes	0x1FFF F800 - 0x1FFF F80F	16B
OTP0 Block		data area	0x1FFF 7800 - 0x1FFF 783F	64B
		lock area	0x1FFF 7840 - 0x1FFF 787F	64B
OTP1 Block		data area	0x1FF0 0000 - 0x1FF1 FFFF	128KB
		lock area	0x1FF2 0100 - 0x1FF2 010F	16B
OTP2 Block		data area	0x1FF2 0000 - 0x1FF2 00FF	256B
		lock area	0x1FF2 0110 - 0x1FF2 018F	128B
OTP3 Block		data area	0x1FFF 7900 - 0x1FFF 792F	48B
		lock area	0x1FFF 7930 - 0x1FFF 793F	16B

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

5.3.2. Read operations

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

For devices with a code area of 128KB and 192KB, the erase operation in the data area will occupy the BUS, and read operations in the code area will be blocked. For devices with a code area of 256KB, the erase operation in the data area will not occupy the BUS, and read operations in the code area will not be blocked. For all devices, programming operations will occupy the BUS, and read operations in the code area will be blocked. To avoid this issue, the read operation code can be placed in SRAM and prevent reading from the flash memory area. The code area size for different devices can be referenced in the datasheet.

5.3.3. Unlock the FMC_CTLx/FMC_OBCTLx registers

After reset, the FMC_CTLx registers are not accessible in write mode, and the LK bit in FMC_CTLx register is 1. An unlocking sequence consists of two write operations to the FMC_KEY0 register to open the access to the FMC_CTL0 register. The two write operations are writing 0x45670123 and 0xCDEF89AB to the FMC_KEY0 register. After the two write operations, the LK bit in FMC_CTL0 register is reset to 0 by hardware. The software can lock the FMC_CTL again by setting the LK bit in FMC_CTL0 register to 1. Any wrong operations to the FMC_KEY0, set the LK bit to 1, and lock FMC_CTL0 register, and lead to a bus error.

The FMC_OBCTLx (x = 0,1,2) registers are still protected even the FMC_CTL is unlocked. The unlocking sequence is two write operations, which are writing 0x45670123 and 0xCDEF89AB to FMC_OBKEY register. After the two write operations, the OB_LK bit in FMC_OBCTL0 register is reset to 0 by hardware. The software can relock FMC_OBCTLx by first setting the LK bit in the FMC_CTL0 register to 1, and then setting the OB_LK bit in the FMC_OBCTL0 register to 1.

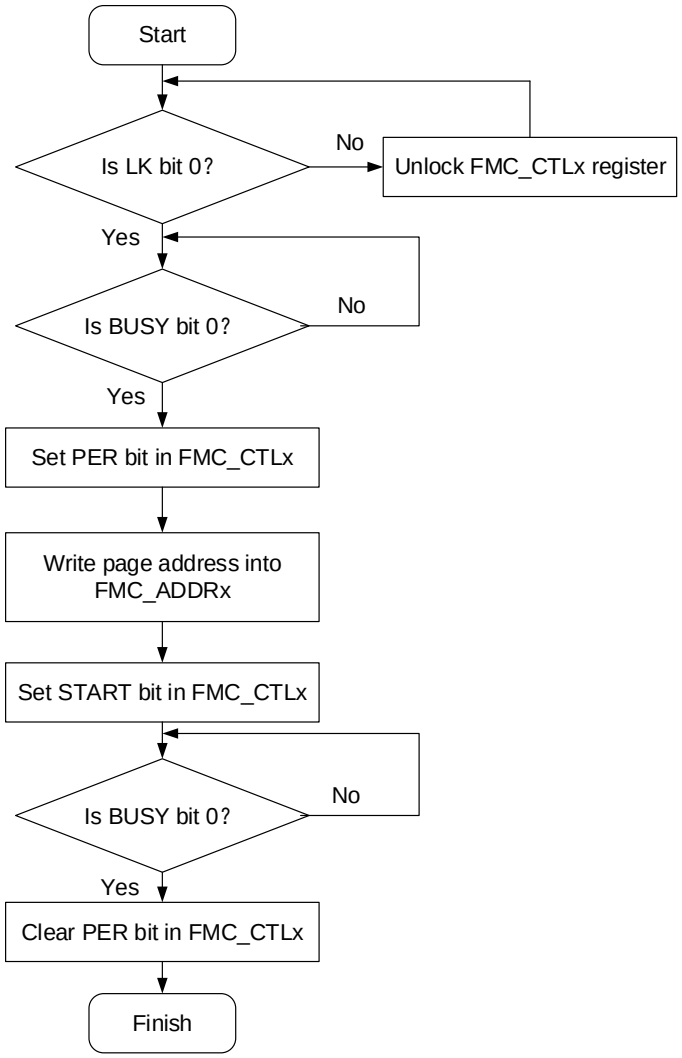
5.3.4. Page erase

The FMC provides a page erase function which is used to initialize 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 registers for a page erase operation.

1. Unlock the FMC_CTLx registers if necessary;
2. Check the BUSY bit in FMC_STATx registers to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished;
3. Set the PER bit in FMC_CTLx registers;
4. Write the page address (0x08XX XXXX) into the FMC_ADDRx registers;
5. Send the page erase command to the FMC by setting the START bit in FMC_CTLx registers;
6. Wait until all the operations have finished by checking the value of the BUSY bit in FMC_STATx registers;
7. Read and verify the page if required using a CBUS access.

When the operation is executed successfully, the ENDF in FMC_STATx registers is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTLx registers is set. Note that a correct target page address must be confirmed. Or the software may run out of control if the target erase page is being used to fetch codes or to access data. The FMC will not provide any notification when this occurs. Additionally, the page erase operation will be ignored on erase/program protected pages. In this condition, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTLx registers is set. The software can check the WPERR bit in the FMC_STATx registers to detect this condition in the interrupt handler. ***Figure 5-1. Process of page erase operation*** shows the page erase operation flow.

Figure 5-1. Process of page erase operation



FMC_STAT0 reflects the operation status of bank0, and FMC_STAT1 reflects the operation status of bank1. The page erase procedure applied to bank1 is similar to the procedure applied to bank0. Especially, when erasing page in bank1 under security protection, the address should not only be written to FMC_ADRR1 but also to FMC_ADDR0.

5.3.5. Mass erase

The FMC provides a complete erase function which is used to initialize the main flash block contents. This erase can affect only on Bank0 by setting MER bit to 1 in the FMC_CTL0 register, or only on Bank1 by setting MER bit to 1 in the FMC_CTL1 register, or on entire flash by setting MER bits to 1 in FMC_CTL0 register and FMC_CTL1 register. The following steps show the mass erase register access sequence.

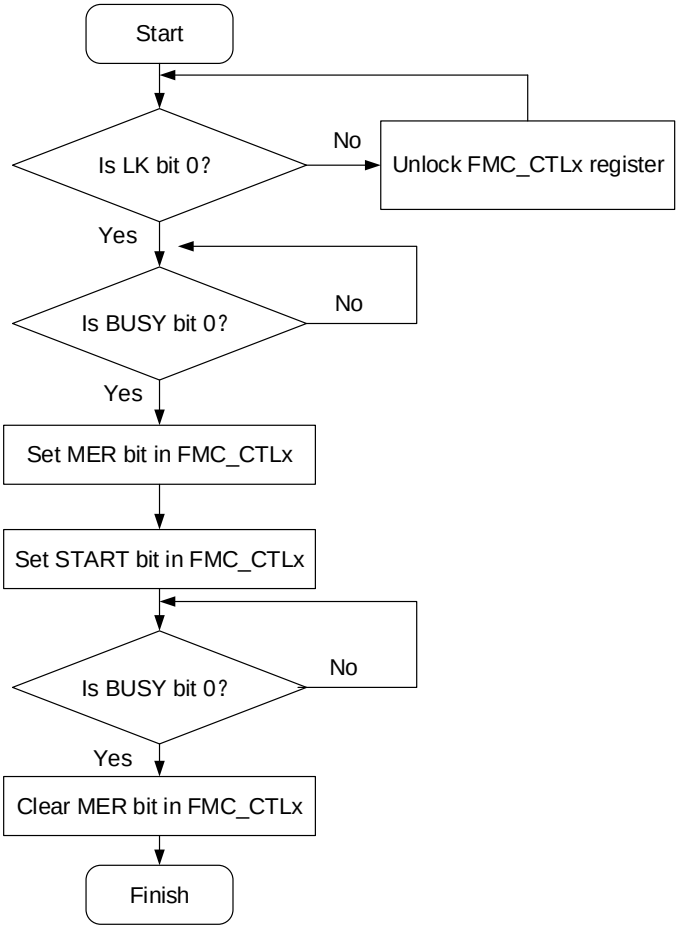
1. Unlock the FMC_CTLx registers if necessary;
2. Check the BUSY bit in FMC_STATx registers to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished;

3. Set MER bit in FMC_CTL0 register if erase Bank0 only. Set MER bit in FMC_CTL1 register if erase Bank1 only. Set MER bits in in FMC_CTL0 register and FMC_CTL1 register if erase entire flash;
4. Send the mass erase command to the FMC by setting the START bit in FMC_CTLx register;
5. Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STATx registers;
6. Read and verify the flash memory if required using a CBUS access.

When the operation is executed successfully, the ENDF in FMC_STATx registers is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTLx registers is set. Since all flash data will be modified 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 that accesses the FMC registers directly.

Figure 5-2. Process of mass erase operation indicates the mass erase operation flow.

Figure 5-2. Process of mass erase operation



5.3.6. Main flash programming

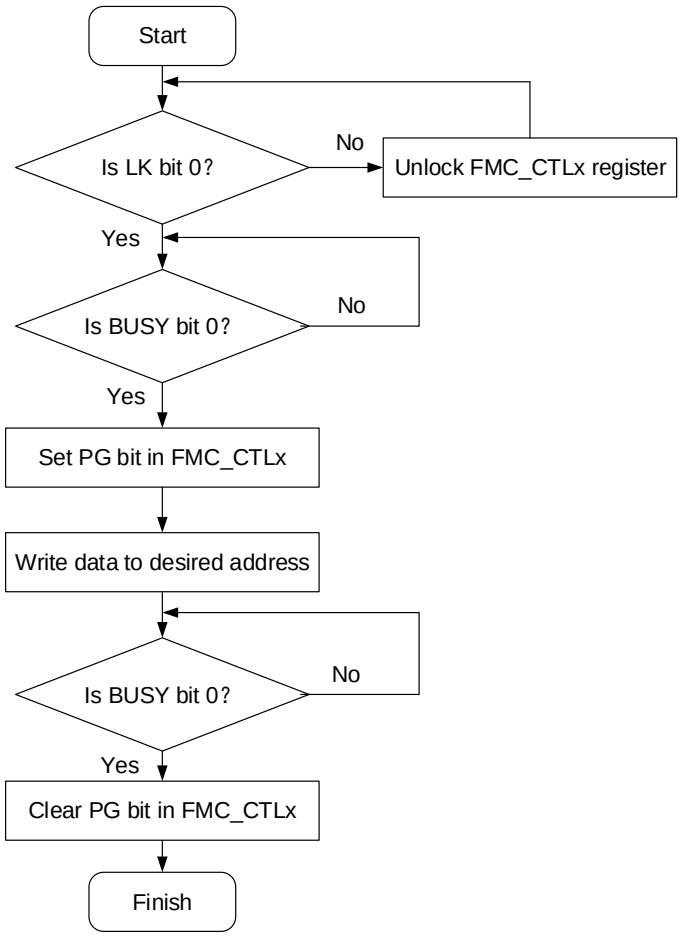
The FMC provides a 32-bit word/16-bit half word programming function which is used to

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

1. Unlock the FMC_CTLx registers if necessary;
2. Check the BUSY bit in FMC_STATx registers to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished;
3. Set the PG bit in FMC_CTLx registers;
4. Write a 32-bit word/16-bit half word to desired absolute address by CBUS;
5. Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STATx registers;
6. Read and verify the Flash memory if required using a CBUS access.

When the operation is executed successfully, the ENDF in FMC_STATx registers is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTLx registers is set. Note that the word/half word programming operation checks the address if it has been erased. If the address has not been erased, PGERR bit in the FMC_STATx registers will be set when program the address except programming 0x0. Note that the PG bit must be set before the word/half word programming operation. Additionally, the program operation will be ignored on erase/program protected pages and WPERR bit in FMC_STATx is set. In these conditions, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTLx registers is set. The software can check the PGERR bit or WPERR bit in the FMC_STATx registers to detect which condition occurred in the interrupt handler. ***Figure 5-3. Process of word program operation*** displays the word programming operation flow.

Figure 5-3. Process of word program operation



Note: Flash memory accesses failed if the CPU enters the power saving modes.

5.3.7. Option bytes modify

The FMC provides an erase and then program function which is used to modify the option bytes block in flash. There are 8 pair option bytes. The MSB is the complement of the LSB in each pair. And when the option bytes are modified, the MSB is generated by FMC automatically, not the value of input data. The following steps show the modify sequence.

1. Unlock the FMC_CTL0 register if necessary;
2. Check the BUSY bit in FMC_STAT0 register to confirm that no Flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished;
3. Unlock the option bytes operation bits in FMC_CTL0 register if necessary;
4. Write the new options byte contents by programming the FMC_OBCTLx register.
5. Send the option bytes modify command to the FMC by setting the OB_START bit in FMC_OBCTL0 register.
6. Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STAT0 register;
7. Read and verify the Flash memory if required using a CBUS access.

When the operation is executed successfully, the ENDF in FMC_STAT0 register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL0 register is set.

Note: The modified option bytes only take effect after a system reset (NWLDE bit of FMC_CTL0 should be set) is generated.

5.3.8. Option bytes description

The option bytes block is reloaded to FMC_OBCTLx (x = 0,1,2) and FMC_OBSTAT registers after each system reset(NWLDE bit of FMC_CTL0 should be set), and the option bytes take effect. The complement option bytes are the opposite of option bytes. When option bytes reload, if the complement option byte and option byte do not match, the OBERR bit in FMC_OBCTL1 register is set, and the option byte is set to 0xFF. The OBERR bit is not set if both the option byte and its complement byte are 0xFF.*Table 5-2. Option byte* is the detail of option bytes.

Table 5-2. Option byte

Address	Name	factory value	Description
0x1fff f800	SPC	0xAA	option byte Security Protection value 0xAA: no security protection 0xCC: security protection level high any value except 0xAA or 0xCC : security protection level low.
0x1fff f801	SPC_N	0x55	SPC complement value
0x1fff f802	USER	0x9F	[7]: reserved [6:5]: NWLD_CLK Select the clock for loading no waiting time area. 00: 200M PLL CLK 01: 160M PLL CLK 10: 120M PLL CLK 11: 8M IRC8M CLK [4]: ECC_EN 0: SRAM ECC disable 1: SRAM ECC enable [3]: SRAM_RST 0: Initialize SRAM after power reset. 1: Do not initialize SRAM after power reset; [2]: nRST_STDBY 0: generator a reset instead of entering standby mode 1: no reset when entering standby mode [1]: nRST_DPSLP 0: generator a reset instead of entering Deep-sleep mode 1: no reset when entering Deep-sleep mode [0]: nWDG_HW 0: hardware free watchdog

Address	Name	factory value	Description
			1: software free watchdog
0x1fff f803	USER_N	0x60	USER complement value
0x1fff f804	DATA[7:0]	0xFF	user defined data bit 7 to 0
0x1fff f805	DATA_N[7:0]	0xFF	DATA complement value bit 7 to 0
0x1fff f806	DATA[15:8]	0xFF	user defined data bit 15 to 8
0x1fff f807	DATA_N[15:8]	0xFF	DATA complement value bit 15 to 8
0x1fff f808	WP[7:0]	0xFF	Page Erase/Program Protection bit 7 to 0 0: protection active 1: unprotected
0x1fff f809	WP_N[7:0]	0xFF	WP complement value bit 7 to 0
0x1fff f80a	WP[15:8]	0xFF	Page Erase/Program Protection bit 15 to 8
0x1fff f80b	WP_N[15:8]	0xFF	WP complement value bit 15 to 8
0x1fff f80c	WP[23:16]	0xFF	Page Erase/Program Protection bit 23 to 16
0x1fff f80d	WP_N[23:16]	0xFF	WP complement value bit 23 to 16
0x1fff f80e	WP[31:24]	0xFF	Page Erase/Program Protection bit 31 to 24
0x1fff f80f	WP_N[31:24]	0xFF	WP complement value bit 31 to 24

5.3.9. 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_STATx registers 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 draw the attention of the CPU. The page protection function can be individually enabled by configuring the WP [31:0] bit field to 0 in the option bytes. If a page erase operation is executed on the option bytes block, all the Flash Memory page protection functions will be disabled. When WP in the option bytes is modified, a system reset followed is necessary.

Table 5-3. WP bit for pages protected

WP bit	protected pages
WP[0]	Page0, Page1
WP[1]	Page2, Page3
…	…
WP[29]	Page58, Page59
WP[30]	Page60, Page61
WP[31]	Page62 - Page383

5.3.10. OTP block programming

The OTP programming method is the same as Bank0, using FMC_CTL0 and FMC_STAT0

register. All OTP can only be programmed one time and not be erased. Each byte of lock blocks can only be programmed one time from 0xFF to 0x00, no other value.

The FMC provides a 32-bit word / 16-bit half word/8-bit byte programming function which is used to modify OTP0/OTP1/OTP2/OTP3 contents. Available programming width is shown as *Table 5-4. OTP available programming width*.

Table 5-4. OTP available programming width

Block	Name	Size	Available programming width		
			32bit	16bit	8bit
OTP0	data area	64*1Byte	-	-	1
	lock area	1*64Byte	-	-	1
OTP1	data area	16*8KByte	1	1	1
	lock area	1*16Byte	-	-	1
OTP2	data area	64*4Byte	1	1	1
	lock area	1*128Byte	-	-	1
OTP3	data area	3*4Byte	1	-	-
	lock area	3*4Byte	1	-	-

Note: If the BootLoader is used to modify the content of OTP1 and OTP2, only the programming function of 32-bit word t is supported, and it must be 4-byte aligned.

The OTP0 block can be divided to 64 data blocks which has 1 byte each and 1 lock block which has 64 bytes. The lock block address is from 0x1FFF 7840 to 0x1FFF 787F. The data block address is from 0x1FFF 7800 to 0x1FFF 783F. Each lock byte (0x00: lock 0xFF:no lock) can lock corresponding data blocks to prevent program to this data block. The lock byte 0 on 0x1FFF 7840 locks data block 0 on 0x1FFF 7800 and so on. Programming a locked data block will result in a WPERR bit programming protection error.

Table 5-5. OTP0 lock

Lock byte	Lock byte address	Locked data block	Lockeddata address
0	0x1FFF 7840	0	0x1FFF 7800
1	0x1FFF 7841	1	0x1FFF 7801
...	...	...	...
62	0x1FFF 787E	62	0x1FFF 783E
63	0x1FFF 787F	63	0x1FFF 783F

The OTP1 block can be divided to 16 data blocks which has 8K bytes each and 1 lock block which has 16 bytes. The data block address is from 0x1FF0 0000 to 0x1FF1 FFFF. The lock block address is from 0x1FF2 0100 to 0x1FF2 010F. Each lock byte (0x00: lock 0xFF:no lock) can lock corresponding data blocks to prevent program to this data block. Programming a locked data block will result in a WPERR bit programming protection error. OTP1 data blocks can be read or not determined by OTP1REN[15:0] in FMC_OTP1CFG register. Reading a locked block of data will cause a bus error.

Table 5-6. OTP1 lock and data

Lock byte	Lock byte address	Locked data block	Locked data address
0	0x1FF2 0100	0	0x1FF0 0000 - 0x1FF0 1FFF
1	0x1FF2 0101	1	0x1FF0 2000 - 0x1FF0 3FFF
...	...	...	...
14	0x1FF2 010E	14	0x1FF1 C000 - 0x1FF1 DFFF
15	0x1FF2 010F	15	0x1FF1 E000 - 0x1FF1 FFFF

The OTP2 block can be divided to 64 data blocks which has 4 bytes each and 1 lock block which has 128 bytes. The data block address is from 0x1FF2 0000 to 0x1FF2 00FF. The lock block address is from 0x1FF2 0110 to 0x1FF2 018F.

The OTP2 write lock block address is from 0x1FF2 0110 to 0x1FF2 014F. Each lock byte (0x00:lock 0xFF:no lock) can lock corresponding data blocks to prevent program to this data block. The lock byte 0 on 0x1FF2 0110 locks data block 0, and so on. Programming a locked data block will result in a WPERR bit programming protection error.

The OTP2 read lock block address is from 0x1FF2 0150 to 0x1FF2 018F. Each lock byte (0x00:lock 0xFF:no lock) can lock corresponding data blocks to prevent read access. The lock byte 64 on 0x1FF2 0150 locks data block 0, and so on.

When the RLBE bit in FMC_CTL0 is set, the OTP2 data block corresponding to the read lock block cannot be read. For example, security verification data is stored in OTP2, and the security startup program starting from OTP1 can read OTP2 information for verification. After the verification is completed, the RLBE is set and then jumps to other programs. the OTP2 data block corresponding to the read lock block cannot be read until the next reset. Reading a locked block of data will cause a bus error.

Table 5-7. OTP2 lock and data

Write lock byte	Write lock byte address	Read lock byte	Read lock byte address	Locked data block	Locked data address
0	0x1FF2 0110	64	0x1FF2 0150	0	0x1FF2 0000 - 0x1FF2 0003
1	0x1FF2 0111	65	0x1FF2 0151	1	0x1FF2 0004 - 0x1FF2 0007
...	...	...	...	...	...
63	0x1FF2 014F	127	0x1FF2 018F	63	0x1FF2 00FC - 0x1FF2 00FF

The OTP3 block can be divided to 3 data blocks which has 4 bytes each and 3 lock block which has 4 bytes each. The lock block address is from 0x1FFF 7930, 0x1FFF 7934, 0x1FFF 7938. The data block address is from 0x1FFF 7900, 0x1FFF 7910, 0x1FFF 7920. Each lock word (All 0: lock, All 1:no lock) can lock corresponding data blocks to prevent program to this data block. The lock block 0 on 0x1FFF 7930 - 0x1FFF 7933 locks data block 0 on 0x1FFF 7900 - 0x1FFF 7903 and so on. OTP3 is write-only and read is invalid. It will take effect immediately after successful programming, and the result can be read through the corresponding bit field of the FMC_OTP3_STAT register.

Table 5-8. OTP3 lock and data

Lock block	Lock byte address write-only	Name All 0 valid	Data block	Locked data address write-only	Name All 0 valid
0	0x1FFF 7930 - 0x1FFF 7933	NDBG_LK	0	0x1FFF 7900 - 0x1FFF 7903	NDBG
1	0x1FFF 7934 - 0x1FFF 7937	NBTSB_LK	1	0x1FFF 7910 - 0x1FFF 7913	NBTSB
2	0x1FFF 7938 - 0x1FFF 793B	BTFOSEL_LK	2	0x1FFF 7920 - 0x1FFF 7923	BTFOSEL

5.3.11. 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. ***Table 5-9. Security protection*** shows different configurations. There are 3 levels for protecting:

No protection: when setting SPC byte to 0xAA, no protection performed. The main flash and option bytes block are accessible by all operations.

Protection level low: when setting SPC byte to any value except 0xAA or 0xCC, 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 WPERR bit in FMC_STATx 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 SPC byte to 0xAA, a mass erase for main flash will be performed.

Protection level high: when setting SPC byte to 0xCC, protection level high performed. When this level is programmed, debug mode, boot from SRAM or boot from boot loader mode are disabled. The main flash block is accessible by all operations from user code. The SPC byte cannot be reprogrammed. So, if protection level high is programmed, it cannot move back to protection level low or no protection level.

Table 5-9. Security protection

SPC[7:0]	Security Protection
0xAA	No protection
except 0xAA or 0xCC	Protection level low
0xCC	Protection level high

5.3.12. Frequency Control

The FMC module uses the clock CK_FMC to access SIP flash, and it must meet certain frequency relationships with the clock CK_AHB. CK_FMC must not be slower than the clock CK_AHB, but not exceed seven times the system clock:

$$CK_FMC \geq CK_AHB \geq 1/7\ CK_FMC$$

The recommended frequency switching configuration method is as follows:

If increasing the frequency:

1. Ensure that CK_FMC selects the clock source as the clock CK_AHB, and simultaneously increase the frequency of CK_FMC and CK_AHB;
2. Then, increase the frequency of CK_FMC separately as needed.

If decreasing the frequency:

1. First, set CK_FMC to select the clock source as the clock CK_AHB;
2. Synchronize the frequency reduction of CK_FMC and CK_AHB.

5.4. FMC registers

FMC base address: 0x4002 2000

5.4.1. Unlock key register 0(FMC_KEY0)

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

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

5.4.2. Option byte 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]	Option bytes operation unlock register. These bits can only be written by software. Write OBKEY[31:0] with keys to unlock option bytes command in FMC_OBCTLx (x = 0,1,2) register.

5.4.3. Status register 0 (FMC_STAT0)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

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

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 erase/program 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 program 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 is busy bit When the operation is in progress, this bit is set to 1. When the operation is end or an error is generated, this bit is cleared to 0.

5.4.4. Control register 0(FMC_CTL0)

Address offset: 0x10
Reset value: 0x0000 0080

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

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

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

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

Bits	Fields	Descriptions
31	NWLDE	Enable no waiting time area load when system reset. No waiting time area load include load option bytes, OTP, main flash code area. This bit is set or cleared by software, not reset after a system reset, but reset after a power-on reset. 0: Don't copy flash content to buffer memory when system reset. 1: Copy flash content to buffer memory when system reset.
30	PWDN	Flash enter to Deep Power Down mode when no operation. This bit is set or cleared by software, not reset after a system reset, but reset after a power-on reset. 1: Enter to Deep Power Down mode. 0: Not enter to Deep Power Down mode. **Note**: 1. In power-saving mode, the flash memory will only enter deep power-down mode when the PWDN bit is set to 1. 2. When CPU Cbus timeout is enabled (CPUCBUSTO=1) and PWDN=1, accessing the non-zero wait area (data area) of the flash memory after it enters deep power-down mode will cause CBUS timeout, triggering Hardfault error. 3. When CPU Cbus timeout is not enabled (CPUCBUSTO=0) and PWDN=1, accessing the non-zero wait area (data area) of the flash memory after it enters deep power-down mode will wake up the flash memory, requiring flash wake-up time. Accessing the zero wait area (code area) of the flash memory will not wake up the flash memory and does not require waiting.
29:14	Reserved	Must be kept at reset value.
13	RLBE	Enable read lock block for OTP2. This bit can only be set to 1 to enable read lock block for OTP2. Once the software sets the value to 1, the data block corresponding to the read lock block cannot be read. Reset value is restored after 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
9:8	Reserved	Must be kept at reset value.
7	LK	FMC_CTL0 lock bit

		This bit is cleared by hardware when right sequence written to FMC_KEY0 register. This bit can be set by software.
6	START	Send erase command to FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
5:3	Reserved	Must be kept at reset value.
2	MER	Main flash mass erase for bank0 command bit This bit is set or cleared by software. 0: No effect 1: Main flash mass erase command for bank0
1	PER	Main flash page erase for bank0 command bit This bit is set or clear by software. 0: No effect 1: Main flash page erase command for bank0
0	PG	Main flash program for bank0 command bit This bit is set or clear by software. 0: No effect 1: Main flash program command for bank0

5.4.5. Address register 0 (FMC_ADDR0)

Address offset: 0x14

Reset value: 0x0000 0000.

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

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

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

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

5.4.6. Option byte control register 0 (FMC_OBCTL0)

Address offset: 0x18

Reset value: 0x0XXX 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
SPC								Reserved						OB_START	OB_LK
rw														rs	rs

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	SPC	Option byte security protection code. 0xAA: no security protection 0xCC: security protection level high Any value except 0xAA or 0xCC : security protection level low.
7:2	Reserved	Must be kept at reset value.
1	OB_START	Send option byte change command to FMC bit. This bit is set by software to send option byte change command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
0	OB_LK	FMC_OBCTL0/1/2 lock bit This bit is cleared by hardware when right sequence written to FMC_OBKEY register. This bit can be set by software.

5.4.7. Option byte control register 1 (FMC_OBCTL1)

Address offset: 0x1C

Reset value: 0x0XXX 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						DATA[15:6]									
						rw									

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:10	DATA[15:0]	Store DATA of option bytes block after system reset.
9:2	USER[7:0]	Store USER of option bytes block after system reset.

1	Reserved	Must be kept at reset value.
0	OBERR	Option bytes read error bit. This bit is set by hardware when the option bytes and its complement byte do not match, then the option bytes is set to 0xFF.

5.4.8. Option byte control register 2 (FMC_OBCTL2)

Address offset: 0x20

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
WP[31:16]															
rw															

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

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

5.4.9. OTP1 configuration register (FMC_OTP1CFG)

Address offset: 0x24

Reset value: 0x0000 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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTP1REN[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	OTP1REN[15:0]	OTP1 read enable. OTP1REN[x] decides OTP1 data block x read or not, x=0...15. 0: Data block can not be read 1: Data block can be read Software can write 0, but only reset can set to 1.

5.4.10. Option bytes status register (FMC_OBSTAT)

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
Reserved														SPCH	SPCL
														r	r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SPCH	Security protection is level high now.
0	SPCL	Security protection is level low now.

5.4.11. Unlock key register 1(FMC_KEY1)

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
KEY[31:16]															
w															

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

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

5.4.12. Status register 1 (FMC_STAT1)

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
Reserved										ENDF	WPERR	Reserved	PGERR	Reserved	BUSY
										rc_w1	rc_w1		rc_w1		rc_w1

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 erase/program 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 program 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 is busy bit. When the operation is in progress, this bit is set to 1. When the operation is end or an error is generated, this bit is cleared to 0.

5.4.13. Control register 1(FMC_CTL1)

Address offset: 0x50

Reset value: 0x0000 0080

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

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

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			ENDIE	Reserved	ERRIE	Reserved		LK	START	Reserved			MER	PER	PG
			rw		rw			rs	rs				rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	ENDIE	End of operation interrupt enable bit This bit is set or cleared by software.

		0: No interrupt generated by hardware 1: End of operation interrupt enable
11	Reserved	Must be kept at reset value
10	ERRIE	Error interrupt enable bit This bit is set or cleared by software. 0: No interrupt generated by hardware 1: Error interrupt enable
9:8	Reserved	Must be kept at reset value.
7	LK	FMC_CTL1 lock bit This bit is cleared by hardware when right sequence written to FMC_KEY1 register. This bit can be set by software.
6	START	Send erase command to FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
5:3	Reserved	Must be kept at reset value
2	MER	Main flash mass erase for bank1 command bit This bit is set or cleared by software. 0: No effect 1: Main flash mass erase command for bank1
1	PER	Main flash page erase for bank1 command bit This bit is set or clear by software. 0: No effect 1: Main flash page erase command for bank1
0	PG	Main flash program for bank1 command bit This bit is set or clear by software. 0: No effect 1: Main flash program command for bank1

5.4.14. Address register 1 (FMC_ADDR1)

Address offset: 0x54

Reset value: 0x0000 0000

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

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

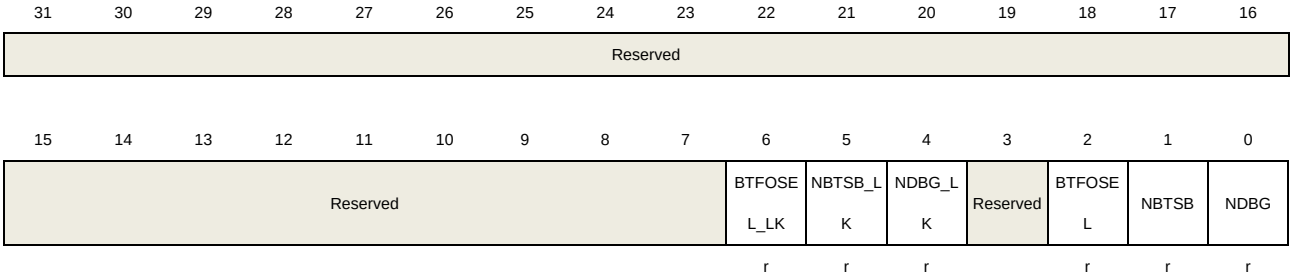
Bits	Fields	Descriptions
31:0	ADDR[31:0]	Flash erase command address bits These bits are configured by software. ADDR bits are the address of flash erase command

5.4.15. OTP3 status register (FMC_OTP3_STAT)

Address offset: 0x60

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	BTFOSEL_LK	BTFOSEL lock status. 0: No lock 1: Locked
5	NBTSB_LK	NBTSB lock status. 0: No lock 1: Locked
4	NDBG_LK	NDBG lock status. 0: No lock 1: Locked
3	Reserved	Must be kept at reset value.
2	BTFOSEL	Select Boot from flash or OTP1. This bit actives when NBTSB is 1 or BOOT0 is 0. 0: Boot from main flash block 1: Boot from OTP1 block
1	NBTSB	Not boot from SRAM or bootloader. 0: Can boot from SRAM or bootloader 1: Can’t boot from SRAM or bootloader
0	NDBG	Debugging permission setting. 0: No influence

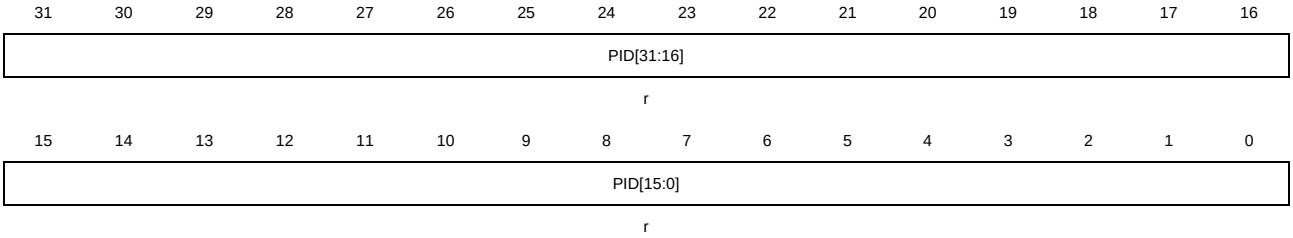
1: Can not debug

5.4.16. Product ID register (FMC_PID)

Address offset: 0x100

Reset value: 0xXXXX XXXX

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



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

6. Power management unit (PMU)

6.1. Overview

The power consumption is regarded as one of the most important issues for the devices of this series. Power management unit (PMU) provides three types of power saving modes, including Sleep, Deep-sleep and Standby mode. These modes reduce the power consumption and allow the application to achieve the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. For this devices, 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 VDD. 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.

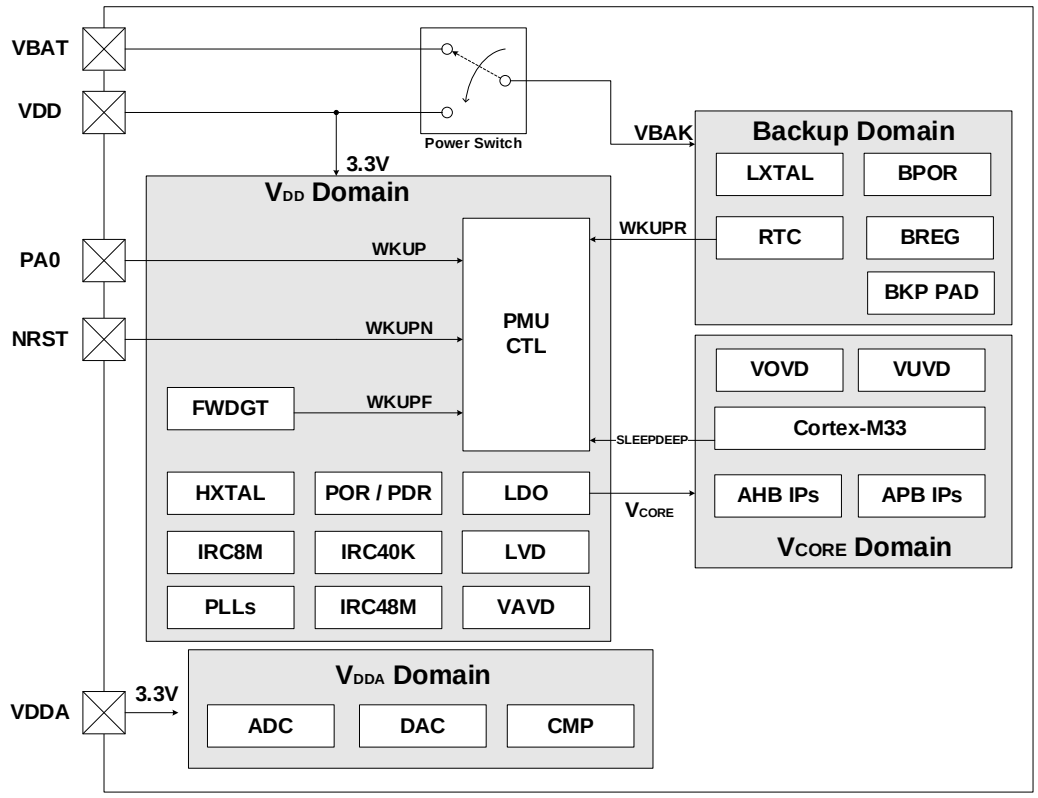
6.2. Characteristics

- Three power domains: Backup, V_{DD}/V_{DDA} and V_{CORE} power domains.
- Three power saving modes: Sleep, Deep-sleep and Standby modes.
- Internal Voltage regulator(LDO) supplies around V_{CORE} voltage source for V_{CORE} domain.
- Low Voltage Detector (LVD) issue an interrupt or event when the power is lower than a programmed threshold.
- Battery power (VBAT) for Backup domain when V_{DD} is shut down.
- LDO output voltage select for power saving.
- Power supply supervision: POR / PDR monitor / LVD monitor /VOVD monitor / VAVD monitor / VUVD monitor.
- Ultra power saving for low-driver mode in Deep-sleep mode.

6.3. Function overview

Figure 6-1. Power supply overview provides details on the internal configuration of the PMU and the relevant power domains.

Figure 6-1. Power supply overview



LVD: Low Voltage Detector LDO: Voltage Regulator BPOR: V_{BAK} Power On Reset VOVD: V_{CORE} Over Voltage Detector
POR: Power On Reset PDR: Power Down Reset BREG: Backup registers VUVD: V_{CORE} Under Voltage Detector
VAVD: Analog Voltage Detector

6.3.1. Backup domain

The Backup domain is powered by the VDD or the battery power source (VBAT) selected by the internal power switch, and the VBAK pin which drives Backup Domain, power supply for RTC unit, LXTAL oscillator, BPOR and BREG, and three BKP PAD including PC13 to PC15. In order to ensure 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 includes 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 the Real Time Clock (RTC) circuit can be derived from the Internal 40KHz RC oscillator (IRC40K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL clock divided by 128, or AHB clock devided by 10. When V_{DD} is shut down, only LXTAL is valid for RTC.

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

When the Backup domain is supplied by VDD (VBAK pin is connected to VDD), the following functions are available:

- PC13 can be used as GPIO or RTC function pin described in the RTC chapter.
- PC14 and PC15 can be used as either GPIO or LXTAL Crystal oscillator pins.

When the Backup domain is supplied by VBAT (VBAK pin is connected to VBAT), the following functions are available:

- PC13 can be used as RTC function pin described in the RTC chapter.
- PC14 and PC15 can be used as LXTAL Crystal oscillator pins only.

Note: Since PC13, PC14, 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).

6.3.2. VDD / VDDA power domain

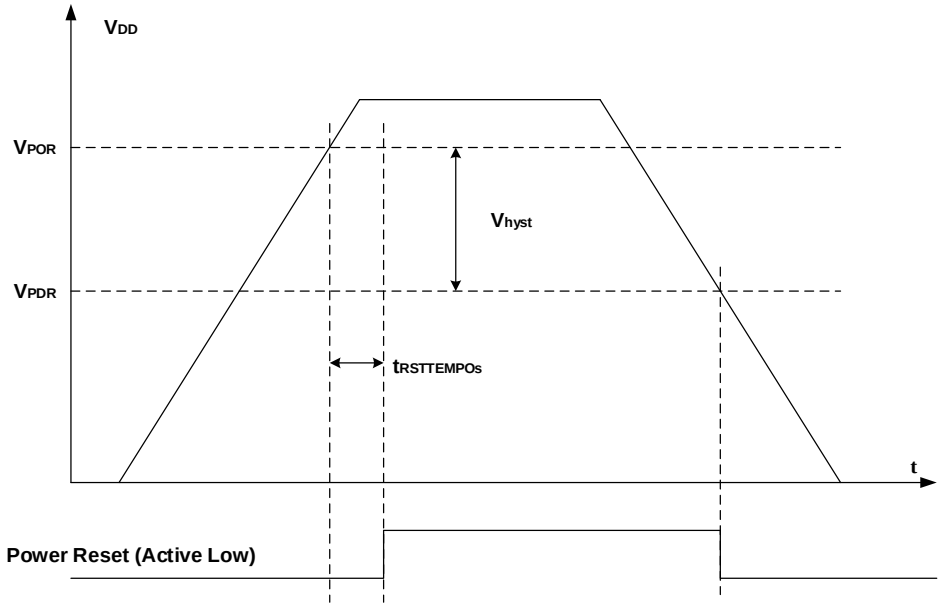
V_{DD} / V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes POR / PDR (power on / down reset), HXTAL (high speed crystal oscillator), LDO (voltage regulator), LVD(Low Voltage Detector), FWDGT (free watchdog timer), IRC8M (internal 8MHz RC oscillator), IRC48M (internal 48MHz RC oscillator at 48MHz frequency), IRC40K (internal 40KHz RC oscillator), , PLLs (phase locking loop), VAVD (Analog Voltage Detector), all pads except PC13 / PC14 / PC15, etc. VDDA domain includes ADC / DAC (AD / DA converter), CMP(comparator), etc.

VDD 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 statuses, including in the Sleep mode (full power on), in the Deep-sleep mode (on or low power), and in the Standby mode (power off).

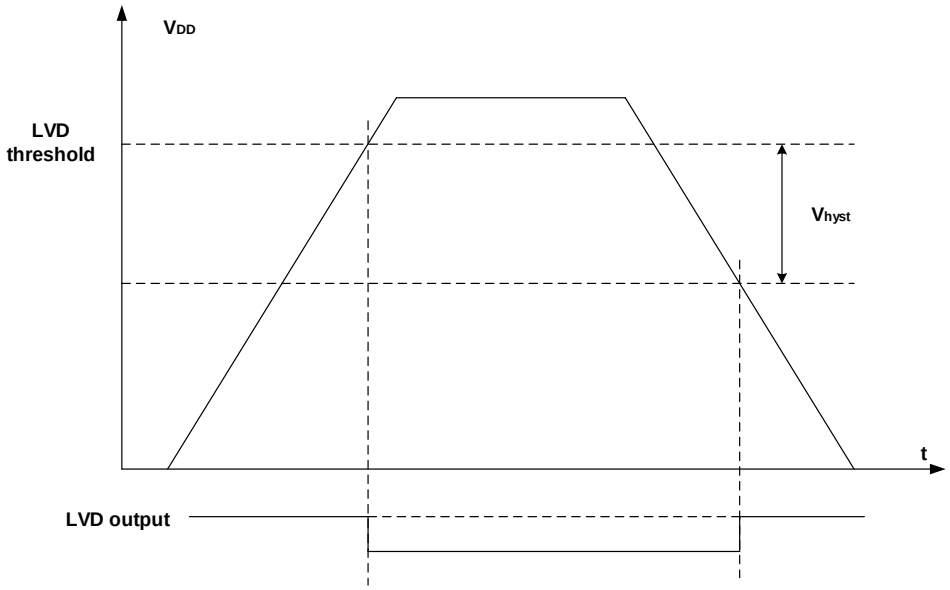
The POR/PDR circuit is implemented to detect V_{DD} and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold.***Figure 6-2. Waveform of the POR/PDR*** shows the relationship between the supply voltage and the power reset signal. V_{POR}, which typical value is refer to device datasheet, indicates the threshold of power on reset, while V_{PDR}, which typical value is refer to device datasheet, means the threshold of power down reset. The hysteresis voltage (V_{hyst}) is refer to device datasheet.

Figure 6-2. Waveform of the POR/PDR



The LVD is used to detect whether the VDD supply voltage is lower than a programmed threshold selected by the LVDT[2:0] bits in the Power control register(PMU_CTL0). The LVD is enabled by setting the LVDEN bit, and LVDF bit, which in PMU_CS, indicates if V_{DD} 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 6-3. Waveform of the LVD threshold*** shows the relationship between the LVD threshold and the LVD output. The following figure shows the relationship between the supply voltage and the LVD signal. The hysteresis voltage (V_{hyst}) please refer to device datasheet.

Figure 6-3. Waveform of the LVD threshold

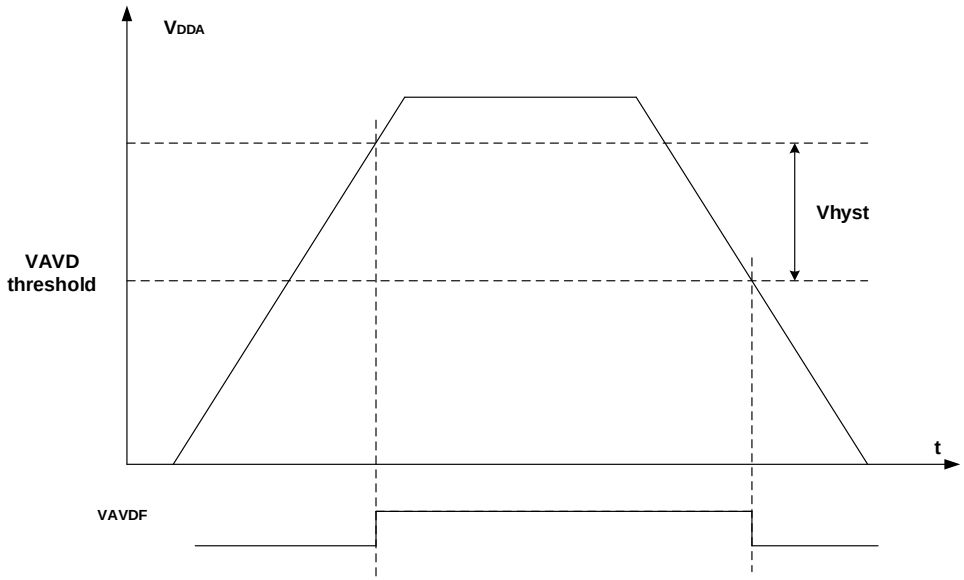


VDDA domain

The V_{DDA} analog voltage detector is used to detect whether the VDDA supply voltage is lower

than a programmed threshold selected by the VAVDVC[1:0] bits in the power control register(PMU_CTL0). The VAVD is enabled by setting the VAVDEN bit, and VAVDF bit, which in PMU_CS, indicates if V_{DDA} is higher or lower than the specified VAVD threshold..*Figure 6-4. Waveform of the VAVD threshold* shows the relationship between the VAVD threshold and the VAVDF The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 6-4. Waveform of the VAVD threshold



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

To ensure a high accuracy on ADC and DAC, the ADC/DAC independent external reference voltage should be connected to VREFP/VREFN pins. According to the different packages, VREFP pin can be connected to VDDA pin, or external reference voltage which refers to*Table 20-2. ADC input pins definition* and *Table 21-1. DAC I/O description*. VREFN pin must be connected to VSSA pin. The VREFN pin is only available on no less than 48-pin packages, or else the VREFN pin is not available and internally connected to VSSA.

6.3.3. V_{CORE} power domain

V_{CORE} power domain supplies power for Cortex®-M33 logic, AHB/APB peripherals, the APB interfaces for the Backup domain and the V_{DD} / V_{DDA} domain, etc, includes VUVD (V_{CORE} Under Voltage Detector) and VOVD(V_{CORE} Over Voltage Detector). Once the V_{CORE} is powered up, the POR will generate a reset sequence on the V_{CORE} power domain. If need 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

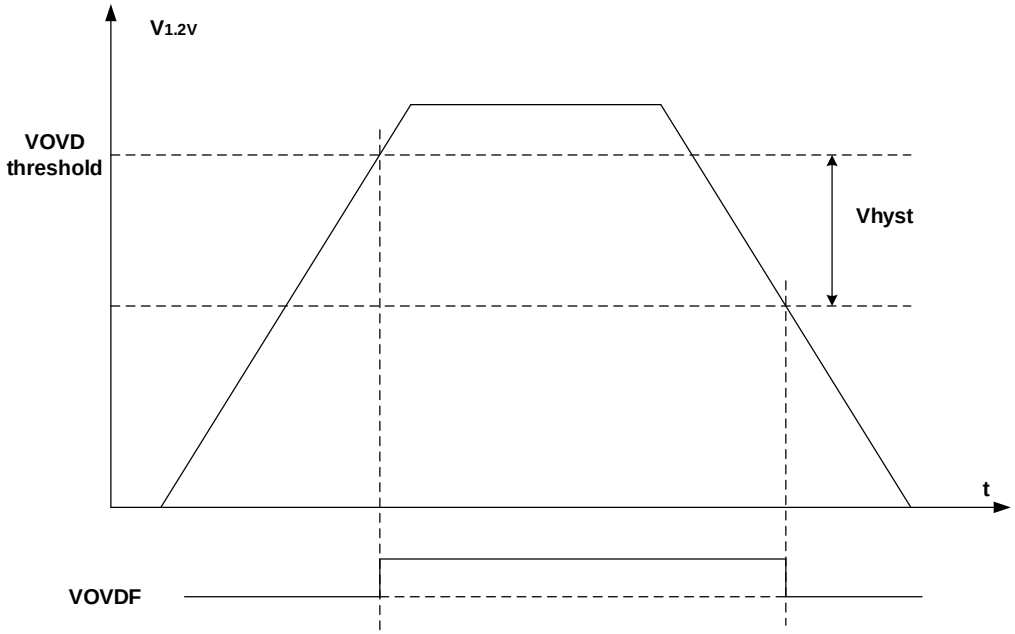
will enter an expected power saving mode which will be discussed in the following section. The voltage of this power domain can be configured by LDOVS[2:0] in the PMU CTL0 register. Refer to the datasheet for the V_{CORE} voltage value.

V_{CORE} voltage thresholds detector

There is an internal V_{CORE} power over voltage detector, when VOVDEN is 0b1, it means V_{CORE} power over voltage detector is enabled. Once V_{CORE} power domain is over than a programmed threshold selected by the VOVDVC[1:0] bits in the power control register(PMU_CTL0), VOVDF will be set immediately after two flip-flops synchronization of the analog output. VOVDF will be set after digital filter which can be used by configuring the VOVDO_DNF[7:0]bit in PMU_CTL1 register.This allows to suppress spikes with a programmable length of 1 to 255 of 1024* T_{pclk}(period of PCLK1) The hysteresis voltage (Vhyst) please refer to device datasheet.

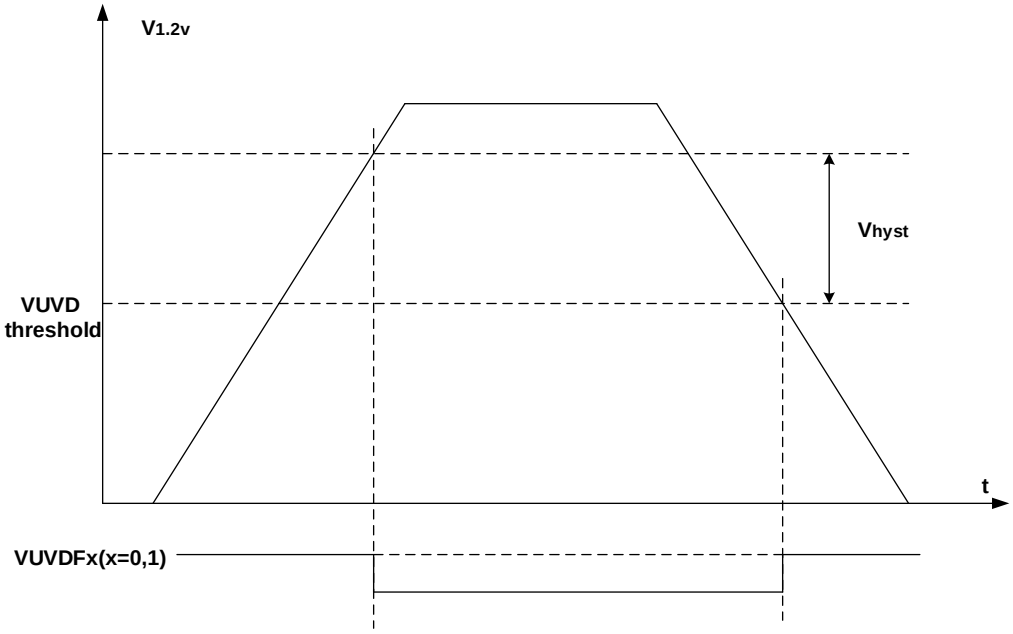
Note: Before enable VOVD, LVD must be enabled first. After a delay of 50μs, VOVD can then be enabled. Otherwise, VOVD may generate false trigger signals. Subsequently, LVD can either remain enabled or be disabled.

Figure 6-5. waveform of VOVD



There is an internal V_{CORE} power under voltage detector, when VUVDEN is 0b1, it means V_{CORE} power under voltage detector is enabled. Once V_{CORE} power domain is lower than a programmed threshold selected by the VUVDVC[1:0] bits in the power control register(PMU_CTL0), VUVDF0 will be set immediately after two flip-flops synchronization of the analog output, VUVDF1 will be set after digital filter which can be used by configuring the VUVDO_DNF[7:0] bits in PMU_CTL1 register.This allows to suppress spikes with a programmable length of 1 to 255 of 1024* T_{pclk}(period of PCLK1) VUVDF1 interrupt is internally connected. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 6-6. waveform of VUVD



6.3.4. Power saving modes

After a system reset or a power reset, the 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, and PCLK2) or gating the clocks of the unused peripherals or configuring the LDO output voltage by LDOVS bits in PMU_CTL register. The LDOVS bits should be configured only when the PLL is off, and the programmed value is selected to drive V_{CORE} domain after the PLL opened. While the PLL is off, LDO output voltage low mode is selected to drive V_{CORE} domain. Besides, three power saving modes are provided to achieve even lower power consumption, they are Sleep mode, Deep-sleep mode, and Standby mode.

Sleep mode

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

According to the SLEEPONEXIT bit in the Cortex®-M33 System Control Register, there are two options to select the 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®-M33. In Deep-sleep mode, all clocks in the V_{CORE} domain are off, and all of IRC8M, IRC48M, 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®-M33 System Control Register, and clear the STBMOD bit in the PMU_CTL register. Then, the device enters the Deep-sleep mode after a WFI or WFE instruction is executed. If the Deep-sleep mode is entered by executing a WFI instruction, any interrupt from EXTI lines can wake up the system. If it is entered by executing a WFE instruction, any wakeup event from EXTI lines can wake up the system (If SEVONPEND is 1, any interrupt from EXTI lines can wake up the system, refer to Cortex®-M33 Technical Reference Manual). When exiting the Deep-sleep mode, the IRC8M is selected as the system clock. Notice that an additional wakeup delay will be incurred if the LDO operates in low power mode.

The low-driver mode in Deep-sleep mode can be entered by configuring the LDEN, LDNP, LDLP, LDOLP bits in the PMU_CTL register. The Low-driver mode provides lower drive capability, and the Low-power mode take lower power.

Normal-driver / Normal-power: The Deep-sleep mode is not in low-driver mode by configure LDEN to 00 in the PMU_CTL register, and not in low-power mode depending on the LDOLP bit reset in the PMU_CTL register.

Normal-driver / Low-power: The Deep-sleep mode is not in low-driver mode by configure LDEN to 00 in the PMU_CTL register. The low-power mode enters depending on the LDOLP bit set in the PMU_CTL register.

Low-driver / Normal-power: The low-driver mode in Deep-sleep mode when the LDO in normal-power mode depending on the LDOLP bit reset in the PMU_CTL register enters by configure LDEN to 0b11 and LDNP to 1 in the PMU_CTL register.

Low-driver / Low-power: The low-driver mode in Deep-sleep mode when the LDO in low-power mode depending on the LDOLP bit set in the PMU_CTL register enters by configure LDEN to 0b11 and LDLP to 1 in the PMU_CTL register.

No Low-driver: The Deep-sleep mode is not in low-driver mode by configure LDEN to 00 in the PMU_CTL register.

Note: In order to enter Deep-sleep mode smoothly, all EXTI line pending status (in the EXTI_PD register) and related peripheral flags must be reset, refer to***Table 2-3. 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®-M33, too. In Standby mode, the whole V_{CORE} domain is power off, the LDO is shut down, and all of IRC8M, IRC48M,

HXTAL and PLL are disabled. Before entering the Standby mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, and set the STBMOD bit in the PMU_CTL register, and clear WUF bit in the PMU_CS register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS register indicates that the MCU has been in Standby mode. There are four wakeup sources for the Standby mode, including the external reset from NRST pin, the RTC alarm, the FWDGT reset, and the rising edge on WKUP pin. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in V_{CORE} power domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset occurs and the Cortex®-M33 will execute instruction code from the 0x00000000 address.

Table 6-1. Power saving mode summary

Mode	Sleep	Deep-sleep	Standby
Description	Only CPU clock is off	1. All clocks in the V_{CORE} domain are off 2. Disable IRC8M, IRC48M, HXTAL and PLL	1. The V_{CORE} domain is power off 2. Disable IRC8M, IRC48M, HXTAL and PLL
LDO Status	On (normal power mode, normal driver mode)	On (normal power mode or low power mode, normal driver mode or low driver mode)	Off
Configuration	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST = 1
Entry	WFI or WFE	WFI or WFE	WFI or WFE
Wakeup	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	1. NRST pin 2. WKUP pin 3. FWDGT reset 4. RTC
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

6.4. PMU registers

PMU base address: 0x4000 7000

6.4.1. Control register0 (PMU_CTL0)

Address offset: 0x00

Reset value: 0x1600 7000 (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
UNLOCK	Reserved		VUVDVC[1:0]		VOVDVC[1:0]		VUVDEN	VOVDEN	VAVDVC[1:0]		VAVDEN	LDEN[1:0]		Reserved	
rw			rw		rw		rw	rw	rw		rw	rw			

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

Bits	Fields	Descriptions
31	UNLOCK	Register lock 0: Other registers and bits are not writable 1: Other registers and bits are writable
30:29	Reserved	Must be kept at reset value.
28:27	VUVDVC[1:0]	V_{CORE}under voltage detector voltage level configure bits These bits are set and cleared by software 00: Configure V_{CORE} under voltage detector voltage level to 1.05V 01: Configure V_{CORE} under voltage detector voltage level to 0.95V. 1x: Configure V_{CORE} under voltage detector voltage level to 0.85V.
26:25	VOVDVC[1:0]	V_{CORE} over voltage detector voltage level configure bits These bits are set and cleared by software 00: Configure V_{CORE} over voltage detector voltage level to 1.25V. 01: Configure V_{CORE} over voltage detector voltage level to 1.30V. 10: Configure V_{CORE} over voltage detector voltage level to 1.35V. 11: Configure V_{CORE} over voltage detector voltage level to 1.40V.
24	VUVDEN	V_{CORE} under Voltage detector enable 0: Disable V_{CORE} under Voltage Detector. 1: Enable V_{CORE} under Voltage Detector.
23	VOVDEN	V_{CORE} over Voltage detector enable This bit is set and cleared by software. 0: Peripheral voltage on V_{CORE} detector disabled. 1: Peripheral voltage on V_{CORE} detector enabled
22:21	VAVDVC[1:0]	VDDA analog voltage detector voltage level configure bits

		These bits are set and cleared by software 00: Configure V_{DDA} analog voltage detector voltage level to 2.3V 01: Configure V_{DDA} analog voltage detector voltage level to 2.5V 10: Configure V_{DDA} analog voltage detector voltage level to 2.7V 11: Configure V_{DDA} analog voltage detector voltage level to 2.9V
20	VAVDEN	V_{DDA} analog voltage detector voltage enable bit This bit is set and cleared by software. 0: V_{DDA} analog voltage detector voltage disabled. 1: V_{DDA} analog voltage detector voltage enabled.
19:18	LDEN[1:0]	Low-driver mode enable in Deep-sleep mode 00: Low-driver mode disable in Deep-sleep mode 01: Reserved 10: Reserved 11: Low-driver mode enable in Deep-sleep mode
17:15	Reserved	Must be kept at reset value.
14:12	LDOVS[2:0]	LDO output voltage select These bits are set by software when the main PLL closed. And the LDO output voltage selected by LDOVS bits takes effect when the main PLL enabled. If the main PLL closed, the LDO output default value. 000: Reserved 001: 0.9v (not recommended for customers to use) 010: 0.95v (not recommended for customers to use) 011: 1.0v (not recommended for customers to use) 100: 1.05v (not recommended for customers to use) 101: 1.1v 110: 1.15v 111: 1.2v **Note:** 0b000 is not supported for this bit field, configured to 0b000 will cause unpredictable errors.
11	LDNP	Low-driver mode when use normal power LDO 0: normal driver when use normal power LDO 1: Low-driver mode enabled when LDEN is 11 and use normal power LDO
10	LDLP	Low-driver mode when use low power LDO. 0: normal driver when use low power LDO 1: Low-driver mode enabled when LDEN is 11 and use low power LDO
9	PDRVS	PDR threshold selection 0: 2.35V 1: 1.8V
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: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.8V 110: 3.0V 111: 3.1V
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_LKCTL register, LVDEN and LVDT[2:0] are read only.
3	STBRST	Standby Flag Reset 0: No effect 1: Reset the standby flag This bit is always read as 0.
2	WURST	Wakeup Flag Reset 0: No effect 1: Reset the wakeup flag This bit is always read as 0.
1	STBMOD	Standby Mode 0: Enter the Deep-sleep mode when the Cortex®-M33 enters SLEEPDEEP mode 1: Enter the Standby mode when the Cortex®-M33 enters SLEEPDEEP mode
0	LDOLP	LDO Low Power Mode 0: The LDO operates normally during the Deep-sleep mode 1: The LDO is in low power mode during the Deep-sleep mode

6.4.2. Control and 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												LDRF[1:0]		Reserved	
												rc_w1			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LDOVSRF	Reserved					WUPEN	VUVDF1	VOVDF	Reserved	VUVDF0	VAVDF	LVDF	STBF	WUF
	r						rw	r	r		r	r	r	r	r

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:18	LDRF[1:0]	Low-driver mode ready flag These bits are set by hardware when enter Deep-sleep mode and the LDO in Low-driver mode. These bits are cleared by software when write 11. 00: normal driver in Deep-sleep mode 01: Reserved 10: Reserved 11: Low-driver mode in Deep-sleep mode
17:15	Reserved	Must be kept at reset value.
14	LDOVSRF	LDO voltage select ready flag 0: LDO voltage select not ready 1: LDO voltage select ready
13:9	Reserved	Must be kept at reset value.
8	WUPEN	WKUP Pin Enable 0: Disable WKUP pin function 1: Enable WKUP pin function If WUPEN is set before entering the Standby mode, a rising edge on the WKUP pin wakes up the system from the Standby mode. As the WKUP pin is active high, the WKUP pin is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
7	VUVDF1	V_{CORE} under voltage detector flag bit after digital filter. this bit is set and cleared by hardware. It is valid only if VUVDEN is enabled. 0: V_{CORE} is higher than VUVD threshold. 1: V_{CORE} is equal or lower than VUVD threshold.
6	VOVDF	V_{CORE} over voltage detector flag bit after digital filter. this bit is set and cleared by hardware. It si valid only if VOVDEN is enabled. 0: V_{CORE} is lower than VOVDVC[1:0] threshold 1: V_{CORE} is equal or higher than VOVDVC[1:0] threshold
5	Reserved	Must be kept at reset value.
4	VUVDF0	V_{CORE} under voltage detector flag bit this bit is set and cleared by hardware. It si valid only if VUVDEN is enabled. 0: V_{CORE} is higher than VUVD threshold. 1: V_{CORE} is equal or lower than VUVD threshold.
3	VAVDF	V_{DDA} analog voltage detector voltage output on V_{DDA} flag bit.

		This bit is set and cleared by hardware. It is valid only if VAVDEN is enabled. 0: V_{DDA} is equal or higher than the VAVD threshold configured by VAVDVC bits. 1: V_{DDA} is lower than the VAVD threshold configured by VAVDVC bits.
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 the Standby mode This bit is cleared only by a POR / PDR or by setting the STBRST bit in the PMU_CTL0 register.
0	WUF	Wakeup Flag 0: No wakeup event has been received 1: Wakeup event occurred from the WKUP pin or RTC alarm event This bit is cleared only by a POR / PDR or by setting the WURST bit in the PMU_CTL0 register.

6.4.3. Control register 1(PMU_CTL1)

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
VUVDO_DNF[7:0]								VOVDO_DNF[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	VUVDO_DNF[7:0]	VUVD analog output digital noise filter These bits are used to configure the digital noise filter on VUVD analog output. The digital filter will filter spikes with a length of up to VUVDO_DNF[7:0] * 1024 *T_{PCLK1} 0: Digital filter is disabled 1: Digital filter is enabled and filter spikes with a length of up to 1024 * T_{PCLK1} ... 255: Digital filter is enabled and filter spikes with a length of up to 255*1024 * T_{PCLK1}

7:0	VOVDO_DNF[7:0]	VOVD analog output digital noise filter These bits are used to configure the digital noise filter on VOVD analog output. The digital filter will filter spikes with a length of up to VOVDO_DNF[7:0] * 1024 * T_{PCLK1} 0: Digital filter is disabled 1: Digital filter is enabled and filter spikes with a length of up to 1024 * T_{PCLK1} ... 255: Digital filter is enabled and filter spikes with a length of up to 255*1024 * T_{PCLK1}

7. Backup registers (BKP)

7.1. Introduction

The Backup registers are located in the Backup domain that remains powered-on by V_{BAT} even if V_{DD} power is shut down, they are forty two 16-bit (84 bytes) registers for data protection of user application data, and the wake-up action from Standby mode or system reset do not affect these registers.

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

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

7.2. Main features

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

7.3. Function description

7.3.1. RTC clock calibration

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

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

7.3.2. Tamper detection

In order to protect the important user data, the MCU provides the tamper detection function,

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

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

7.4. BKP registers

BKP base address: 0x4000 6C00

7.4.1. Backup data register x (BKP_DATAx) (x= 0..41)

Address offset: 0x04 + 0x04*x(x=0..9), 0x40 + 0x04*(x-10)(x=10..41)
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
DATA [15:0]															
rw															

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

7.4.2. RTC signal output control register (BKP_OCTL)

Address offset: 0x2C
Reset value: 0x0000 0000

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

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
CALDIR	CCOSEL	Reserved				ROSEL	ASOEN	COEN	RCCV[6:0]						
rw	rw					rw	rw	rw	rw						

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

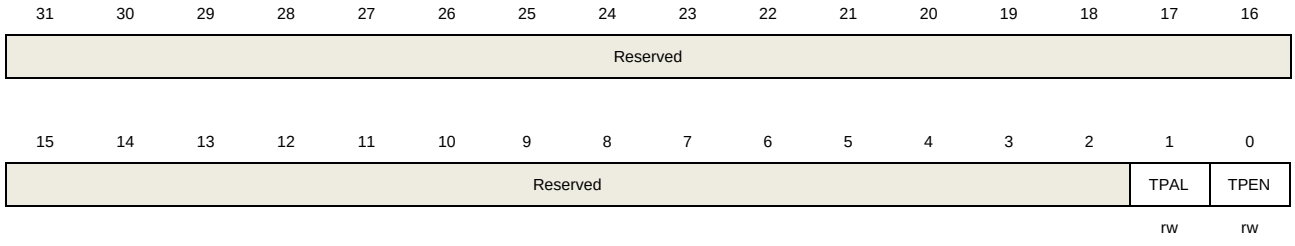
14	CCOSEL	RTC clock output selection 0: RTC clock div 64 1: RTC clock This bit is reset only by a POR.
13:10	Reserved	Must be kept at reset value.
9	ROSEL	RTC output selection 0: RTC alarm pulse is selected as the RTC output 1: RTC second pulse is selected as the RTC output This bit is reset only by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

7.4.3. Tamper pin control register (BKP_TPCTL)

Address offset: 0x30

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.

1	TPAL	TAMPER pin active level 0: The TAMPER pin is active high 1: The TAMPER pin is active low
0	TPEN	TAMPER detection enable 0: The TAMPER pin is free for GPIO functions 1: The TAMPER pin is dedicated for the Backup Reset function. The active level on the TAMPER pin resets all data of the BKP_DATAx register.

7.4.4. Tamper control and status register (BKP_TPCS)

Address offset: 0x34

Reset value: 0x0000 0000

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

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

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

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

0: No effect

1: Reset the TEF bit

This bit is always read as 0.

8. Reset and clock unit (RCU)

8.1. Reset control unit (RCTL)

8.1.1. Overview

GD32F50x reset control includes the control of three kinds of reset: power reset, system reset and backup domain reset. The power reset, known as a cold reset, resets the full system except the backup domain. The system reset resets the processor core and peripheral IP components except for the SW-DP controller and the backup domain. The 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.

8.1.2. 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 whose active signal is low, it will be de-asserted when the internal LDO voltage regulator is ready to provide V_{CORE} power domain. The reset service routine vector is fixed at address 0x0000_0004 in the memory map.

System reset

A system reset is generated by the following events:

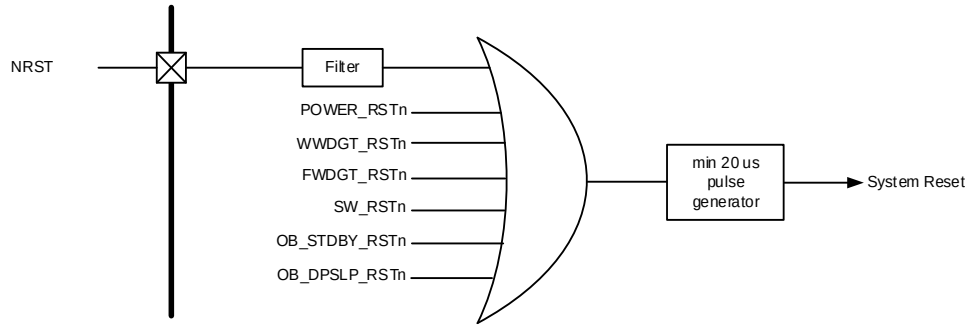
- A power reset (POWER_RSTn).
- A external pin reset (NRST).
- A window watchdog timer reset (WWDGT_RSTn).
- A free watchdog timer reset (FWDGT_RSTn).
- The SYSRESETREQ bit in Cortex®-M33 application interrupt and reset control register is set (SW_RSTn).
- Reset generated when entering Standby mode when resetting nRST_STDBY bit in user option bytes (OB_STDBY_RSTn).
- Reset generated when entering Deep-sleep mode when resetting nRST_DPSLP bit in user option bytes (OB_DPSLP_RSTn).

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 8-1. 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).

8.2. Clock control unit (CCTL)

8.2.1. Overview

The clock control unit provides a range of frequencies and clock functions. These include a Internal 8M RC oscillator (IRC8M), a Internal 48M RC oscillator (IRC48M), a High Speed crystal oscillator (HXTAL), a Low Speed Internal 40K RC oscillator (IRC40K), a Low Speed crystal oscillator (LXTAL), two Phase Lock Loop (PLL), a HXTAL clock monitor, a LXTAL clock monitor, clock frequency monitor, clock prescalers, clock multiplexers and clock gating circuitry.

The clocks of the AHB, APB and Cortex®-M33 are derived from the system clock (CK_SYS) which can source from the IRC8M, HXTAL or PLL0P. The maximum operating frequency of the system clock (CK_SYS) can be up to 200 MHz (for GD32F502xx), 252 MHz (for GD32F503xx) or 280 MHz (for GD32F505xx).

Note: When switching between low and high frequencies (e.g., 8MHz and 280MHz), a switching frequency patch is required. The specific implementation can refer to the relevant code in the firmware library system_gd32f50x.c and the clock switching configuration description can refer to the "AN250 GD32 Clock Switching Configuration User Guide".

Figure 8-2. Clock tree (for GD32F502xx)

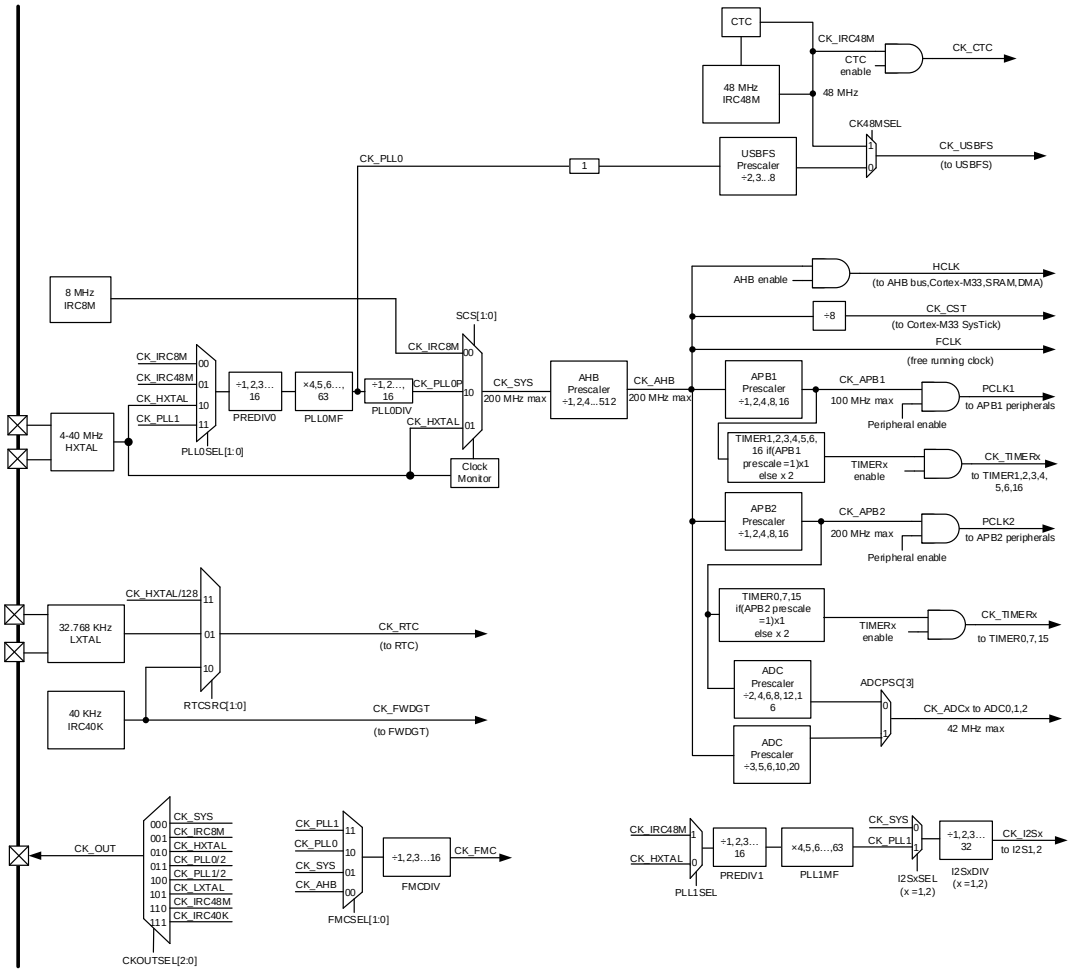


Figure 8-3. Clock tree (for GD32F503xx)

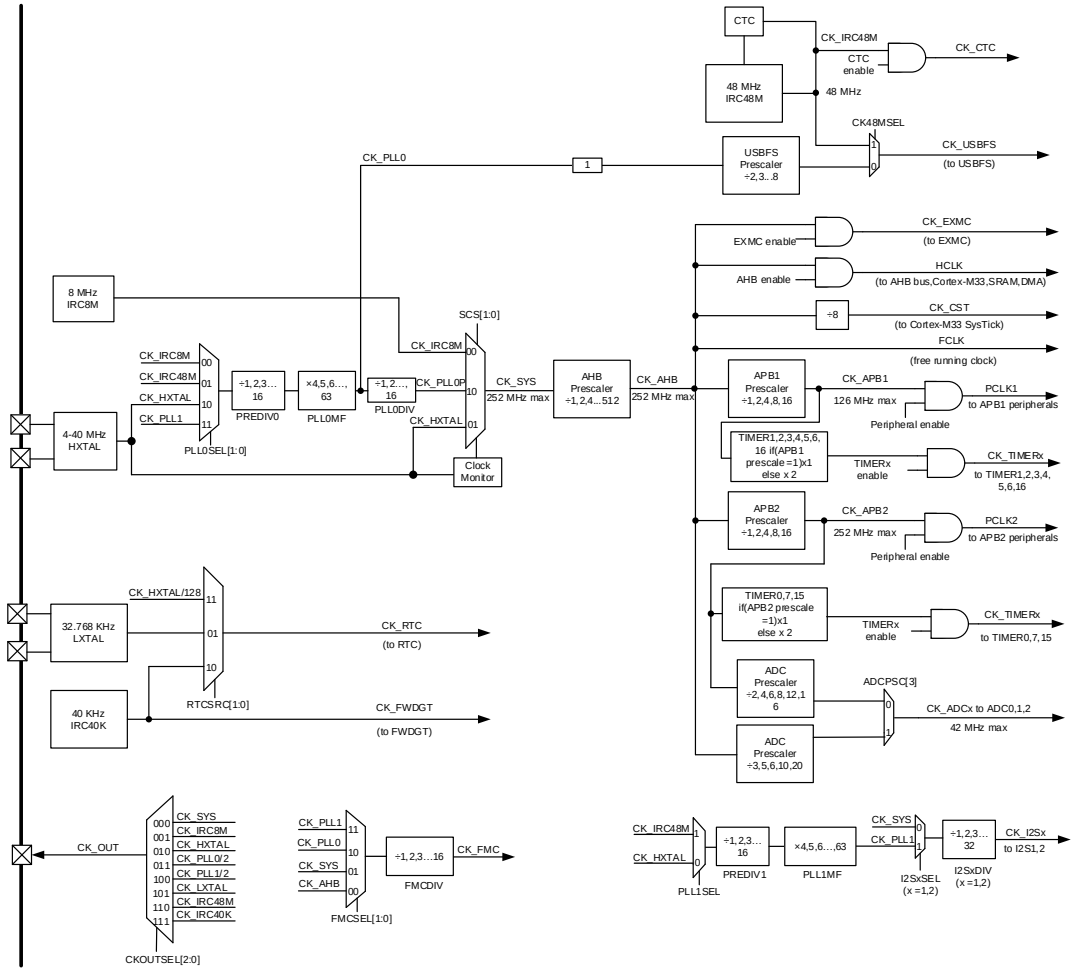
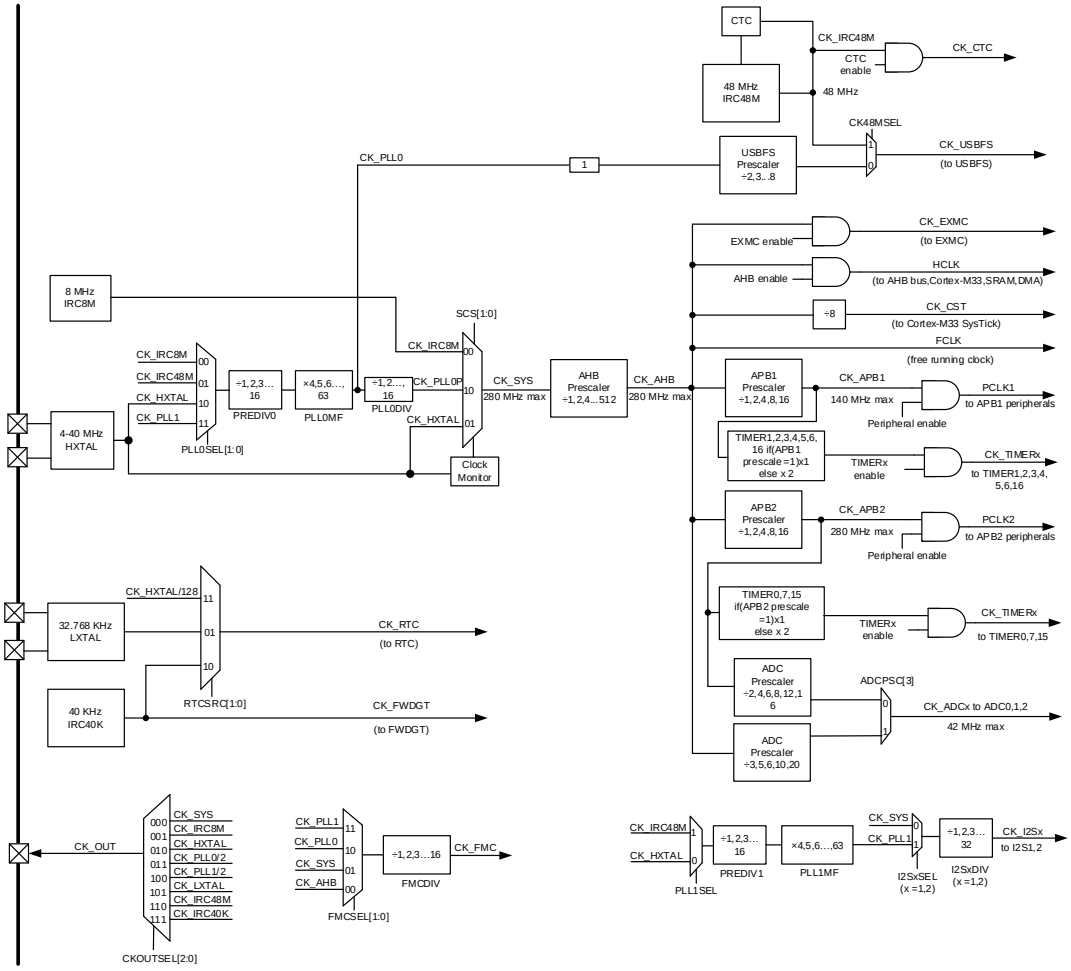


Figure 8-4. Clock tree (for GD32F505xx)



The frequency of AHB, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB, APB2 and APB1 domains is 200 MHz / 200 MHz / 100 MHz (for GD32F502xx), 252 MHz / 252 MHz / 126 MHz (for GD32F503xx) or 280 MHz / 280 MHz / 140 MHz (for GD32F505xx). The cortex® system timer (systick) external clock is clocked with the AHB clock (HCLK) divided by 8. The systick can work either with this clock or with the AHB clock (HCLK), configurable in the systick control and status register.

The ADCs are clocked by the clock of APB2 divided by 2, 4, 6, 8, 12, 16 or by the clock of AHB divided by 3, 5, 6, 10, 20, which defined by ADCPSC in RCU_CFG0 and RCU_CFG1 register.

The TIMERs are clocked by the clock divided from CK_APB2 and CK_APB1. The frequency of TIMERs clock is equal to CK_APBx (APB prescaler is 1), twice the CK_APBx (APB prescaler is not 1).

The USBFS is clocked by the clock of CK48M. The CK48M is selected from the clock of CK_PLL0 or the clock of IRC48M by CK48MSEL bit in RCU_ADDCTL register.

The CTC is clocked by the clock of IRC48M. The IRC48M can be automatically trimmed by CTC unit.

The I2S is clocked by the clock of CK_SYS or CK_PLL1 which defined by I2SxSEL (x =1, 2) bit in RCU_CFG1 register.

The FMC interface clock can be selected as one of CK_AHB, CK_SYS, CK_PLL0, and CK_PLL1 through the FMCSEL bit field in the RCU_ADDCTL register, and is provided after division by FMCDIV.

The RTC is clocked by LXTAL clock or IRC40K clock or HXTAL clock divided by 128 (defined which select by RTCSRC bit in backup domain control register (RCU_BDCTL). After the RTC select HXTAL clock divided by 128, the clock disappeared when the V_{CORE} domain power off. After the RTC select IRC40K, the clock disappeared when V_{DD} power off. After the RTC select LXTAL, the clock disappeared when V_{DD} and V_{BAT} power off.

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

8.2.2. Characteristics

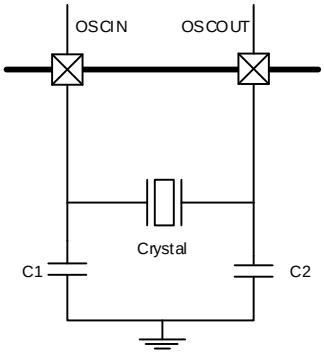
- 4 to 40 MHz high speed crystal oscillator (HXTAL).
- Internal 8 MHz RC oscillator (IRC8M).
- Internal 48 MHz RC oscillator (IRC48M).
- 32,768 Hz Low Speed crystal oscillator (LXTAL).
- Internal 40KHz RC oscillator (IRC40K).
- PLL0 clock source can be HXTAL, IRC8M or IRC48M.
- HXTAL clock monitor.
- LXTAL clock monitor.
- Clock frequency monitor.

8.2.3. Function overview

High speed crystal oscillator (HXTAL)

The high speed external crystal oscillator (HXTAL), which has a frequency from 4 to 40 MHz, produces a highly accurate clock source for use as the system clock. A crystal with a specific frequency must be connected and located close to the two HXTAL pins. The external resistor and capacitor components connected to the crystal are necessary for proper oscillation.

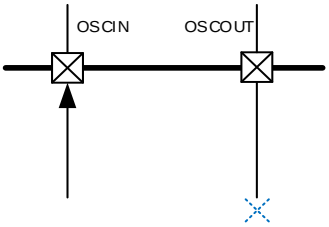
Figure 8-5. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register RCU_CTL. The HXTALSTB flag in control register RCU_CTL indicates if the high-speed external crystal oscillator is stable. When the HXTAL is powered up, it will not be released for use until this HXTALSTB bit is set by the hardware. This specific delay period is known as the oscillator “Start-up time”. As the HXTAL becomes stable, an interrupt will be generated if the related interrupt enable bit HXTALSTBIE in the interrupt register RCU_INT is set. At this point the HXTAL clock can be used directly as the system clock source or the PLL0 input clock.

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the control register RCU_CTL. During bypass mode, the signal is connected to OSCIN, and OSCOUT remains in the suspended state, as shown in ***Figure 8-6. HXTAL clock source in bypass mode***. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 8-6. HXTAL clock source in bypass mode



Internal 8M RC oscillators (IRC8M)

The internal 8M RC oscillator, IRC8M, has a fixed frequency of 8 MHz and is the default clock source selection for the CPU when the device is powered up. The IRC8M oscillator provides a lower cost type clock source as no external components are required. The IRC8M RC oscillator can be switched on or off using the IRC8MEN bit in the control register RCU_CTL. The IRC8MSTB flag in the control register RCU_CTL is used to indicate if the internal 8M 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 clock interrupt register, RCU_INT, is set when the IRC8M becomes stable. The IRC8M clock can also be used as the system clock source or the PLL0 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 PLL0P 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.

Internal 48M RC oscillators (IRC48M)

The internal 48M RC oscillator, IRC48M, has a fixed frequency of 48 MHz. The IRC48M oscillator provides a lower cost type clock source as no external components are required when USBFS used. The IRC48M RC oscillator can be switched on or off using the IRC48MEN

bit in the RCU_ADDCTL register. The IRC48MSTB flag in the RCU_ADDCTL register is used to indicate if the internal 48M RC oscillator is stable. An interrupt can be generated if the related interrupt enable bit, IRC48MSTBIE, in the RCU_ADDINT register, is set when the IRC48M becomes stable. The IRC48M clock is used for the clocks of USBFS.

The frequency accuracy of the IRC48M can be calibrated by the manufacturer, but its operating frequency is still not enough accurate because the USB need the frequency must between 48MHz with 500ppm accuracy. A hardware automatically dynamic trim performed in CTC unit adjust the IRC48M to the needed frequency.

Phase locked loop (PLL)

There are two internal Phase Locked Loop, the PLL0 and PLL1.

The PLL0 can be switched on or off by using the PLL0EN bit in the RCU_CTL register. The PLL0STB flag in the RCU_CTL register will indicate if the PLL0 clock is stable. An interrupt can be generated if the related interrupt enable bit, PLL0STBIE, in the RCU_INT register, is set as the PLL0 becomes stable.

The PLL1 can be switched on or off by using the PLL1EN bit in the RCU_CTL register. The PLL1STB flag in the RCU_CTL register will indicate if the PLL1 clock is stable. An interrupt can be generated if the related interrupt enable bit, PLL1STBIE, in the RCU_INT register, is set as the PLL1 becomes stable.

The two PLLs are closed by hardware when entering the Deepsleep / Standby mode or HXTAL monitor fail when HXTAL used as the source clock of the PLLs.

Low speed crystal oscillator (LXTAL)

The low speed external 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 40K RC oscillator (IRC40K)

The internal 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 clock interrupt register (RCU_INT) is set when the IRC40K becomes stable.

The frequency accuracy of the IRC40K can be calibrated by the manufacturer.

System clock (CK_SYS) selection

After the system reset, the default CK_SYS source will be IRC8M and can be switched to HXTAL or CK_PLL0P by changing the system clock switch bits, SCS, in the clock 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 directly or indirectly (by PLL0) used as the CK_SYS, it is not possible to stop it.

HXTAL clock monitor (HCKM)

The HXTAL clock monitor function is enabled by the HXTAL clock monitor enable bit, HCKMEN, in the control register (RCU_CTL). This function should be enabled after the HXTAL start-up delay and disabled when the HXTAL is stopped. Once the HXTAL failure is detected, the HXTAL will be automatically disabled. The HXTAL clock stuck interrupt flag, HCKMIF, in the clock 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®-M33. If the HXTAL is selected as the clock source of CK_SYS, PLL and CK_RTC, the HXTAL failure will force the CK_SYS source to IRC8M, the PLL will be disabled automatically. If the HXTAL is selected as the clock source of PLL, the HXTAL failure will force the PLL closed automatically. If the HXTAL is selected as the clock source of RTC, the HXTAL failure will reset the RTC clock selection.

LXTAL clock monitor (LCKM)

A clock monitor on LXTAL can be activated by software writing the LCKMEN bit in the backup domain control register (RCU_BDCTL). LCKMEN can not be enabled before LXTAL is enabled and ready.

The clock monitor on LXTAL is working in all modes except V_{BAT}. If a failure is detected on the external 32 KHz oscillator, an interrupt can be sent to CPU.

The software must then disable the LCKMEN bit, stop the defective external 32 KHz oscillator, and change the RTC clock source, or take any required action to secure the application.

A 4-bits plus one counter will work at IRC40K domain when LCKMD enable. If the LXTAL clock has stuck at 0/1 error or slow down about 20KHz, the counter will overflow. The LXTAL clock failure will been found. Once the LXTAL failure is detected, the LXTAL clock stuck interrupt flag, LCKMIF, in the clock interrupt register, RCU_INT, will be set and the LXTAL failure event will be generated.

Clock frequency monitor (CKFM)

The clock frequency monitor can use IRC48M to monitor IRC8M, HXTAL, PLL0P and PLL1 clock frequency range. For IRC8M and HXTAL, 1000 IRC48M clock cycle is used as the monitor window. For PLL0P and PLL1, 100 IRC48M clock cycle is used as the monitor window. User can configure the monitoring range of the clock frequency by configuring the RCU_CKFMCFGx (x = 0, 1, 2, 3) register. If the corresponding interrupt is enabled and the clock frequency failure flag is set, the interrupt will be occurred.

When the IRC48M clock is disabled or lost, the clock frequency monitor is invalid.

Clock output capability

There are several clock signals can be selected via the CK_OUT clock source selection bits, CKOUTSEL, in the clock 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 8-1. Clock output source select

Clock Source Selection bits	Clock Source
000	CK_SYS
001	CK_IRC8M
010	CK_HXTAL
011	CK_PLL0/2
100	CK_PLL1/2
101	CK_LXTAL
110	CK_IRC48M
111	CK_IRC40K

Voltage control

The V_{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 8-2. V_{CORE} domain voltage selected in deep-sleep mode

DSLPVS[2:0]	Deep-sleep mode voltage(V)
000	default value
001	default value – 0.05
010	default value – 0.1
011	default value – 0.15
100	default value – 0.2
101	default value – 0.25
110	default value – 0.3
111	default value – 0.35

8.3. Register definition

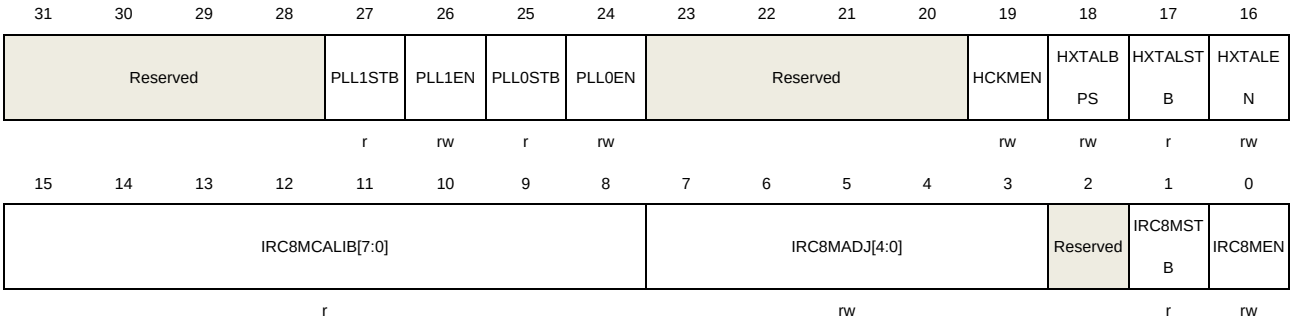
RCU base address: 0x4002 1000

8.3.1. Control register (RCU_CTL)

Address offset: 0x00

Reset value: 0x0000 xx83 where x is undefined.

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	PLL1STB	PLL1 clock stabilization flag Set by hardware to indicate if the PLL1 output clock is stable and ready for use. 0: PLL1 is not stable 1: PLL1 is stable
26	PLL1EN	PLL1 enable Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL1 is switched off 1: PLL1 is switched on
25	PLL0STB	PLL0 clock stabilization flag Set by hardware to indicate if the PLL0 output clock is stable and ready for use. 0: PLL0 is not stable 1: PLL0 is stable
24	PLL0EN	PLL0 enable Set and reset by software. This bit cannot be reset if the PLL0 clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL0 is switched off 1: PLL0 is switched on
23:20	Reserved	Must be kept at reset value.
19	HCKMEN	HXTAL clock monitor enable

		0: Disable the High speed 4 ~ 40 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the High speed 4 ~ 40 MHz crystal oscillator (HXTAL) clock monitor When the hardware detects that the HXTAL clock is stuck at a low or high state, the internal hardware will switch the system clock to be the internal high speed IRC8M RC clock. The way to recover the original system clock is by either an external reset, power on reset or clearing CKMIF by software. **Note**: When the HXTAL clock monitor is enabled, the hardware will automatically enable the IRC8M internal RC oscillator regardless of the control bit, IRC8MEN, state.
18	HXTALBPS	High speed 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	High speed 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	High speed crystal oscillator (HXTAL) Enable Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock or the PLL0 input clock when PLL0 clock is selected to the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: High speed 4 ~ 40 MHz crystal oscillator disabled 1: High speed 4 ~ 40 MHz crystal oscillator enabled
15:8	IRC8MCALIB[7:0]	Internal 8MHz RC oscillator calibration value register These bits are load automatically at power on.
7:3	IRC8MADJ[4:0]	Internal 8MHz RC oscillator clock trim adjust value These bits are set by software. The trimming value is these 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 internal 8MHz RC 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 8MHz RC 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 CKMEN is set. 0: Internal 8 MHz RC oscillator disabled

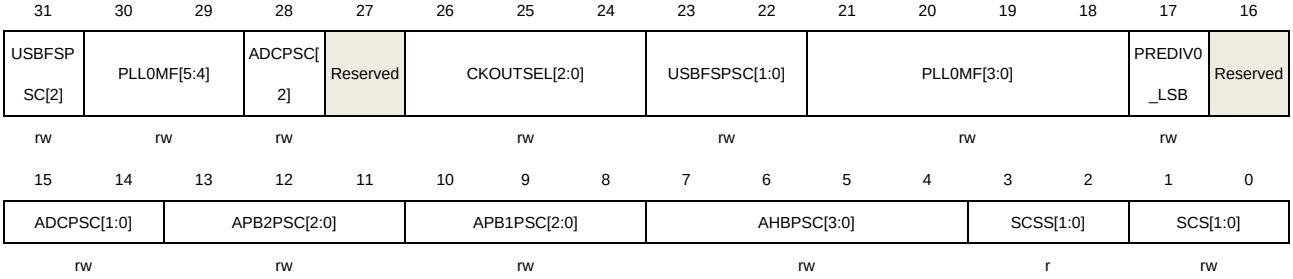
1: Internal 8 MHz RC oscillator enabled

8.3.2. Clock configuration register 0 (RCU_CFG0)

Address offset: 0x04

Reset value: 0x0010 0000

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



Bits	Fields	Descriptions
31	USBFSPSC[2]	Bit 2 of USBFSPSC see bits 23:22 of RCU_CFG0
30:29	PLL0MF[5:4]	Bit 5 and Bit 4 of PLL0MF see bits 21:18 of RCU_CFG0
28	ADCPSC[2]	Bit 2 of ADCPSC see bits 15:14 of RCU_CFG0
27	Reserved	Must be kept at reset value.
26:24	CKOUTSEL[2:0]	CKOUT clock source selection Set and reset by software. 000: System clock selected 001: Internal 8MHz RC Oscillator clock selected 010: External high speed oscillator clock selected 011: (CK_PLL0 / 2) clock selected 100: (CK_PLL1 / 2) clock selected 101: External low speed oscillator clock selected 110: Internal 48MHz RC Oscillator clock selected 111: Internal 40K RC Oscillator clock selected
23:22	USBFSPSC[1:0]	USBFS clock prescaler selection Set and reset by software to control the USBFS clock prescaler value. The USBFS clock must be 48MHz. These bits can't be reset if the USBFS clock is enabled. 000: CK_USBFS = CK_PLL0 / 3 001: CK_USBFS = CK_PLL0 / 2 010: CK_USBFS = CK_PLL0 / 5 011: CK_USBFS = CK_PLL0 / 4 100: CK_USBFS = CK_PLL0 / 6

		101: CK_USBFS = CK_PLL0 / 7 11x :CK_USBFS = CK_PLL0 / 8
21:18	PLL0MF[3:0]	The PLL0 clock multiplication factor Bit 29, bit 30 of RCU_CFG0 and these bits are written by software to define the PLL0 multiplication factor **Note:** The PLL output frequency must not exceed 200 MHz (for GD32F502xx), The PLL output frequency must not exceed 252 MHz (for GD32F503xx), The PLL output frequency must not exceed 280 MHz (for GD32F505xx). 0000xx: reserve 000100: (PLL0 source clock x 4) 000101: (PLL0 source clock x 5) 000110: (PLL0 source clock x 6) ... 111111: (PLL0 source clock x 63)
17	PREDIV0_LSB	The LSB of PREDIV0 division factor This bit is the same bit as PREDIV0 division factor bit [0] from RCU_CFG1. Changing the PREDIV0 division factor bit [0] from RCU_CFG1, this bit is also changed. When the PREDIV0 division factor bits [3:1] are not set, this bit controls PREDIV0 input clock divided by 2 or not.
16	Reserved	Must be kept at reset value.
15:14	ADCPSC[1:0]	ADC clock prescaler selection These bits, bit 28 of RCU_CFG0 and bit 29 of RCU_CFG1 are written by software to define the ADC prescaler factor.Set and cleared by software. 0000: (CK_APB2 / 2) selected 0001: (CK_APB2 / 4) selected 0010: (CK_APB2 / 6) selected 0011: (CK_APB2 / 8) selected 0100: (CK_APB2 / 2) selected 0101: (CK_APB2 / 12) selected 0110: (CK_APB2 / 8) selected 0111: (CK_APB2 / 16) selected 1000: (CK_AHB / 3) selected 1100: (CK_AHB / 5) selected 1x01: (CK_AHB / 6) selected 1x10: (CK_AHB / 10) selected 1x11: (CK_AHB / 20) 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_PLL0P 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 HXTAL failure is detected by HXTAL clock monitor when HXTAL is selected directly or indirectly as the clock source of CK_SYS 00: Select CK_IRC8M as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select CK_PLL0P as the CK_SYS source 11: Reserved

8.3.3. Clock 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						LCKMIF	LCKMIC	HCKMIC	Reserved	PLL1 STBIC	PLL0 STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
						r	w	w		w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKMIE	Reserved	PLL1 STBIE	PLL0 STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	HCKMIF	Reserved	PLL1 STBIF	PLL0 STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF
rw		rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	LCKMIF	LXTAL clock stuck interrupt flag Set by hardware when the LXTAL clock is stuck. Reset when setting the LCKMIC bit by software. 0: Clock operating normally 1: LXTAL clock stuck
24	LCKMIC	LXTAL clock stuck interrupt clear Write 1 by software to reset the LCKMIF flag. 0: Not reset LCKMIF flag 1: Reset LCKMIF flag
23	HCKMIC	HXTAL clock stuck interrupt clear Write 1 by software to reset the HCKMIF flag. 0: Not reset HCKMIF flag 1: Reset HCKMIF flag
22	Reserved	Must be kept at reset value.
21	PLL1STBIC	PLL1 stabilization interrupt clear Write 1 by software to reset the PLL1STBIF flag. 0: Not reset PLL1STBIF flag 1: Reset PLL1STBIF flag
20	PLL0STBIC	PLL0 stabilization interrupt clear Write 1 by software to reset the PLL0STBIF flag. 0: Not reset PLL0STBIF flag 1: Reset PLL0STBIF flag
19	HXTALSTBIC	HXTAL stabilization interrupt clear Write 1 by software to reset the HXTALSTBIF flag. 0: Not reset HXTALSTBIF flag 1: Reset HXTALSTBIF flag
18	IRC8MSTBIC	IRC8M stabilization interrupt clear Write 1 by software to reset the IRC8MSTBIF flag. 0: Not reset IRC8MSTBIF flag

		1: Reset IRC8MSTBIF flag
17	LXTALSTBIC	LXTAL stabilization interrupt clear Write 1 by software to reset the LXTALSTBIF flag. 0: Not reset LXTALSTBIF flag 1: Reset LXTALSTBIF flag
16	IRC40KSTBIC	IRC40K stabilization interrupt clear Write 1 by software to reset the IRC40KSTBIF flag. 0: Not reset IRC40KSTBIF flag 1: Reset IRC40KSTBIF flag
15	LCKMIE	LXTAL clock stuck interrupt enable Set and reset by software to enable/disable the LXTAL clock stuck interrupt. 0: Disable the LXTAL clock stuck interrupt 1: Enable the LXTAL clock stuck interrupt
14	Reserved	Must be kept at reset value.
13	PLL1STBIE	PLL1 stabilization interrupt enable Set and reset by software to enable/disable the PLL1 stabilization interrupt. 0: Disable the PLL1 stabilization interrupt 1: Enable the PLL1 stabilization interrupt
12	PLL0STBIE	PLL0 stabilization interrupt enable Set and reset by software to enable/disable the PLL0 stabilization interrupt. 0: Disable the PLL0 stabilization interrupt 1: Enable the PLL0 stabilization interrupt
11	HXTALSTBIE	HXTAL stabilization interrupt enable Set and reset by software to enable/disable the HXTAL stabilization interrupt 0: Disable the HXTAL stabilization interrupt 1: Enable the HXTAL stabilization interrupt
10	IRC8MSTBIE	IRC8M stabilization interrupt enable Set and reset by software to enable/disable the IRC8M stabilization interrupt 0: Disable the IRC8M stabilization interrupt 1: Enable the IRC8M stabilization interrupt
9	LXTALSTBIE	LXTAL stabilization interrupt enable LXTAL stabilization interrupt enable/disable control 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt
8	IRC40KSTBIE	IRC40K stabilization interrupt enable IRC40K stabilization interrupt enable/disable control 0: Disable the IRC40K stabilization interrupt 1: Enable the IRC40K stabilization interrupt
7	HCKMIF	HXTAL clock stuck interrupt flag

		Set by hardware when the HXTAL clock is stuck. Reset when setting the HCKMIC bit by software. 0: Clock operating normally 1: HXTAL clock stuck
6	Reserved	Must be kept at reset value.
5	PLL1STBIF	PLL1 stabilization interrupt flag Set by hardware when the PLL1 is stable and the PLL1STBIE bit is set. Reset when setting the PLL1STBIC bit by software. 0: No PLL1 stabilization interrupt generated 1: PLL1 stabilization interrupt generated
4	PLL0STBIF	PLL0 stabilization interrupt flag Set by hardware when the PLL0 is stable and the PLL0STBIE bit is set. Reset when setting the PLL0STBIC bit by software. 0: No PLL0 stabilization interrupt generated 1: PLL0 stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the High speed 4 ~ 40 MHz crystal oscillator clock is stable and the HXTALSTBIE bit is set. Reset when setting the HXTALSTBIC bit by software. 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 when setting the IRC8MSTBIC bit by software. 0: No IRC8M stabilization interrupt generated 1: IRC8M stabilization interrupt generated
1	LXTALSTBIF	LXTAL stabilization interrupt flag Set by hardware when the Low speed 32,768 Hz crystal oscillator clock is stable and the LXTALSTBIE bit is set. Reset when setting the LXTALSTBIC bit by software. 0: No LXTAL stabilization interrupt generated 1: LXTAL stabilization interrupt generated
0	IRC40KSTBIF	IRC40K stabilization interrupt flag Set by hardware when the Internal 40kHz RC oscillator clock is stable and the IRC40KSTBIE bit is set. Reset when setting the IRC40KSTBIC bit by software. 0: No IRC40K stabilization clock ready interrupt generated 1: IRC40K stabilization interrupt generated

8.3.4. 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				CAN1RST	CAN0RST	SYSCFGRST	TRIGSELRST	Reserved			TIMER15RST	Reserved			
				rw	rw	rw	rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2RST	USART0RST	TIMER7RST	SPI0RST	TIMER0RST	ADC1RST	ADC0RST	Reserved		PERST	PDRST	PCRST	PBRST	PARST	Reserved	AFRST
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	CAN1RST	CAN1 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN1
26	CAN0RST	CAN0 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN0
25	SYSCFGRST	SYSCFG reset This bit is set and reset by software. 0: No reset 1: Reset the SYSCFG
24	TRIGSELRST	TRIGSEL reset This bit is set and reset by software. 0: No reset 1: Reset the TRIGSEL
23:21	Reserved	Must be kept at reset value.
20	TIMER15RST	TIMER15 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER15
19:16	Reserved	Must be kept at reset value.
15	ADC2RST	ADC2 reset This bit is set and reset by software.

		0: No reset 1: Reset the ADC2
14	USART0RST	USART0 reset This bit is set and reset by software. 0: No reset 1: Reset the USART0
13	TIMER7RST	Timer 7 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER7
12	SPI0RST	SPI0 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI0
11	TIMER0RST	Timer 0 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER0
10	ADC1RST	ADC1 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC1
9	ADC0RST	ADC0 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC0
8:7	Reserved	Must be kept at reset value.
6	PERST	GPIO port E reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port E
5	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port D
4	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port C

3	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port B
2	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset 1: Reset the GPIO port A
1	Reserved	Must be kept at reset value.
0	AFRST	Alternate function I/O reset This bit is set and reset by software. 0: No reset 1: Reset Alternate Function I/O

8.3.5. 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		DACRST	PMURST	BKPIRST	Reserved			CMPRST	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	Reserved
		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
SPI2RST	SPI1RST	Reserved		WWDGTRST	Reserved				TIMER16RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw			rw					rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset DAC unit
28	PMURST	Power control reset This bit is set and reset by software. 0: No reset 1: Reset power control unit
27	BKPIRST	Backup interface reset This bit is set and reset by software.

		0: No reset 1: Reset backup interface
26:24	Reserved	Must be kept at reset value.
23	CMPRST	CMP interface reset This bit is set and reset by software. 0: No reset 1: Reset CMP interface
22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C1
21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C0
20	UART4RST	UART4 reset This bit is set and reset by software. 0: No reset 1: Reset the UART4
19	UART3RST	UART3 reset This bit is set and reset by software. 0: No reset 1: Reset the UART3
18	USART2RST	USART2 reset This bit is set and reset by software. 0: No reset 1: Reset the USART2
17	USART1RST	USART1 reset This bit is set and reset by software. 0: No reset 1: Reset the 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 the SPI2
14	SPI1RST	SPI1 reset This bit is set and reset by software. 0: No reset

		1: Reset the SPI1
13:12	Reserved	Must be kept at reset value.
11	WWDGTRST	WWDGT reset This bit is set and reset by software. 0: No reset 1: Reset the WWDGT
10:7	Reserved	Must be kept at reset value.
6	TIMER16RST	TIMER16 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER16
5	TIMER6RST	TIMER6 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER6
4	TIMER5RST	TIMER5 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER5
3	TIMER4RST	TIMER4 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER4
2	TIMER3RST	TIMER3 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER3
1	TIMER2RST	TIMER2 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER2
0	TIMER1RST	TIMER1 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER1

8.3.6. 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												DMAMUXEN	TRNGEN	HAUEN	CAUEN
												rw	rw	rw	rw

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

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	DMAMUXEN	DMAMUX clock enable This bit is set and reset by software. 0: Disabled DMAMUX clock 1: Enabled DMAMUX clock
18	TRNGEN	TRNG clock enable This bit is set and reset by software. 0: Disabled TRNG clock 1: Enabled TRNG clock
17	HAUEN	HAU clock enable This bit is set and reset by software. 0: Disabled HAU clock 1: Enabled HAU clock
16	CAUEN	CAU clock enable This bit is set and reset by software. 0: Disabled CAU clock 1: Enabled CAU clock
15:13	Reserved	Must be kept at reset value.
12	USBFSEN	USBFS clock enable This bit is set and reset by software. 0: Disabled USBFS clock 1: Enabled USBFS clock
11:9	Reserved	Must be kept at reset value.
8	EXMCEN	EXMC clock enable This bit is set and reset by software. 0: Disabled EXMC clock 1: Enabled EXMC clock

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	DMA1EN	DMA1 clock enable This bit is set and reset by software. 0: Disabled DMA1 clock 1: Enabled DMA1 clock
0	DMA0EN	DMA0 clock enable This bit is set and reset by software. 0: Disabled DMA0 clock 1: Enabled DMA0 clock

8.3.7. 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				CAN1EN	CAN0EN	SYSCFGEN	TRIGSELEN	Reserved			TIMER15EN	Reserved			
				rw	rw	rw	rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2EN	USART0EN	TIMER7EN	SPI0EN	TIMER0EN	ADC1EN	ADC0EN	Reserved		PEEN	PDEN	PCEN	PBEN	PAEN	Reserved	AFEN
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	CAN1EN	CAN1 clock enable This bit is set and reset by software. 0: Disabled CAN1 clock 1: Enabled CAN1 clock
26	CAN0EN	CAN0 clock enable This bit is set and reset by software. 0: Disabled CAN0 clock 1: Enabled CAN0 clock
25	SYSCFGEN	SYSCFG clock enable This bit is set and reset by software. 0: Disabled SYSCFG clock 1: Enabled SYSCFG clock
24	TRIGSELEN	TRIGSEL clock enable This bit is set and reset by software. 0: Disabled TRIGSEL clock 1: Enabled TRIGSEL clock
23:21	Reserved	Must be kept at reset value.
20	TIMER15EN	TIMER15 clock enable This bit is set and reset by software. 0: Disabled TIMER15 clock 1: Enabled TIMER15 clock
19:16	Reserved	Must be kept at reset value.
15	ADC2EN	ADC2 clock enable This bit is set and reset by software. 0: Disabled ADC2 clock 1: Enabled ADC2 clock
14	USART0EN	USART0 clock enable This bit is set and reset by software. 0: Disabled USART0 clock 1: Enabled USART0 clock
13	TIMER7EN	TIMER7 clock enable This bit is set and reset by software. 0: Disabled TIMER7 clock 1: Enabled TIMER7 clock
12	SPI0EN	SPI0 clock enable This bit is set and reset by software. 0: Disabled SPI0 clock

		1: Enabled SPI0 clock
11	TIMER0EN	TIMER0 clock enable This bit is set and reset by software. 0: Disabled TIMER0 clock 1: Enabled TIMER0 clock
10	ADC1EN	ADC1 clock enable This bit is set and reset by software. 0: Disabled ADC1 clock 1: Enabled ADC1 clock
9	ADC0EN	ADC0 clock enable This bit is set and reset by software. 0: Disabled ADC0 clock 1: Enabled ADC0 clock
8:7	Reserved	Must be kept at reset value.
6	PEEN	GPIO port E clock enable This bit is set and reset by software. 0: Disabled GPIO port E clock 1: Enabled GPIO port E clock
5	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
4	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
3	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
2	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
1	Reserved	Must be kept at reset value.
0	AFEN	Alternate function IO clock enable This bit is set and reset by software. 0: Disabled Alternate Function IO clock

1: Enabled Alternate Function IO clock

8.3.8. APB1 enable register (RCU_APB1EN)

Address offset: 0x1C

Reset value: 0x1000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DACEN	PMUEN	BKPIEN	Reserved			CMPEN	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	Reserved
		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
SPI2EN	SPI1EN	Reserved		WWDGTEN	Reserved				TIMER16EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw			rw					rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DACEN	DAC clock enable This bit is set and reset by software. 0: Disabled DAC clock 1: Enabled DAC clock
28	PMUEN	PMU clock enable This bit is set and reset by software. 0: Disabled PMU clock 1: Enabled PMU clock
27	BKPIEN	Backup interface clock enable This bit is set and reset by software. 0: Disabled backup interface clock 1: Enabled backup interface clock
26:24	Reserved	Must be kept at reset value.
23	CMPEN	CMP clock enable This bit is set and reset by software. 0: Disabled CMP clock 1: Enabled CMP 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	UART4EN	UART4 clock enable This bit is set and reset by software. 0: Disabled UART4 clock 1: Enabled UART4 clock
19	UART3EN	UART3 clock enable This bit is set and reset by software. 0: Disabled UART3 clock 1: Enabled UART3 clock
18	USART2EN	USART2 clock enable This bit is set and reset by software. 0: Disabled USART2 clock 1: Enabled USART2 clock
17	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
16	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	WWDGT clock enable This bit is set and reset by software. 0: Disabled WWDGT clock 1: Enabled WWDGT clock
10:7	Reserved	Must be kept at reset value.
6	TIMER16EN	TIMER16 clock enable This bit is set and reset by software. 0: Disabled TIMER16 clock 1: Enabled TIMER16 clock

5	TIMER6EN	TIMER6 clock enable This bit is set and reset by software. 0: Disabled TIMER6 clock 1: Enabled TIMER6 clock
4	TIMER5EN	TIMER5 clock enable This bit is set and reset by software. 0: Disabled TIMER5 clock 1: Enabled TIMER5 clock
3	TIMER4EN	TIMER4 clock enable This bit is set and reset by software. 0: Disabled TIMER4 clock 1: Enabled TIMER4 clock
2	TIMER3EN	TIMER3 clock enable This bit is set and reset by software. 0: Disabled TIMER3 clock 1: Enabled TIMER3 clock
1	TIMER2EN	TIMER2 clock enable This bit is set and reset by software. 0: Disabled TIMER2 clock 1: Enabled TIMER2 clock
0	TIMER1EN	TIMER1 clock enable This bit is set and reset by software. 0: Disabled TIMER1 clock 1: Enabled TIMER1 clock

8.3.9. 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 (RCU_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) is set.

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

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	BKPRST	Backup domain reset This bit is set and reset by software. 0: No reset 1: Resets backup domain
15	RTCEN	RTC clock enable This bit is set and reset by software. 0: Disabled RTC clock 1: Enabled RTC clock
14:10	Reserved	Must be kept at reset value.
9:8	RTCSRC[1:0]	RTC clock entry selection Set and reset by software to control the RTC clock source. Once the RTC clock source has been selected, it cannot be changed anymore unless the backup domain is reset. 00: No clock selected 01: CK_LXTAL selected as RTC source clock 10: CK_IRC40K selected as RTC source clock 11: (CK_HXTAL / 128) selected as RTC source clock
7	LXTALRDYRST	LXTAL ready reset 0: LXTAL ready no reset 1: LXTAL ready reset
6	LCKMD	LXTAL clock failure detection Set by hardware to indicate when a failure has been detected by the clock security system on the external 32 KHz oscillator (LXTAL). It can be clean by disable LCKMEN or disable LXTALEN. 0: No failure detected on LXTAL (32 KHz oscillator) 1: Failure detected on LXTAL (32 KHz oscillator)
5	LCKMEN	LXTAL clock monitor enable 0: Disable the LXTAL clock monitor 1: Enable the LXTAL clock monitor Set by software to enable the clock security system on LXTAL (32 KHz oscillator). LCKMEN should be enabled only on the LXTAL is enabled (LXTALEN bit enabled) and ready (LXTALSTB flag set by hardware). **Note:** Once LCKMEN bit is set, this bit can be reset by backup domain reset or reseting this bit after detecting LXTAL clock failure (LCKMD = 1)
4:3	LXTALDRI[1:0]	LXTAL drive capability Set and reset by software. Backup domain reset resets 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	Low speed crystal oscillator stabilization flag 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

8.3.10. Reset source/clock register (RCU_RSTSCK)

Address offset: 0x24

Reset value: 0x0C00 0000, all reset flags reset by power reset only, RSTFC/IRC40KEN reset by system reset.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	Reserved	RSTFC	Reserved							
r	r	r	r	r	r		rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IRC40K STB	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 reset 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 reset generated. Reset by writing 1 to the RSTFC bit. 0: No external pin reset generated 1: External pin reset generated
25	Reserved	Must be kept at reset value.
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
23:2	Reserved	Must be kept at reset value.
1	IRC40KSTB	IRC40K stabilization flag 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

8.3.11. 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										DMA1RST	DMA0RST	DMAMUXRST	TRNGRST	HAURST	CAURST
										rw	rw	rw	rw	rw	rw

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

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	DMA1RST	DMA0 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA0
20	DMA0RST	DMA1 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA1
19	DMAMUXRST	DMAMUX reset This bit is set and reset by software. 0: No reset 1: Reset the DMAMUX
18	TRNGRST	TRNG reset This bit is set and reset by software. 0: No reset 1: Reset the TRNG
17	HAURST	HAU reset This bit is set and reset by software. 0: No reset 1: Reset the HAU
16	CAURST	CAU reset This bit is set and reset by software. 0: No reset 1: Reset the CAU
15:13	Reserved	Must be kept at reset value.
12	USBFSRST	USBFS reset This bit is set and reset by software.

		0: No reset 1: Reset the USBFS
11:0	Reserved	Must be kept at reset value.

8.3.12. Clock configuration register 1 (RCU_CFG1)

Address offset: 0x2C

Reset value: 0x0000 0400

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		ADCPSC[3]	IRC40KCALIB[4:0]					Reserved		PLL0SEL[1:0]		HXTALRDYRST	I2S2SEL	I2S1SEL	PLL1SEL
		rw	r							rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PLL1MF[5:0]						PREDIV1[3:0]				PREDIV0[3:0]			
		rw						rw				rw			

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	ADCPSC[3]	Bit 4 of ADCPSC see bits 15:14 of RCU_CFG0 and bit 28 of RCU_CFG0
28:24	IRC40KCALIB[4:0]	Internal 40KHz RC oscillator calibration value register These bits are load automatically at power on.
23:22	Reserved	Must be kept at reset value.
21:20	PLL0SEL[1:0]	PLL0 clock source selection Set and reset by software to control the PLL0 clock source. 00: IRC8M clock selected as source clock of PLL0 01: IRC48M clock selected as source clock of PLL0 10: HXTAL clock selected as source clock of PLL0 11: CK_PLL1 clock selected as source clock of PLL0
19	HXTALRDYRST	HXTAL ready reset 0: HXTAL ready no reset 1: HXTAL ready reset
18	I2S2SEL	I2S2 Clock Source Selection Set and reset by software to control the I2S2 clock source. 0: System clock selected as I2S2 source clock 1: (CK_PLL1) selected as I2S2 source clock
17	I2S1SEL	I2S1 Clock Source Selection Set and reset by software to control the I2S1 clock source.

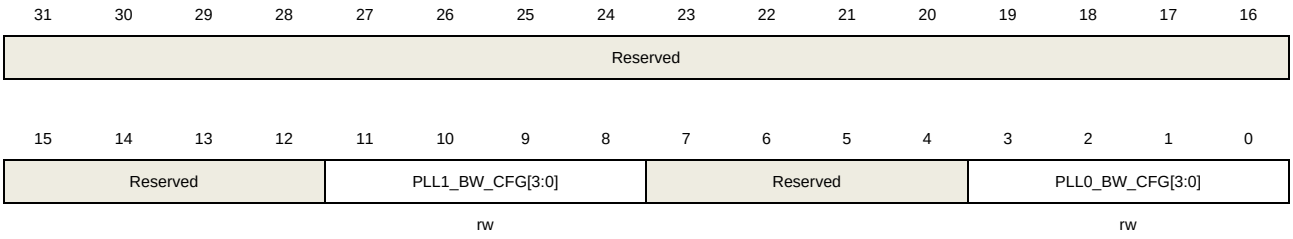
		0: System clock selected as I2S1 source clock 1: (CK_PLL1) selected as I2S1 source clock
16	PLL1SEL	PLL1 clock source selection Set and reset by software to control the PLL1 clock source. 0: HXTAL clock selected as source clock of PLL1 1: IRC48M clock selected as source clock of PLL1
15:14	Reserved	Must be kept at reset value.
13:8	PLL1MF[5:0]	The PLL1 clock multiplication factor Set and reset by software. 0000xx: reserve 000100: PLL1 source clock x 4) 000111: (PLL1 source clock x 5) ... 111111: (PLL1 source clock x 63)
7:4	PREDIV1[3:0]	PREDIV1 division factor This bit is set and reset by software. These bits can be written when PLL1 is disable. 0000: PREDIV1 input source clock not divided 0001: PREDIV1 input source clock divided by 2 0010: PREDIV1 input source clock divided by 3 0011: PREDIV1 input source clock divided by 4 0100: PREDIV1 input source clock divided by 5 0101: PREDIV1 input source clock divided by 6 0110: PREDIV1 input source clock divided by 7 0111: PREDIV1 input source clock divided by 8 1000: PREDIV1 input source clock divided by 9 1001: PREDIV1 input source clock divided by 10 1010: PREDIV1 input source clock divided by 11 1011: PREDIV1 input source clock divided by 12 1100: PREDIV1 input source clock divided by 13 1101: PREDIV1 input source clock divided by 14 1110: PREDIV1 input source clock divided by 15 1111: PREDIV1 input source clock divided by 16
3:0	PREDIV0[3:0]	PREDIV0 division factor This bit is set and reset by software. These bits can be written when PLL0 is disable. **Note:** The bit 0 of PREDIV0 is same as bit 17 of RCU_CFG0, so modifying Bit 17 of RCU_CFG0 also modifies bit 0 of RCU_CFG1. 0000: PREDIV0 input source clock not divided 0001: PREDIV0 input source clock divided by 2 0010: PREDIV0 input source clock divided by 3 0011: PREDIV0 input source clock divided by 4

0100: PREDIV0 input source clock divided by 5
0101: PREDIV0 input source clock divided by 6
0110: PREDIV0 input source clock divided by 7
0111: PREDIV0 input source clock divided by 8
1000: PREDIV0 input source clock divided by 9
1001: PREDIV0 input source clock divided by 10
1010: PREDIV0 input source clock divided by 11
1011: PREDIV0 input source clock divided by 12
1100: PREDIV0 input source clock divided by 13
1101: PREDIV0 input source clock divided by 14
1110: PREDIV0 input source clock divided by 15
1111: PREDIV0 input source clock divided by 16

8.3.13. PLL bandwidth configuration register (RCU_PLLBWCFG)

Address offset: 0x30
Reset value: 0x0000 0000

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

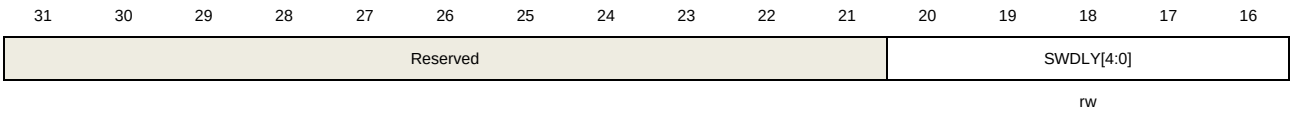


Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:8	PLL1_BW_CFG[3:0]	PLL1 bandwidth configuration Bit[2] and Bit[0] are inverted before to PLL1 analog interface
7:4	Reserved	Must be kept at reset value.
3:0	PLL0_BW_CFG[3:0]	PLL0 bandwidth configuration Bit[2] and Bit[0] are inverted before to PLL0 analog interface

8.3.14. Deep-sleep mode voltage register (RCU_DSV)

Address offset: 0x34
Reset value: 0x000A 0000

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



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

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	SWDLY[4:0]	switch delay time When enter deep-sleep mode, switch to IRC8M clock and wait the SWDLY IRC8M counter and then enter deep-sleep mode. The default is 10 IRC8M clock. Note：don't configure to 0,because it's a sequence.
15:3	Reserved	Must be kept at reset value.
2:0	DSLPVS[2:0]	Deep-sleep mode voltage select These bits are set and reset by software. 000: The core voltage is default value in Deep-sleep mode 001: The core voltage is (default value-0.05)V in Deep-sleep mode 010: The core voltage is (default value-0.1)V in Deep-sleep mode 011: The core voltage is (default value-0.15)V in Deep-sleep mode 100: The core voltage is (default value-0.2)V in Deep-sleep mode 101: The core voltage is (default value-0.25)V in Deep-sleep mode 110: The core voltage is (default value-0.3)V in Deep-sleep mode (customers are not recommended to use it) 111: The core voltage is (default value-0.35)V in Deep-sleep mode (customers are not recommended to use it)

8.3.15. Clock frequency monitor configuration register 0 (RCU_CKFMCFG0)

Address offset: 0x40

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
Reserved										IRC8MCKFFIE	IRC8MCKFFC	IRC8MCKFFF	IRC8MCKFMC[1:0]		IRC8MCKFMEN
										rw	w	r	rw		rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	IRC8MCKFFIE	IRC8M clock frequency failure interrupt enable This bit is set and reset by software.

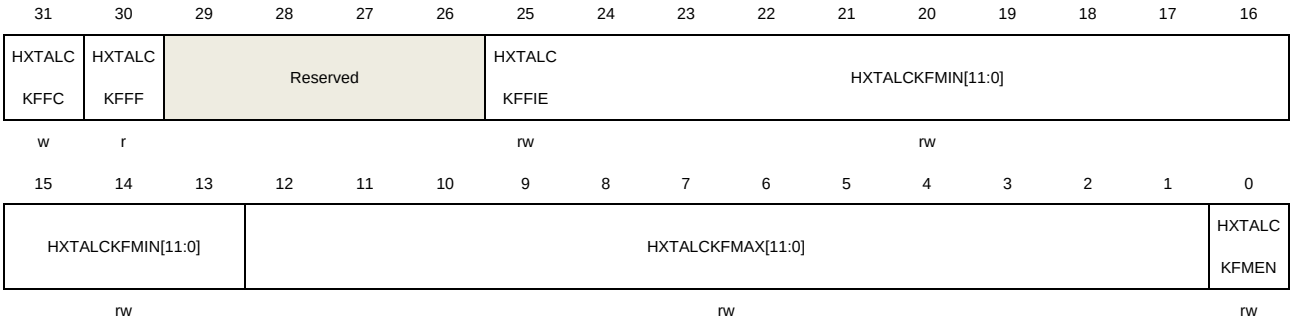
		0: Disable IRC8M clock frequency failure interrupt 1: Enable IRC8M clock frequency failure interrupt
4	IRC8MCKFFC	IRC8M clock frequency failure flag clear Write 1 by software to reset the IRC8MCKFFF flag and this bit is automatically reset by the hardware after IRC8MCKFFF is cleared. 0: Not reset IRC8MCKFFF flag 1: Reset IRC8MCKFFF flag
3	IRC8MCKFFF	IRC8M clock frequency failure flag This bit indicates whether IRC8M clock frequency is out of monitoring range. 0: IRC8M clock frequency is not out of monitoring range 1: IRC8M clock frequency is out of monitoring range
2:1	IRC8MCKFMC[1:0]	IRC8M clock frequency monitor configuration This bit is set and reset by software. 00: The clock frequency monitoring range is 8M Hz ± 5% 01: The clock frequency monitoring range is 8M Hz ± 10% 10: The clock frequency monitoring range is 8M Hz ± 15% 11: The clock frequency monitoring range is 8M Hz ± 20%
0	IRC8MCKFMEN	IRC8M clock frequency monitor enable This bit is set and reset by software. 0: Disable the Internal 8M RC oscillators (IRC8M) clock frequency monitor 1: Enable the Internal 8M RC oscillators (IRC8M) clock frequency monitor

8.3.16. Clock frequency monitor configuration register 1 (RCU_CKFMCFG1)

Address offset: 0x44

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	HXTALCKFFC	HXTAL clock frequency failure flag clear Write 1 by software to reset the HXTALCKFFF flag and this bit is automatically reset by the hardware after HXTALCKFFF is cleared. 0: Not reset HXTALCKFFF flag

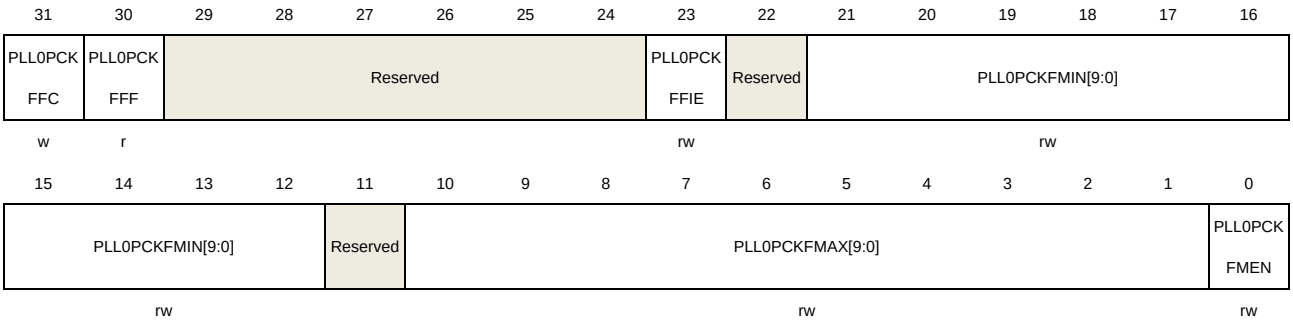
		1: Reset HXTALCKFFF flag
30	HXTALCKFFF	HXTAL clock frequency failure flag This bit indicates whether HXTAL clock frequency is out of monitoring range. 0: HXTAL clock frequency is not out of monitoring range 1: HXTAL clock frequency is out of monitoring range
29:26	Reserved	Must be kept at reset value.
25	HXTALCKFFIE	HXTAL clock frequency failure interrupt enable This bit is set and reset by software. 0: Disable HXTAL clock frequency failure interrupt 1: Enable HXTAL clock frequency failure interrupt
24:13	HXTALCKFMIN[11:0]	HXTAL clock frequency monitor configuration minimum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 1000 IRC48M clock cycle.
12:1	HXTALCKFMAX[11:0]	HXTAL clock frequency monitor configuration maximum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 1000 IRC48M clock cycle.
0	HXTALCKFMEN	HXTAL clock frequency monitor enable This bit is set and reset by software. 0: Disable the HXTAL clock frequency monitor 1: Enable the HXTAL clock frequency monitor

8.3.17. Clock frequency monitor configuration register 2 (RCU_CKFMCFG2)

Address offset: 0x48

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	PLL0PCKFFC	PLL0P clock frequency failure flag clear Write 1 by software to reset the PLL0PCKFFF flag and this bit is automatically reset

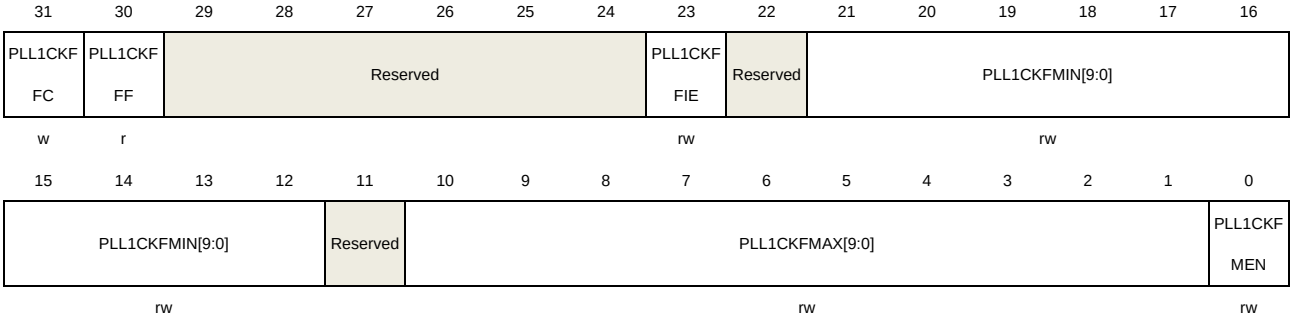
		by the hardware after PLL0PCKFFF is cleared. 0: Not reset PLL0PCKFFF flag 1: Reset PLL0PCKFFF flag
30	PLL0PCKFFF	PLL0P clock frequency failure flag This bit indicates whether PLL0P clock frequency is out of monitoring range. 0: PLL0P clock frequency is not out of monitoring range 1: PLL0P clock frequency is out of monitoring range
29:24	Reserved	Must be kept at reset value.
23	PLL0PCKFFIE	PLL0P clock frequency failure interrupt enable This bit is set and reset by software. 0: disable PLL0P clock frequency failure interrupt 1: enable PLL0P clock frequency failure interrupt
22	Reserved	Must be kept at reset value.
21:12	PLL0PCKFMIN[9:0]	PLL0P clock frequency monitor configuration minimum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 100 IRC48M clock cycle.
11	Reserved	Must be kept at reset value.
10:1	PLL0PCKFMAX[9:0]	PLL0P clock frequency monitor configuration maximum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 100 IRC48M clock cycle.
0	PLL0PCKFMEN	PLL0P clock frequency monitor enable This bit is set and reset by software. 0: Disable the PLL0P clock frequency monitor 1: Enable the PLL0P clock frequency monitor

8.3.18. Clock frequency monitor configuration register 3 (RCU_CKFMCFG3)

Address offset: 0x4C

Reset value: 0x0000 0000

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



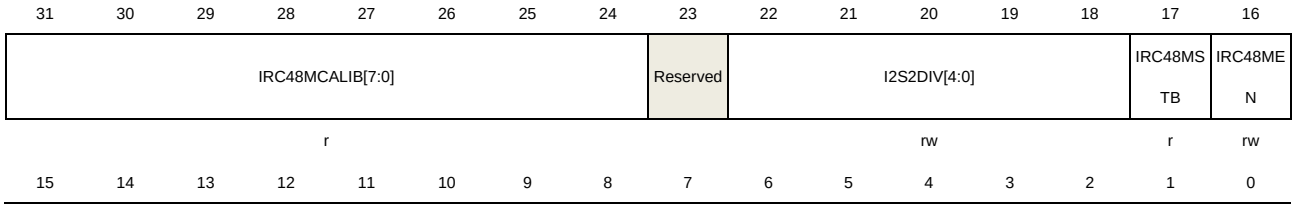
Bits	Fields	Descriptions
31	PLL1CKFFC	PLL1 clock frequency failure flag clear Write 1 by software to reset the PLL1CKFFF flag and this bit is automatically reset by the hardware after PLL1CKFFF is cleared. 0: Not reset PLL1CKFFF flag 1: Reset PLL1CKFFF flag
30	PLL1CKFFF	PLL1 clock frequency failure flag This bit indicates whether PLL1 clock frequency is out of monitoring range. 0: PLL1 clock frequency is not out of monitoring range 1: PLL1 clock frequency is out of monitoring range
29:24	Reserved	Must be kept at reset value.
23	PLL1CKFFIE	PLL1 clock frequency failure interrupt enable This bit is set and reset by software. 0: disable PLL1 clock frequency failure interrupt 1: enable PLL1 clock frequency failure interrupt
22	Reserved	Must be kept at reset value.
21:12	PLL1CKFMIN[9:0]	PLL1 clock frequency monitor configuration minimum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 100 IRC48M clock cycle.
11	Reserved	Must be kept at reset value.
10:1	PLL1CKFMAX[9:0]	PLL1 clock frequency monitor configuration maximum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 100 IRC48M clock cycle.
0	PLL1CKFMEN	PLL1 clock frequency monitor enable This bit is set and reset by software. 0: Disable the PLL1 clock frequency monitor 1: Enable the PLL1 clock frequency monitor

8.3.19. Additional clock control register (RCU_ADDCTL)

Address offset: 0xC0

Reset value: 0x8000 0000

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



FMCDIV[3:0]	PLL0DIV[3:0]	I2S1DIV[4:0]	FMCSEL[1:0]	CK48MSEL
rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	IRC48MCALIB [7:0]	Internal 48MHz RC oscillator calibration value register These bits are load automatically at power on.
23	Reserved	Must be kept at reset value.
22:18	I2S2DIV[4:0]	I2S2 divider The divider is I2S2DIV +1.
17	IRC48MSTB	Internal 48MHz RC oscillator clock stabilization flag Set by hardware to indicate if the IRC48M oscillator is stable and ready for use. 0: IRC48M is not stable 1: IRC48M is stable
16	IRC48MEN	Internal 48MHz RC oscillator enable Set and reset by software. Reset by hardware when entering Deep-sleep or Standby mode. 0: IRC48M disable 1: IRC48M enable
15:12	FMCDIV[3:0]	FMC divider The divider is FMCDIV +1. **Note：** During the lifecycle of the chip, ensure that the FMC interface clock is greater than or equal to the CK_AHB clock, but no more than 7 times the CK_AHB clock, otherwise unpredictable problems may occur.
11:8	PLL0DIV[3:0]	PLL0 divider The divider is PLL0DIV +1.
7:3	I2S1DIV[4:0]	I2S1 divider The divider is I2S1DIV +1.
2:1	FMCSEL[1:0]	FMC clock source selection Set and reset by software to control the FMC clock source 00: CK_AHB selected as FMC source clock 01: CK_SYS selected as FMC source clock 10: CK_PLL0 selected as FMC source clock 11: CK_PLL1 selected as FMC source clock **Note：** During the lifecycle of the chip, ensure that the FMC interface clock is greater than or equal to the CK_AHB clock, but no more than 7 times the CK_AHB clock , otherwise unpredictable problems may occur.
0	CK48MSEL	48MHz clock selection Set and reset by software. This bit used to generate CK48M clock which select IRC48M clock or PLL48M clock.

0: Don't select IRC48M clock (use CK_PLL0 clock divided by USBFSPSC)

1: Select IRC48M clock

8.3.20. Additional clock interrupt register (RCU_ADDINT)

Address offset: 0xCC

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									IRC48MSTBIC	Reserved					
									w						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IRC48MSTBIE	Reserved							IRC48MSTBIF	Reserved					
	rw								r						

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	IRC48MSTBIC	Internal 48 MHz RC oscillator stabilization interrupt clear Write 1 by software to reset the IRC48MSTBIF flag. 0: Not reset IRC48MSTBIF flag 1: Reset IRC48MSTBIF flag
21:15	Reserved	Must be kept at reset value.
14	IRC48MSTBIE	Internal 48 MHz RC oscillator stabilization interrupt enable Set and reset by software to enable/disable the IRC48M stabilization interrupt 0: Disable the IRC48M stabilization interrupt 1: Enable the IRC48M stabilization interrupt
13:7	Reserved	Must be kept at reset value.
6	IRC48MSTBIF	IRC48M stabilization interrupt flag Set by hardware when the Internal 48 MHz RC oscillator clock is stable and the IRC48MSTBIE bit is set. Reset by software when setting the IRC48MSTBIC bit. 0: No IRC48M stabilization interrupt generated 1: IRC48M stabilization interrupt generated
5:0	Reserved	Must be kept at reset value.

8.3.21. APB1 additional reset register (RCU_ADDAPB1RST)

Address offset: 0xE0

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

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	CTCRST	CTC reset This bit is set and reset by software. 0: No reset 1: Reset CTC
26:0	Reserved	Must be kept at reset value.

8.3.22. APB1 additional enable register (RCU_ADDAPB1EN)

Address offset: 0xE4

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

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	CTCEN	CTC clock enable This bit is set and reset by software. 0: Disabled CTC clock 1: Enabled CTC clock
26:0	Reserved	Must be kept at reset value.

8.3.23. LOCK register (RCU_LOCK)

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
LOCK	Reserved														
rw															

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

Bits	Fields	Descriptions
31	LOCK	RCU register lock bit This bit is set and reset by software. 0: RCU register can be configured 1: RCU register can not be configured but read **Note:** HardFault exception will be generated When LOCK = 1 and Writing RCU register.
30:0	Reserved	Must be kept at reset value.

9. Clock trim controller (CTC)

9.1. Overview

The Clock Trim Controller (CTC) is used to trim internal 48MHz RC oscillator (IRC48M) automatically by hardware. The CTC unit trim the frequency of the IRC48M based on an external accurate reference signal source. It can automatically adjust the trim value to provide a precise IRC48M clock.

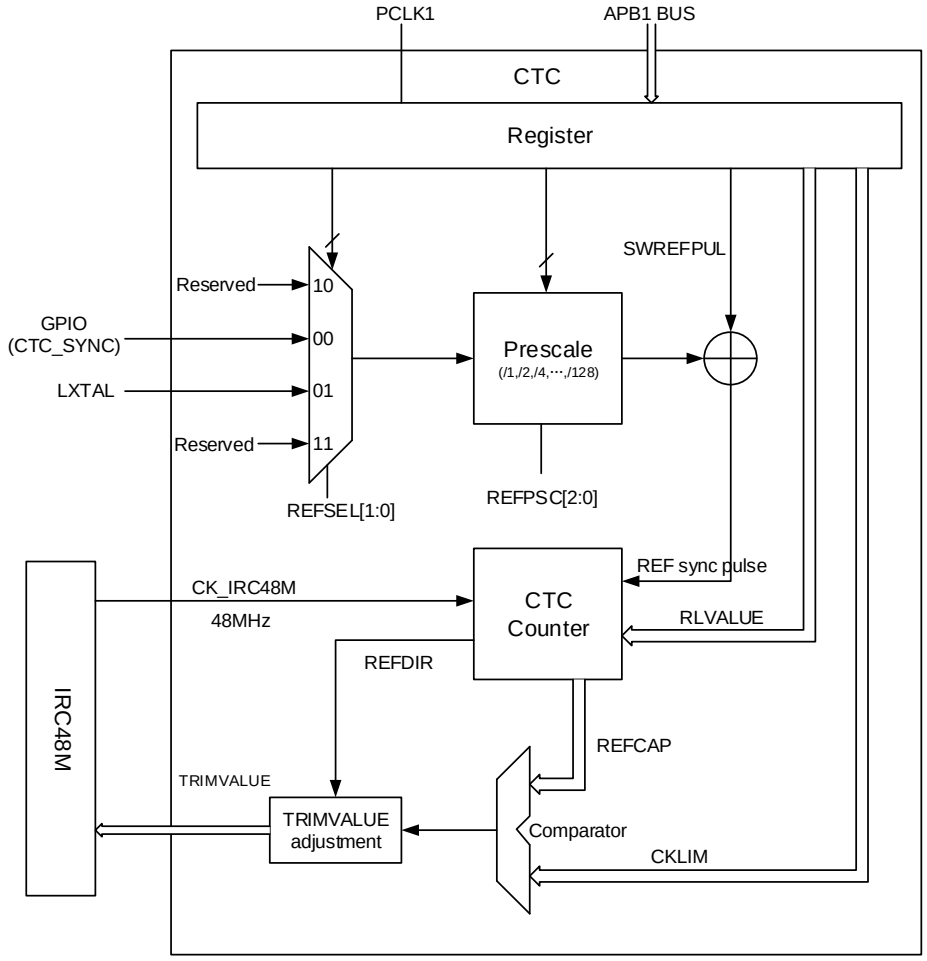
9.2. Characteristics

- Two external reference signal sources: GPIO(CTC_SYNC), LXTAL clock.
- Provide software reference sync pulse.
- Automatically trimmed by hardware without any software action.
- 16 bits trim counter with reference signal source capture and reload.
- 8 bits clock trim base value to frequency evaluation and automatically trim.
- Enough flag or interrupt to indicate the clock is OK (CKOKIF), warning (CKWARNIF) or error (ERRIF).

9.3. Function overview

Figure 9-1. CTC overview provides details on the internal configuration of the CTC.

Figure 9-1. CTC overview



9.3.1. REF sync pulse generator

Firstly, the reference signal source can select GPIO(CTC_SYNC), LXTAL clock by setting REFSEL bits in CTC_CTL1 register.

Secondly, the selected reference signal source uses a configurable polarity by setting REFPOL bit in CTC_CTL1 register, and can be divided to a suitable frequency with a configurable prescaler by setting REFPSC bits in CTC_CTL1 register.

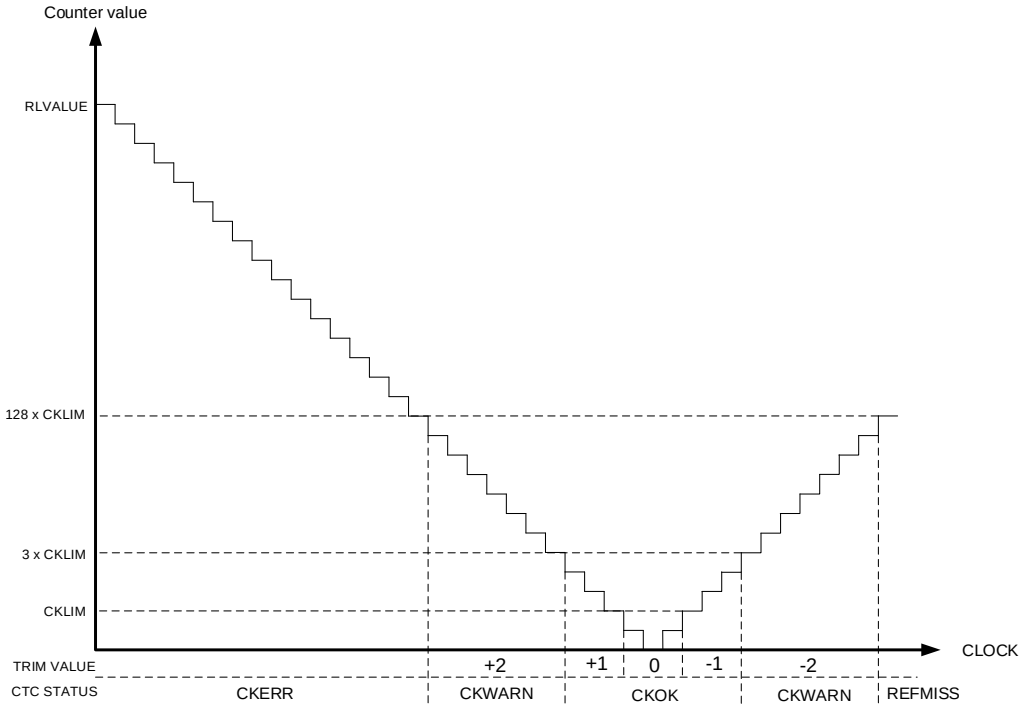
Thirdly, if a software reference pulse needed, write 1 to SWREFPUL bit in CTC_CTL0 register. The software reference pulse generated in last step is logical OR with the external reference pulse.

9.3.2. CTC trim counter

The CTC trim counter is clocked by CK_IRC48M. After CNTEN bit in CTC_CTL0 register set, and a first REF sync pulse detected, the counter start down-counting from RLVALUE (defined in CTC_CTL1 register). If any REF sync pulse detected, the counter reload the RLVALUE and start down-counting again. If no REF sync pulse detected, the counter down-count to zero,

and then up- counting to 128 x CKLIM (defined in CTC_CTL1 register), and then stop until next REF sync pulse detected. If any REF sync pulse detected, the current CTC trim counter value is captured to REFCAP in status register (CTC_STAT), and the counter direction is captured to REFDIR in status register (CTC_STAT). The detail is showing in ***Figure 9-2. CTC trim counter***.

Figure 9-2. CTC trim counter



9.3.3. Frequency evaluation and automatically trim process

The clock frequency evaluation is performed when a REF sync pulse occur. If a REF sync pulse occurs on down-counting, it means the current clock is slower than correct clock (the frequency of 48M). It needs to improve TRIMVALUE in CTC_CTL0 register. If a REF sync pulse occurs on up-counting, it means the current clock is faster than correct clock (the frequency of 48M). It needs to reduce TRIMVALUE in CTC_CTL0 register. The CKOKIF, CKWARNIF, CKERR and REFMISS in CTC_STAT register shows the frequency evaluation scope.

If the AUTOTRIM bit in CTC_CTL0 register is setting, the automatically hardware trim mode enabled. In this mode, if a REF sync pulse occurs on down-counting, it means the current clock is slower than correct clock, the TRIMVALUE will be increased automatically to raise the clock frequency. Vice versa when it occurs on up-counting, the TRIMVALUE will be reduced automatically to reduce the clock frequency.

- Counter < CKLIM when REF sync pulse is detected.

The CKOKIF in CTC_STAT register set, and an interrupt generated if CKOKIE bit in CTC_CTL0 register is 1.

If the AUTOTRIM bit in CTC_CTL0 register set, the TRIMVALUE in CTC_CTL0 register is not

changed.

- CKLIM ≤ Counter < 3 x CKLIM when REF sync pulse is detected.

The CKOKIF in CTC_STAT register set, and an interrupt generated if CKOKIE bit in CTC_CTL0 register is 1.

If the AUTOTRIM bit in CTC_CTL0 register set, the TRIMVALUE in CTC_CTL0 register add 1 when down-counting or sub 1 when up-counting.

- 3 x CKLIM ≤ Counter < 128 x CKLIM when REF sync pulse is detected.

The CKWARNIF in CTC_STAT register set, and an interrupt generated if CKWARNIE bit in CTC_CTL0 register is 1.

If the AUTOTRIM bit in CTC_CTL0 register set, the TRIMVALUE in CTC_CTL0 register add 2 when down-counting or sub 2 when up-counting.

- Counter ≥ 128 x CKLIM when down-counting when a REF sync pulse is detected.

The CKERR in CTC_STAT register set, and an interrupt generated if ERRIE bit in CTC_CTL0 register is 1.

The TRIMVALUE in CTC_CTL0 register is not changed

- Counter = 128 x CKLIM when up-counting.

The REFMISS in CTC_STAT register set, and an interrupt generated if ERRIE bit in CTC_CTL0 register is 1.

The TRIMVALUE in CTC_CTL0 register is not changed.

If adjusting the TRIMVALUE in CTC_CTL0 register over the value of 63, the overflow will be occurred, while adjusting the TRIMVALUE under the value of 0, the underflow will be occurred. The TRIMVALUE is in the range 0 to 63 (the TRIMVALUE is 63 if overflow, the TRIMVALUE is 0 if underflow). Then, the TRIMERR in CTC_STAT register will be set, and an interrupt generated if ERRIE bit in CTC_CTL0 register is 1.

9.3.4. Software program guide

The RLVALUE and CKLIM bits in CTC_CTL1 register is critical to evaluate the clock frequency and automatically hardware trim. The value is calculated by the correct clock frequency (IRC48M:48 MHz) and the frequency of REF sync pulse. The ideal case is REF sync pulse occur when the CTC counter is zero, so the RLVALUE is:

$$RLVALUE = (F_{clock} \div F_{REF}) - 1 \quad (9\text{-}1)$$

The CKLIM is set by user according to the clock accuracy. It is recommended to set to the half of the step size, so the CKLIM is:

$$CKLIM = (F_{clock} \div F_{REF}) \times 0.12\% \div 2 \quad (9\text{-}2)$$

The typical step size is 0.12%. Where the F_{clock} is the frequency of correct clock (IRC48M), the F_{REF} is the frequency of reference sync pulse.

9.4. Register definition

CTC base address: 0x4000 C800

9.4.1. Control register 0 (CTC_CTL0)

Address offset: 0x00
Reset value: 0x0000 2000

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

31	30	29	28	27	26	25	24	23	22	20	19	18	17	16	15
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TRIMVALUE[5:0]						SWREFPUL	AUTOTRIM	CNTEN	Reserved	EREFIE	ERRIE	CKWARNIE	CKOKIE
		rw						w	rw	rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	TRIMVALUE[5:0]	IRC48M trim value When AUTOTRIM in CTC_CTL0 register is 0, these bits are set and cleared by software. This mode used to software calibration. When AUTOTRIM in CTC_CTL0 register is 1, these bits are read only. The value automatically modified by hardware. This mode used to hardware trim. The middle value is 32. When increase 1, the IRC48M clock frequency add around 57KHz. When decrease 1, the IRC48M clock frequency sub around 57KHz.
7	SWREFPUL	Software reference source sync pulse This bit is set by software, and generates a reference sync pulse to CTC counter. This bit is cleared by hardware automatically and read as 0. 0: No effect 1: generates a software reference source sync pulse
6	AUTOTRIM	Hardware automatically trim mode This bit is set and cleared by software. When this bit is set, the hardware automatic trim enabled, the TRIMVALUE bits in CTC_CTL0 register are modified by hardware automatically, until the frequency of IRC48M clock is close to 48MHz. 0: Hardware automatic trim disabled 1: Hardware automatic trim enabled
5	CNTEN	CTC counter enable This bit is set and cleared by software. This bit used to enable or disable the CTC trim counter. When this bit is set, the CTC_CTL1 register cannot be modified. 0: CTC trim counter disabled

		1: CTC trim counter enabled.
4	Reserved	Must be kept at reset value.
3	EREFIE	Expect reference interrupt enable 0: Expect reference interrupt disable 1: Expect reference interrupt enable
2	ERRIE	Error interrupt enable 0: Error interrupt disable 1: Error interrupt enable
1	CKWARNIE	Clock trim warning interrupt enable 0: Clock trim warning interrupt disable 1: Clock trim warning interrupt enable
0	CKOKIE	Clock trim OK interrupt enable 0: Clock trim OK interrupt disable 1: Clock trim OK interrupt enable

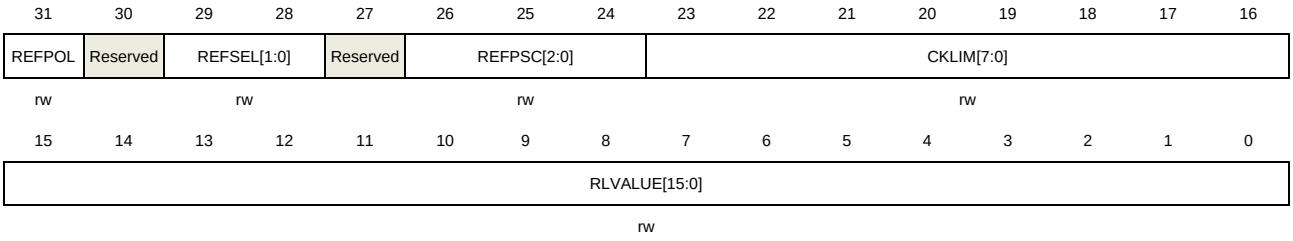
9.4.2. Control register 1 (CTC_CTL1)

Address offset: 0x04

Reset value: 0x2022 BB7F

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

Note: This register cannot be modified when CNTEN is 1.



Bits	Fields	Descriptions
31	REFPOL	Reference signal source polarity This bit is set and cleared by software to select reference signal source polarity 0: rising edge selected 1: falling edge selected
30	Reserved	Must be kept at reset value.
29:28	REFSEL[1:0]	Reference signal source selection These bits are set and cleared by software to select reference signal source. 00: GPIO(CTC_SYNC) selected 01: LXTAL clock selected 10: Reserved

		11: Reserved
27	Reserved	Must be kept at reset value.
26:24	REFPSC[2:0]	Reference signal source prescaler These bits are set and cleared by software 000: Reference signal not divided 001: Reference signal divided by 2 010: Reference signal divided by 4 011: Reference signal divided by 8 100: Reference signal divided by 16 101: Reference signal divided by 32 110: Reference signal divided by 64 111: Reference signal divided by 128
23:16	CKLIM[7:0]	Clock trim base limit value These bits are set and cleared by software to define the clock trim base limit value. These bits used to frequency evaluation and automatically trim process. Please refer to the ***Frequency evaluation and automatically trim process*** for detail.
15:0	RLVALUE[15:0]	CTC counter reload value These bits are set and cleared by software to define the CTC counter reload value. These bits reload to CTC trim counter, when a reference sync pulse is received, so as to start or restart the counter.

9.4.3. Status register (CTC_STAT)

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
REFCAP[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFDIR	Reserved				TRIMERR	REFMISS	CKERR	Reserved				EREFIF	ERRIF	CKWARNIF	CKOKIF
r					r	r	r					r	r	r	r

Bits	Fields	Descriptions
31:16	REFCAP[15:0]	CTC counter capture when reference sync pulse. When a reference sync pulse occurred, the CTC trim counter value is captured to REFCAP bits.
15	REFDIR	CTC trim counter direction when reference sync pulse When a reference sync pulse occurred during the counter is working, the CTC trim

		counter direction is captured to REFDIR bit. 0: Up-counting 1: Down-counting
14:11	Reserved	Must be kept at reset value.
10	TRIMERR	Trim value error bit This bit is set by hardware when the TRIMVALUE in CTC_CTL0 register overflow or underflow. When the ERRIE in CTC_CTL0 register is set, an interrupt occurs. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No trim value error occur 1: Trim value error occur
9	REFMISS	Reference sync pulse miss This bit is set by hardware when the reference sync pulse miss. This is occur when the CTC trim counter reach to 128 x CKLIM during up counting and no reference sync pulse detected. This means the clock is too fast to be trimmed to correct frequency or other error occur. When the ERRIE in CTC_CTL0 register is set, an interrupt occurs. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No Reference sync pulse miss occur 1: Reference sync pulse miss occur
8	CKERR	Clock trim error bit This bit is set by hardware when the clock trim error occurs. This is occur when the CTC trim counter greater or equal to 128 x CKLIM during down counting when a reference sync pulse detected. This means the clock is too slow and cannot be trimmed to correct frequency. When the ERRIE in CTC_CTL0 register is set, an interrupt occurs. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No Clock trim error occur 1: Clock trim error occur
7:4	Reserved	Must be kept at reset value.
3	EREFIF	Expect reference interrupt flag This bit is set by hardware when the CTC counter reach to 0. When the EREFIE in CTC_CTL0 register is set, an interrupt occurs. This bit is cleared by writing 1 to EREFIC bit in CTC_INTC register. 0: No Expect reference occur 1: Expect reference occur
2	ERRIF	Error interrupt flag This bit is set by hardware when an error occurred. If any error of TRIMERR, REFMISS or CKERR occurred, this bit will be set. When the ERRIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No Error occur 1: An error occur

1	CKWARNIF	Clock trim warning interrupt flag This bit is set by hardware when a clock trim warning occurred. If the CTC trim counter greater or equal to 3 x CKLIM and smaller to 128 x CKLIM when a reference sync pulse detected, this bit will be set. This means the clock is too slow or too fast, but can be trim to correct frequency. The TRIMVALUE add 2 or sub 2 when a clock trim warning occurred. When the CKWARNIE in CTC_CTL0 register is set, an interrupt occurs. This bit is cleared by writing 1 to CKWARNIC bit in CTC_INTC register. 0: No Clock trim warning occur 1: Clock trim warning occur
0	CKOKIF	Clock trim OK interrupt flag This bit is set by hardware when the clock trim is OK. If the CTC trim counter smaller to 3 x CKLIM when a reference sync pulse detected, this bit will be set. This means the clock is OK to use. The TRIMVALUE need not to adjust or adjust one step. When the CKOKIE in CTC_CTL0 register is 1, an interrupt occurs. This bit is cleared by writing 1 to CKOKIC bit in CTC_INTC register. 0: No Clock trim OK occur 1: Clock trim OK occur

9.4.4. Interrupt clear register (CTC_INTC)

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												EREFIC	ERRIC	CKWARNIC	CKOKIC
												w	w	w	w

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	EREFIC	EREFIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear EREFIF bit in CTC_STAT register. Write 0 is no effect.
2	ERRIC	ERRIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear ERRIF, TRIMERR, REFMISS and CKERR bits in CTC_STAT register. Write 0 is no effect.
1	CKWARNIC	CKWARNIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear CKWARNIF bit in

		CTC_STAT register. Write 0 is no effect.
0	CKOKIC	CKOKIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear CKOKIF bit in CTC_STAT register. Write 0 is no effect.

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

10.1. Overview

There are up to 80 general purpose I/O pins (GPIO), named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15, PD0 ~ PD15, PE0 ~ PE15 for the device to implement logic input/output functions. Each GPIO port has related control and configuration registers to satisfy the requirements of specific applications. The external interrupt on the GPIO pins of the device have related control and configuration registers in the Interrupt/event Controller Unit (EXIT).

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

Each of the GPIO pins can be configured by software as output (push-pull or open-drain), input, peripheral alternate function or analog mode. Each GPIO pin can be configured as pull-up, pull-down or floating. All GPIOs are high-current capable except for analog mode.

10.2. Characteristics

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

10.3. Function overview

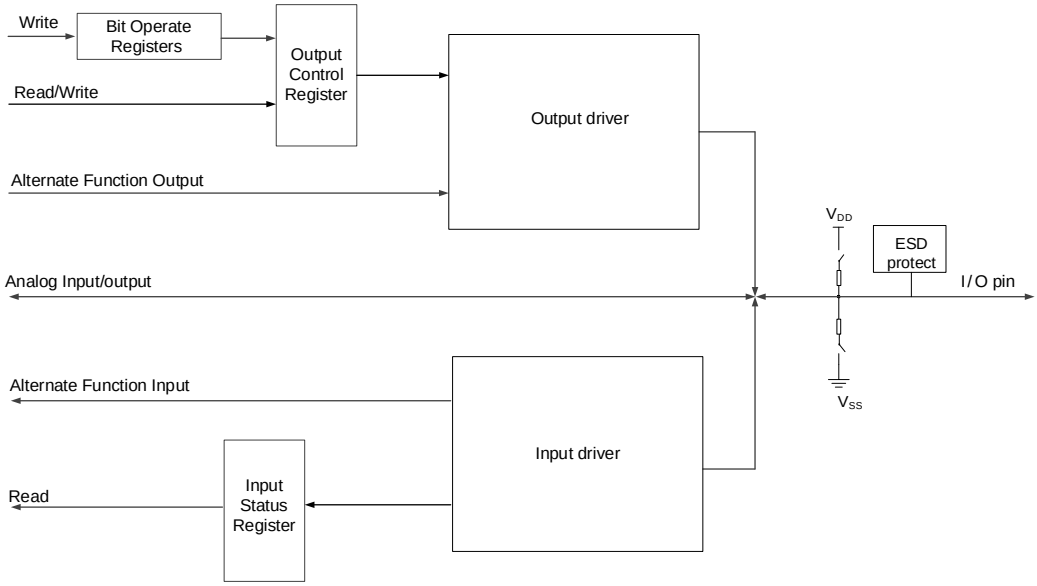
Each of the general-purpose I / O ports can be configured as GPIO inputs, GPIO outputs, AF function or analog mode by GPIO 32-bit configuration registers (GPIOx_CTL). When select AF function, the pad input or output is decided by selected AF function output enable. When the port is output (GPIO output or AFIO output), it can be configured as push-pull or open drain mode by GPIO output mode registers (GPIOx_OMODE). And the port max speed can be configured by GPIO output speed registers (GPIOx_OSPD). Each port can be configured as floating (no pull-up and pull-down), pull-up or pull-down function by GPIO pull-up / pull-down registers (GPIOx_PUD). ***Table 10-1. GPIO configuration table*** shows the details.

Table 10-1. GPIO configuration table

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

Figure 10-1. Basic structure of a standard I/O port bit shows the basic structure of an I/O port bit.

Figure 10-1. Basic structure of a standard I/O port bit



10.3.1. GPIO pin configuration

During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured as the input floating mode without Pull-Up (PU)/Pull-Down (PD) resistors.

But the JTAG/Serial-Wired Debug pins are configured as input PU/PD mode after reset:

PA15: JTDI in PU mode.

PA14: JTCK / SWCLK in PD mode.

PA13: JTMS / SWDIO in PU mode.

PB4: NJTRST in PU mode.

PB3: JTDO in Floating mode.

The GPIO pins can be configured as inputs or outputs. When the GPIO pins are configured as input pins, all GPIO pins have an internal weak pull-up and weak pull-down which can be chosen. And the data on the external pins can be captured at every APB2 clock cycle to the port input status register (GPIOx_ISTAT).

When the GPIO pins are configured as output pins, the user can configure the speed of the ports and choose the output driver mode, push-pull or open-drain mode. The value of the port output control register (GPIOx_OCTL) is output on the I/O pin.

There is no need to read-then-write when programming the GPIOx_OCTL at bit level, the user can modify only one or several bits in a single atomic APB2 write access by programming '1' to the bit operate register (GPIOx_BOP, or for clearing only GPIOx_BC). The other bits will not be affected.

Note:

For LQFP100 package, when the HXTAL function is disabled, the following should be noted when configuring PD0/PD1:

1. When configured as a general GPIO output fucntion, the output of PD0 will both be reflected on PD0 / OSCIN-PD0 pin, and the output of PD1 will both be reflected on PD1 / OSCOUT-PD1 pin.

2. When configured as a general GPIO input fucntion, the status in the corresponding ISTAT will only display the pin status on OSCIN-PD0 / OSCOUT-PD1 and not the status of the PD0 / PD1 pins. If PD0 / PD1 is selected as the EXTI trigger source, triggering can only occur through the OSCIN-PD0 / OSCOUT-PD1 pins.

3. When configured for AF function, with the HXTAL function disabled, the AF8 function only applies to OSCIN-PD0 / OSCOUT-PD1; with the HXTAL function enabled, the AF8 function only applies to PD0 / PD1. Additionally, regardless of whether the HXTAL function is disabled or enabled, AF0~AF7 only apply to PD0 / PD1.

10.3.2. External interrupt/event lines

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

10.3.3. 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.

10.3.4. 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.

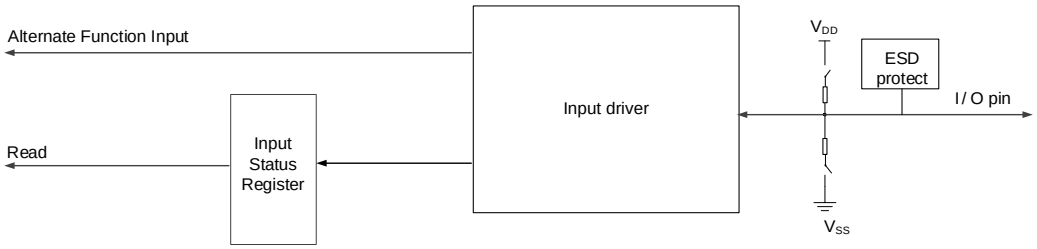
10.3.5. Input configuration

When GPIO pin is configured as Input:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- Every APB2 clock cycle the data present on the I/O pin is got to the port input status register.
- The output buffer is disabled.

Figure 10-2. Basic structure of Input configuration shows the input configuration of the GPIO pin.

Figure 10-2. Basic structure of Input configuration



10.3.6. Output configuration

When GPIO pin is configured as output:

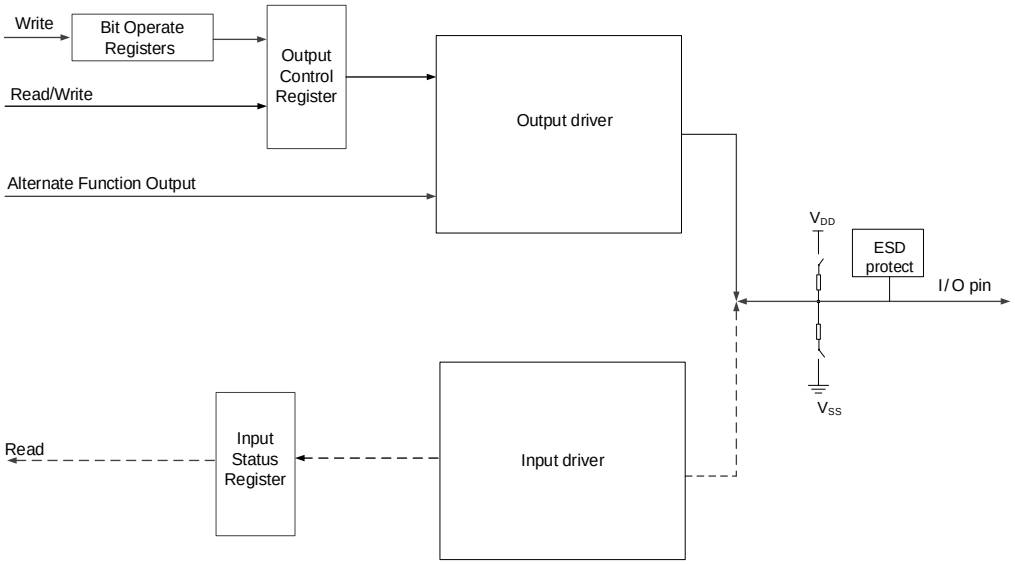
- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen .
- The output buffer is enabled.
- Open Drain Mode, the pad outputs low level when setting “0” in the output control register. while the pad holds Hi-Z when setting “1” in the output control register.
- Push-Pull Mode, the pad outputs low level when setting “0” in the output control register;

while the pad output high level when setting “1” in the output control register.

- A read access to the port output control register gets the last written value.
- A read access to the port input status register gets the I/O state.

Figure 10-3. Basic structure of Output configuration shows the output configuration.

Figure 10-3. Basic structure of Output configuration



10.3.7. Analog configuration

When GPIO pin is used as analog configuration:

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

Figure 10-4. Basic structure of Analog configuration shows the analog configuration.

Figure 10-4. Basic structure of Analog configuration

10.3.8. Alternate function (AF) configuration

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

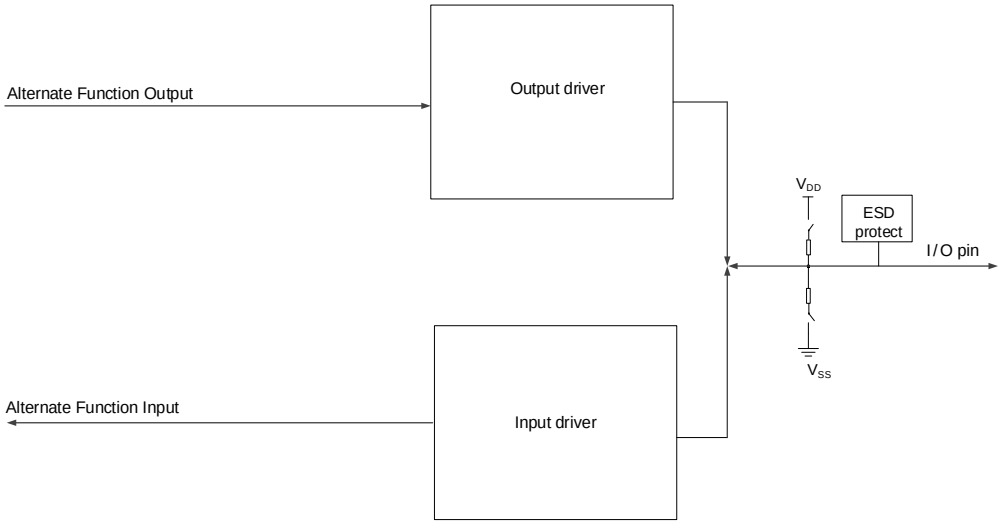
When be configured as alternate function:

- The output buffer is enabled in Open-Drain or Push-Pull configuration.

- The output buffer is driven by the peripheral.
- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen when input.
- The I/O pin data is stored into the port input status register every APB2 clock.
- A read access to the port input status register gets the I/O state.
- A read access to the port output control register gets the last written value.

Figure 10-5. Basic structure of Alternate function configuration shows the alternate function configuration of the GPIO pin.

Figure 10-5. Basic structure of Alternate function configuration



10.3.9. GPIO locking function

The locking mechanism allows the IO configuration to be protected.

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

10.4. Register definition

GPIOA base address: 0x4001 0800

GPIOB base address: 0x4001 0C00

GPIOC base address: 0x4001 1000

GPIOD base address: 0x4001 1400

GPIOE base address: 0x4001 1800

AFIO base address: 0x4001 0000

10.4.1. Port control register (GPIOx_CTL, x = A...E)

Address offset: 0x00

Reset value: 0xABFF FFFF for port A; 0xFFFF FE8F for port B; 0xFFFF FFFF for others.

This register has to be accessed 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: GPIO Input mode 01: GPIO output mode 10: Alternate function mode 11: Analog mode (Input and Output) (reset value)

10.4.2. Port output mode register (GPIOx_OMODE, x = A...E)

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
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

10.4.3. Port output speed register (GPIOx_OSPD, x = A...E)

Address offset: 0x08

Reset value: 0x0C00 0000 for port A; 0x0000 00C0 for port B; 0x0000 0000 for others.

This register has to be accessed 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. 00: Output speed level 0 (reset state) 01: Output speed level 1 10: Output speed level 2 11: Output speed level 3

10.4.4. Port pull-up / pull-down register (GPIOx_PUD, x = A...E)

Address offset: 0x0C

Reset value: 0x6400 0000 for port A; 0x0000 0100 for port B; 0x0000 0000 for others.

This register has to be accessed 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]	Pin 15 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
29:28	PUD14[1:0]	Pin 14 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
27:26	PUD13[1:0]	Pin 13 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
25:24	PUD12[1:0]	Pin 12 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
23:22	PUD11[1:0]	Pin 11 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
21:20	PUD10[1:0]	Pin 10 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.

19:18	PUD9[1:0]	Pin 9 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
17:16	PUD8[1:0]	Pin 8 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
15:14	PUD7[1:0]	Pin 7 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
13:12	PUD6[1:0]	Pin 6 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
11:10	PUD5[1:0]	Pin 5 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
9:8	PUD4[1:0]	Pin 4 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
7:6	PUD3[1:0]	Pin 3 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
5:4	PUD2[1:0]	Pin 2 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
3:2	PUD1[1:0]	Pin 1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description.
1:0	PUD0[1:0]	Pin 0 pull-up or pull-down bits These bits are set and cleared by software. 00: Floating mode, no pull-up and pull-down (reset value) 01: With pull-up mode 10: With pull-down mode 11: Reserved

10.4.5. Port input status register (GPIOx_ISTAT, x = A...E)

Address offset: 0x10
Reset 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	Pin input status (y=0..15) These bits are set and cleared by hardware. 0: Input signal low 1: Input signal high

10.4.6. Port output control register (GPIOx_OCTL, x = A...E)

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
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	Pin output control (y=0..15) These bits are set and cleared by software. 0: Pin output low 1: Pin output high

10.4.7. Port bit operate register (GPIOx_BOP, x = A...E)

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
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

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

BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	CRy	Pin 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 to 0
15:0	BOPy	Pin 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 to 1

10.4.8. Port configuration lock register (GPIOx_LOCK, x = A...E)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	LKK	Lock sequence key It can only be setted using the Lock Key Writing Sequence. And can always be read. 0: GPIO_LOCK register is not locked and the port configuration is not locked. 1: GPIO_LOCK register is locked until an MCU reset. LOCK key configuration sequence. Write 1→Write 0→Write 1→ Read 0→ Read 1 **Note:** The value of LK[15:0] 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: The corresponding bit port configuration is not locked 1: The corresponding bit port configuration is locked when LKK bit is “1”

10.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x = A...E)

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
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

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

0100: AF4 selected
0101: AF5 selected
0110: AF6 selected
0111: AF7 selected
1000: AF8 selected
1001~1111: Reserved

10.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x = A...E)

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
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

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

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

10.4.11. Bit clear register (GPIOx_BC, x = A...E)

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
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	Pin 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

10.4.12. EXTI sources selection register 0 (AFIO_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 0000: PA3 pin 0001: PB3 pin 0010: PC3 pin 0011: PD3 pin 0100: PE3 pin Other configurations are reserved.
11:8	EXTI2_SS [3:0]	EXTI 2 sources selection 0000: PA2 pin 0001: PB2 pin 0010: PC2 pin 0011: PD2 pin 0100: PE2 pin Other configurations are reserved.
7:4	EXTI1_SS [3:0]	EXTI 1 sources selection 0000: PA1 pin 0001: PB1 pin 0010: PC1 pin 0011: PD1 pin 0100: PE1 pin Other configurations are reserved.
3:0	EXTI0_SS [3:0]	EXTI 0 sources selection 0000: PA0 pin 0001: PB0 pin 0010: PC0 pin 0011: PD0 pin 0100: PE0 pin Other configurations are reserved.

10.4.13. EXTI sources selection register 1 (AFIO_EXTISS1)

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
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI7_SS [3:0]	EXTI 7 sources selection 0000: PA7 pin 0001: PB7 pin 0010: PC7 pin 0011: PD7 pin 0100: PE7 pin Other configurations are reserved.
11:8	EXTI6_SS [3:0]	EXTI 6 sources selection 0000: PA6 pin 0001: PB6 pin 0010: PC6 pin 0011: PD6 pin 0100: PE6 pin Other configurations are reserved.
7:4	EXTI5_SS [3:0]	EXTI 5 sources selection 0000: PA5 pin 0001: PB5 pin 0010: PC5 pin 0011: PD5 pin 0100: PE5 pin Other configurations are reserved.
3:0	EXTI4_SS [3:0]	EXTI 4 sources selection 0000: PA4 pin 0001: PB4 pin 0010: PC4 pin 0011: PD4 pin 0100: PE4 pin Other configurations are reserved.

10.4.14. EXTI sources selection register 2 (AFIO_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 0000: PA11 pin 0001: PB11 pin 0010: PC11 pin 0011: PD11 pin 0100: PE11 pin Other configurations are reserved.
11:8	EXTI10_SS [3:0]	EXTI 10 sources selection 0000: PA10 pin 0001: PB10 pin 0010: PC10 pin 0011: PD10 pin 0100: PE10 pin Other configurations are reserved.
7:4	EXTI9_SS [3:0]	EXTI 9 sources selection 0000: PA9 pin 0001: PB9 pin 0010: PC9 pin 0011: PD9 pin 0100: PE9 pin Other configurations are reserved.
3:0	EXTI8_SS [3:0]	EXTI 8 sources selection 0000: PA8 pin 0001: PB8 pin 0010: PC8 pin 0011: PD8 pin 0100: PE8 pin Other configurations are reserved.

10.4.15. EXTI sources selection register 3 (AFIO_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 0000: PA15 pin 0001: PB15 pin 0010: PC15 pin 0011: PD15 pin 0100: PE15 pin Other configurations are reserved.
11:8	EXTI14_SS [3:0]	EXTI 14 sources selection 0000: PA14 pin 0001: PB14 pin 0010: PC14 pin 0011: PD14 pin 0100: PE14 pin Other configurations are reserved.
7:4	EXTI13_SS [3:0]	EXTI 13 sources selection 0000: PA13 pin 0001: PB13 pin 0010: PC13 pin 0011: PD13 pin 0100: PE13 pin Other configurations are reserved.
3:0	EXTI12_SS [3:0]	EXTI 12 sources selection 0000: PA12 pin 0001: PB12 pin 0010: PC12 pin 0011: PD12 pin 0100: PE12 pin Other configurations are reserved.

11. Trigger selection controller (TRIGSEL)

11.1. Overview

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

With TRIGSEL, there are up to 3 trigger selection outputs could be selected for each peripheral. And every output could be selected from different trigger input signal.

11.2. Characteristics

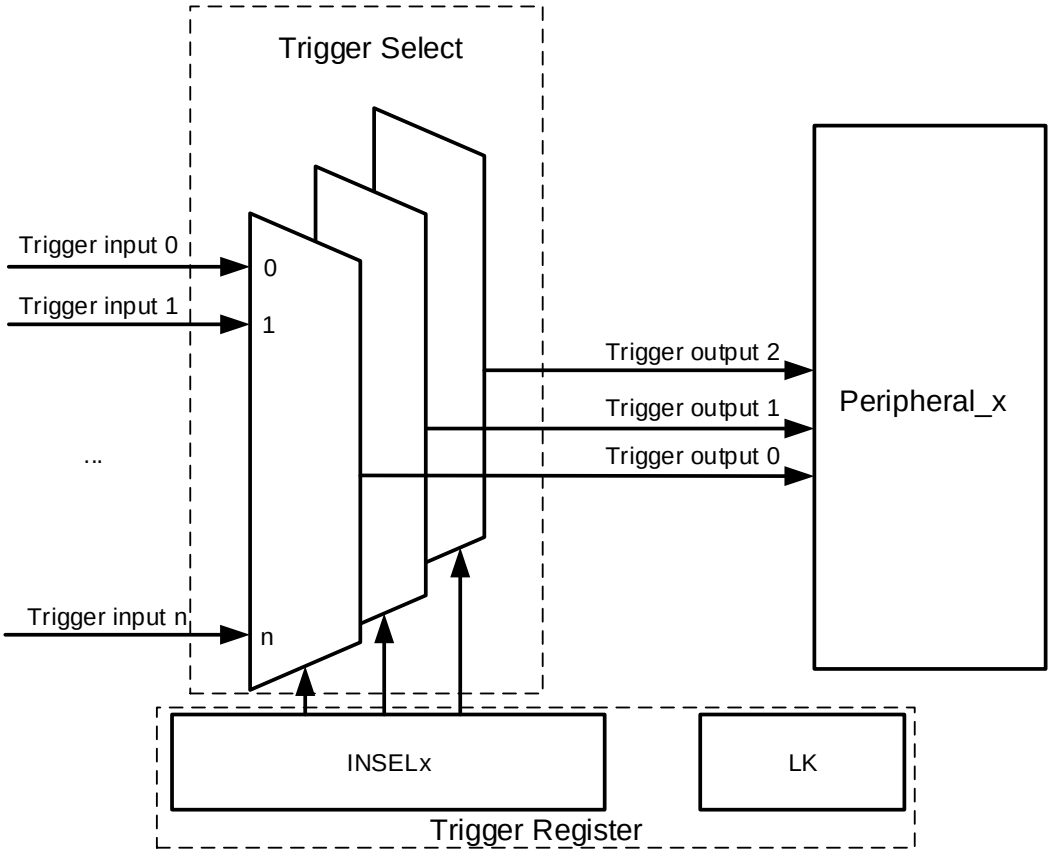
- Support up to 74 trigger input signals.
- Each peripheral has its corresponding register to select trigger input signal.
- TRIGSEL register can be configured up to 3 outputs to peripheral.
- Trigger input source could be external input signal or output of peripheral.
- Trigger selection output could be for external output or peripheral input.

11.3. Function overview

With TRIGSEL, peripherals that support trigger source selection have dedicated registers to select the trigger input source. Each register can be configured with 3 outputs, which are connected to the trigger input of the peripheral. Each output can select different trigger input sources.

The ***Figure 11-1. TRIGSEL main composition example*** shows the main composition of TRIGSEL.

Figure 11-1. TRIGSEL main composition example



11.4. Internal connect

The TRIGSEL allows software to select the trigger input for peripherals. The ***Table 11-1. Trigger input bit fields selection*** shows the trigger input register selection.

Table 11-1. Trigger input bit fields selection

fields	bits value	trigger input selection
INSELx	0x00	V_{SS}
	0x01	V_{DD}
	0x02	TRIGSEL_IN0
	0x03	TRIGSEL_IN1
	0x04	TRIGSEL_IN2
	0x05	TRIGSEL_IN3
	0x06	TRIGSEL_IN4
	0x07	TRIGSEL_IN5
	0x08	TRIGSEL_IN6
	0x09	TRIGSEL_IN7
	0x0a	TIMER0_TRGO
	0x0b	TIMER0_CH0
	0x0c	TIMER0_CH1

fields	bits value	trigger input selection
	0x0d	TIMER0_CH2
	0x0e	TIMER0_CH3
	0x0f	TIMER7_TRGO
	0x10	TIMER7_CH0
	0x11	TIMER7_CH1
	0x12	TIMER7_CH2
	0x13	TIMER7_CH3
	0x14	TIMER5_TRGO
	0x15	TIMER6_TRGO
	0x16	TIMER1_TRGO
	0x17	TIMER1_CH0
	0x18	TIMER1_CH1
	0x19	TIMER1_CH2
	0x1a	TIMER1_CH3
	0x1b	TIMER2_TRGO
	0x1c	TIMER2_CH0
	0x1d	TIMER2_CH1
	0x1e	TIMER2_CH2
	0x1f	TIMER2_CH3
	0x20	TIMER3_TRGO
	0x21	TIMER3_CH0
	0x22	TIMER3_CH1
	0x23	TIMER3_CH2
	0x24	TIMER3_CH3
	0x25	TIMER4_TRGO
	0x26	TIMER4_CH0
	0x27	TIMER4_CH1
	0x28	TIMER4_CH2
	0x29	TIMER4_CH3
	0x2f	TIMER15_TRGO
	0x30	TIMER15_CH0
	0x31	TIMER15_CH1
	0x32	TIMER15_MCH0
	0x33	TIMER16_TRGO
	0x34	TIMER16_CH0
	0x35	TIMER16_CH1
	0x36	TIMER16_MCH0
	0x37	ADC0_WD0_OUT
	0x38	Reserved
	0x39	Reserved
	0x3a	ADC1_WD0_OUT

fields	bits value	trigger input selection
	0x3b	Reserved
	0x3c	Reserved
	0x3d	ADC2_WD0_OUT
	0x3e	Reserved
	0x3f	Reserved
	0x40	CMP0_OUT
	0x41	CK_OUT
	0x42	TIMER0_BKIN
	0x43	TIMER0_CH0BKIN
	0x44	TIMER0_CH1BKIN
	0x45	TIMER0_CH2BKIN
	0x46	TIMER7_BKIN
	0x47	TIMER7_CH0BKIN
	0x48	TIMER7_CH1BKIN
	0x49	TIMER7_CH2BKIN
	0x4a	TIMER15_BKIN
	0x4b	TIMER16_BKIN
	0x4c	EXTI9
	0x4d	EXTI11
	0x4e	EXTI15

Table 11-2. TRIGSEL input and output mapping shows the connection relationship between TRIGSEL input and output. Through the INSELx[7:0] bits of TRIGSEL register, an input trigger source can be selected for the output of TRIGSEL. Each TRIGSEL register can configure with up to 3 outputs, which are connected to the corresponding peripherals.

Table 11-2. TRIGSEL input and output mapping

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
0 1 TRIGSEL_IN0 TRIGSEL_IN1 TRIGSEL_IN2 TRIGSEL_IN3 TRIGSEL_IN4 TRIGSEL_IN5 TRIGSEL_IN6 TRIGSEL_IN7 TIMER0_TRGO TIMER0_CH0	INSELx[7:0]	TRIGSEL_EXTOUT0	Output0 Output1	TRIGSEL_OUT0 TRIGSEL_OUT1
		TRIGSEL_EXTOUT1	Output0 Output1	TRIGSEL_OUT2 TRIGSEL_OUT3
		TRIGSEL_EXTOUT2	Output0 Output1	TRIGSEL_OUT4 TRIGSEL_OUT5

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER7_TRGO		TRIGSEL_EXTOUT3	Output0 Output1	TRIGSEL_OUT6 TRIGSEL_OUT7
TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3		TRIGSEL_TIMER0ITI	Output0	TIMER0_ITI
TIMER5_TRGO TIMER6_TRGO TIMER1_TRGO TIMER1_CH0		TRIGSEL_TIMER1ITI	Output0	TIMER1_ITI
TIMER1_CH1 TIMER1_CH2 TIMER1_CH3 TIMER2_TRGO		TRIGSEL_TIMER2ITI	Output0	TIMER2_ITI
TIMER2_CH0 TIMER2_CH1 TIMER2_CH2 TIMER2_CH3		TRIGSEL_TIMER3ITI	Output0	TIMER3_ITI
TIMER3_TRGO TIMER3_CH0 TIMER3_CH1 TIMER3_CH2		TRIGSEL_TIMER4ITI	Output0	TIMER4_ITI
TIMER3_CH3 TIMER4_TRGO TIMER4_CH0 TIMER4_CH1		TRIGSEL_TIMER7ITI	Output0	TIMER7_ITI
TIMER4_CH2 TIMER4_CH3 TIMER15_TRGO TIMER15_CH0		TRIGSEL_TIMER15ITI	Output0	TIMER15_ITI
TIMER15_CH1 TIMER15_MCH0 TIMER16_TRGO TIMER16_CH0		TRIGSEL_TIMER16ITI	Output0	TIMER16_ITI
TIMER16_CH1 TIMER16_MCH0 ADC0_WD0_OUT ADC1_WD0_OUT		TRIGSEL_DAC	Output0	DAC0_OUT_EXTRG

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
ADC2_WD0_OUT CMP0_OUT CK_OUT TIMER0_BKIN TIMER0_CH0BKIN TIMER0_CH1BKIN TIMER0_CH2BKIN TIMER7_BKIN TIMER7_CH0BKIN TIMER7_CH1BKIN TIMER7_CH2BKIN TIMER15_BKIN TIMER16_BKIN EXTI9 EXTI11 EXTI15		TRIGSEL_ADC0_ROUTRG	Output0	ADC0_ROUTRG
		TRIGSEL_ADC0_INSTRG	Output0	ADC0_INSTRG
		TRIGSEL_ADC1_ROUTRG	Output0	ADC1_ROUTRG
		TRIGSEL_ADC1_INSTRG	Output0	ADC1_INSTRG
		TRIGSEL_ADC2_ROUTRG	Output0	ADC2_ROUTRG
		TRIGSEL_ADC2_INSTRG	Output0	ADC2_INSTRG
		TRIGSEL_TIMER0BRKIN	Output0	TIMER0_BRKIN
		TRIGSEL_TIMER0CHBRKIN	Output0 Output1 Output2	TIMER0_CH0BRKIN TIMER0_CH1BRKIN TIMER0_CH2BRKIN
		TRIGSEL_TIMER7BRKIN	Output0	TIMER7_BRKIN
		TRIGSEL_TIMER7CHBRKIN	Output0 Output1 Output2	TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
		TRIGSEL_TIMER15BRKIN	Output0	TIMER15_BRKIN
		TRIGSEL_TIMER16BRKIN	Output0	TIMER16_BRKIN

Note:

1. TRIGSEL_OUTx (x=0,…,7) cannot select any TRIGSEL_INy (y=0,…,7) as trigger source.
2. TIMERx_ITI (x=0,...,4,7) cannot select signals (TIMERx_TRGO \ TIMERx_CH0 \ TIMERx_CH1 \ TIMERx_CH2 \ TIMERx_CH3) from its own module as trigger sources, and TIMERx_ITI (x=15,16) cannot select signals (TIMERx_TRGO \ TIMERx_CH0 \ TIMERx_CH1 \ TIMERx_MCH0) from its own module as trigger sources.
3. Except as described in points 1 and 2, all outputs can select all inputs as trigger sources.

When trigger input selection INSELx[7:0] bits is configured to be 0x00, TRIGSEL trigger input is selected as low level; when configured to be 0x01, TRIGSEL trigger input is selected as high level.When illegal data is selected for these outputs, the output will be selected as 0.

11.5. Register definition

TRIGSEL base address: 0x4001 4400

11.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0)

Address offset: 0x00
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_0 register. 0: TRIGSEL_EXTOUT_0 register write is enabled. 1: TRIGSEL_EXTOUT_0 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output1 is connected with TRIGSEL_OUT1. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is connected with TRIGSEL_OUT0. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1)

Address offset: 0x04
Reset value: 0x0000 0000

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

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

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

INSEL1[7:0]	INSEL0[7:0]
rw	rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_1 register. 0: TRIGSEL_EXTOUT_1 register write is enabled. 1: TRIGSEL_EXTOUT_1 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output1 is connected with TRIGSEL_OUT3. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is connected with TRIGSEL_OUT2. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.3. Trigger selection for EXTOUT register 2 (TRIGSEL_EXTOUT_2)

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_2 register. 0: TRIGSEL_EXTOUT_2 register write is enabled. 1: TRIGSEL_EXTOUT_2 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output1 is connected with TRIGSEL_OUT5. For the detailed configuration, please refer to

		Table 11-1. Trigger input bit fields selection.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is connected with TRIGSEL_OUT4. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.4. Trigger selection for EXTOUT register 3 (TRIGSEL_EXTOUT_3)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_EXTOUT_3 register. 0: TRIGSEL_EXTOUT_3 register write is enabled. 1: TRIGSEL_EXTOUT_3 register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for output1 These bits are used to select trigger input signal connected to output1. The output1 is connected with TRIGSEL_OUT7. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is connected with TRIGSEL_OUT6. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

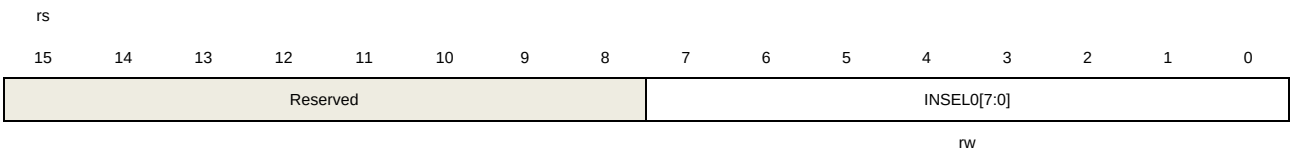
11.5.5. Trigger selection for TIMER0_ITI register (TRIGSEL_TIMER0ETI)

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
LK	Reserved														



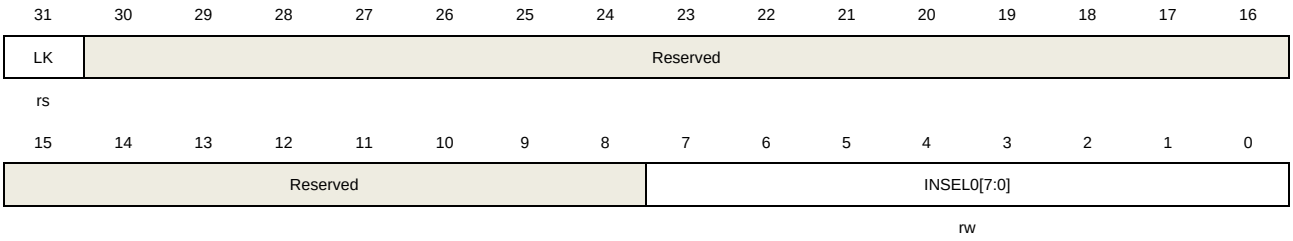
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0ETI register. 0: TRIGSEL_TIMER0ETI register write is enabled. 1: TRIGSEL_TIMER0ETI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER0_ETI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.6. Trigger selection for TIMER1_ETI register (TRIGSEL_TIMER1ITI)

Address offset: 0x14

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER1ITI register. 0: TRIGSEL_TIMER1ITI register write is enabled. 1: TRIGSEL_TIMER1ITI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER1_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.7. Trigger selection for TIMER2_ITI register (TRIGSEL_TIMER2ITI)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER2ITI register. 0: TRIGSEL_TIMER2ITI register write is enabled. 1: TRIGSEL_TIMER2ITI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER2_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.8. Trigger selection for TIMER3_ITI register (TRIGSEL_TIMER3ITI)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER3ITI register. 0: TRIGSEL_TIMER3ITI register write is enabled. 1: TRIGSEL_TIMER3ITI register write is disabled.

30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER3_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***

11.5.9. Trigger selection for TIMER4_ITI register (TRIGSEL_TIMER4ITI)

Address offset: 0x20

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER4ITI register. 0: TRIGSEL_TIMER4ITI register write is enabled. 1: TRIGSEL_TIMER4ITI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER4_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.10. Trigger selection for TIMER7_ITI register (TRIGSEL_TIMER7ITI)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7ITI register. 0: TRIGSEL_TIMER7ITI register write is enabled. 1: TRIGSEL_TIMER7ITI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER7_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.11. Trigger selection for TIMER15_ITI register (TRIGSEL_TIMER15ITI)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER15ITI register. 0: TRIGSEL_TIMER15ITI register write is enabled. 1: TRIGSEL_TIMER15ITI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER15_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.12. Trigger selection for TIMER16_ITI register (TRIGSEL_TIMER16ITI)

Address offset: 0x30

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER16ITI register. 0: TRIGSEL_TIMER16ITI register write is enabled. 1: TRIGSEL_TIMER16ITI register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER16_ITI trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.13. Trigger selection for DAC register (TRIGSEL_DAC)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_DAC register. 0: TRIGSEL_DAC register write is enabled. 1: TRIGSEL_DAC register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0

These bits are used to select trigger input signal connected to output0. The output0 is used as the source of DAC_OUT_EXTRIG (DAC_OUT external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.14. Trigger selection for ADC0_ROUTRG register (TRIGSEL_ADC0_ROUTRG)

Address offset: 0x38
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC0_ROUTRG register. 0: TRIGSEL_ADC0_ROUTRG register write is enabled. 1: TRIGSEL_ADC0_ROUTRG register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as the source of ADC0_ROUTRG (ADC0 routine sequence external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.15. Trigger selection for ADC0_INSTRG register (TRIGSEL_ADC0_INSTRG)

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
LK	Reserved														
rs															

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

rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC0_INSTRG register. 0: TRIGSEL_ADC0_INSTRG register write is enabled. 1: TRIGSEL_ADC0_INSTRG register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as the source of ADC0_INSTRG (ADC0 inserted sequence external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.16. Trigger selection for ADC1_ROUTRG register (TRIGSEL_ADC1_ROUTRG)

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
LK	Reserved														
rs															

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

rw

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC1_ROUTRG register. 0: TRIGSEL_ADC1_ROUTRG register write is enabled. 1: TRIGSEL_ADC1_ROUTRG register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as the source of ADC1_ROUTRG (ADC1 routine sequence external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.17. Trigger selection for ADC1_INSTRG register (TRIGSEL_ADC1_INSTRG)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC1_INSTRG register. 0: TRIGSEL_ADC1_INSTRG write is enabled. 1: TRIGSEL_ADC1_INSTRG register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as the source of ADC1_INSTRG (ADC1 inserted sequence external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.18. Trigger selection for ADC2_ROUTRG register (TRIGSEL_ADC2_ROUTRG)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it

		disables write access to TRIGSEL_ADC2_ROUTRG register. 0: TRIGSEL_ADC2_ROUTRG register write is enabled. 1: TRIGSEL_ADC2_ROUTRG register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as the source of ADC2_ROUTRG (ADC2 routine sequence external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***

11.5.19. Trigger selection for ADC2_INSTRG register (TRIGSEL_ADC2_INSTRG)

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
LK	Reserved														
rs															

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC2_INSTRG register. 0: TRIGSEL_ADC2_INSTRG register write is enabled. 1: TRIGSEL_ADC2_INSTRG register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is used as the source of ADC2_INSTRG (ADC2 inserted sequence external trigger input). For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.20. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN)

Address offset: 0x50
Reset value: 0x0000 0042

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

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

LK	Reserved
rs	

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0BRKIN register. 0: TRIGSEL_TIMER0BRKIN register write is enabled. 1: TRIGSEL_TIMER0BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER0_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER0_BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***

11.5.21. Trigger selection for TIMER0_CHBRKIN register (TRIGSEL_TIMER0CHBRKIN)

Address offset: 0x54

Reset value: 0x0045 4443

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved							INSEL2[7:0]							
rs								rw							

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

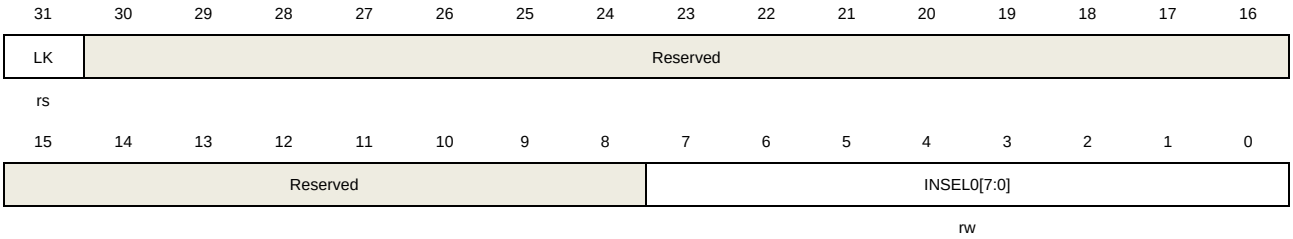
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER0CHBRKIN register. 0: TRIGSEL_TIMER0CHBRKIN register write is enabled. 1: TRIGSEL_TIMER0CHBRKIN register write is disabled.
30:24	Reserved	Must be kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for TIMER0_CH2BRKIN. These bits are used to select trigger input signal connected to output2. The output2

		is used as the source of TIMER0_CH2BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***
15:8	INSEL1[7:0]	Trigger input source selection for TIMER0_CH1BRKIN These bits are used to select trigger input signal connected to output1. The output1 is used as the source of TIMER0_CH1BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER0_CH0BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER0_CH0BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***

11.5.22. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN)

Address offset: 0x58
Reset value: 0x0000 0046

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



Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7BRKIN register. 0: TRIGSEL_TIMER7BRKIN register write is enabled. 1: TRIGSEL_TIMER7BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER7_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER7_BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.23. Trigger selection for TIMER7_CHBRKIN register (TRIGSEL_TIMER7CHBRKIN)

Address offset: 0x5C
Reset value: 0x0049 4847

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	Reserved							INSEL2[7:0]							
rs								rw							

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER7CHBRKIN register. 0: TRIGSEL_TIMER7CHBRKIN register write is enabled. 1: TRIGSEL_TIMER7CHBRKIN register write is disabled.
30:24	Reserved	Must be kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for TIMER7_CH2BRKIN. These bits are used to select trigger input signal connected to output2. The output2 is used as the source of TIMER7_CH2BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.
15:8	INSEL1[7:0]	Trigger input source selection for TIMER7_CH1BRKIN These bits are used to select trigger input signal connected to output1. The output1 is used as the source of TIMER7_CH1BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER7_CH0BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER7_CH0BRKIN trigger input For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.24. Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN)

Address offset: 0x60

Reset value: 0x0000 004A

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER15BRKIN register. 0: TRIGSEL_TIMER15BRKIN register write is enabled. 1: TRIGSEL_TIMER15BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER15_BRKIN. These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER15_BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

11.5.25. Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN)

Address offset: 0x64

Reset value: 0x0000 004B

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

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

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

Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER16BRKIN register. 0: TRIGSEL_TIMER16BRKIN register write is enabled. 1: TRIGSEL_TIMER16BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER16_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER16_BRKIN trigger input. For the detailed configuration, please refer to ***Table 11-1. Trigger input bit fields selection***.

12. TIMER (TIMERx)

Table 12-1. Timers (TIMERx) are divided into four sorts

TIMER		TIMER0/7	TIMER1/2/3/4	TIMER15/16	TIMER5/6
TYPE		Advanced	General-L0	General-L3	Basic
Prescaler		16-bit	16-bit	16-bit	16-bit
Counter		16-bit	16-bit (TIMER2/3/4) 32-bit (TIMER1)	16-bit	16-bit
Count mode		UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP ONLY
Repetition		●	×	●	×
Channel Capture/Compare		4	4	3	0
Composite PWM mode		●	x	●	×
Asymmetric PWM mode		●	×	×	×
Output match pulse select		●	x	●	×
Complementary & Dead-time		●	×	●	×
Break function	BREAK	●	×	●	×
	Channel BREAK	●	×	×	×
Single Pulse		●	●	●	●
Quadrature decoder		●	●	×	×
Non-quadrature decoder		×	●	×	×
Master-slave management		●	●	●	×
Inter Connection		● (1)	● (2)	● (3)	TRGO TO DAC
Synchronization and initial direction and value refresh		●	×	●	×
DMA		●	●	●	● (4)
Debug Mode		●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI14
(1)	TIMER0	TIMER4_TRGO	TIMER1_TRGO	TIMER2_TRGO	TRIGSEL	-
	TIMER7	TIMER0_TRGO	TIMER1_TRGO	TIMER3_TRGO		
(2)	TIMER1	TIMER0_TRGO	TIMER7_TRGO	TIMER2_TRGO		
	TIMER2	TIMER0_TRGO	TIMER1_TRGO	TIMER4_TRGO		
	TIMER3	TIMER0_TRGO	TIMER1_TRGO	TIMER2_TRGO		
	TIMER4	TIMER1_TRGO	TIMER2_TRGO	TIMER3_TRGO		
(3)	TIMER15	TIMER0_TRGO	TIMER2_TRGO	TIMER3_TRGO	TIMER16_CH0	TRIGSEL
	TIMER16	TIMER0_TRGO	TIMER2_TRGO	TIMER15_CH0	TIMER7_TRGO	
(4)	Only update events will generate a DMA request. TIMER5/6 do not have DMAS bit (DMA request source selection).					

12.1. Advanced timer (TIMERx, x=0, 7)

12.1.1. Overview

The advanced timer module (TIMER0/7) is an 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 is suitable for motor control applications.

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

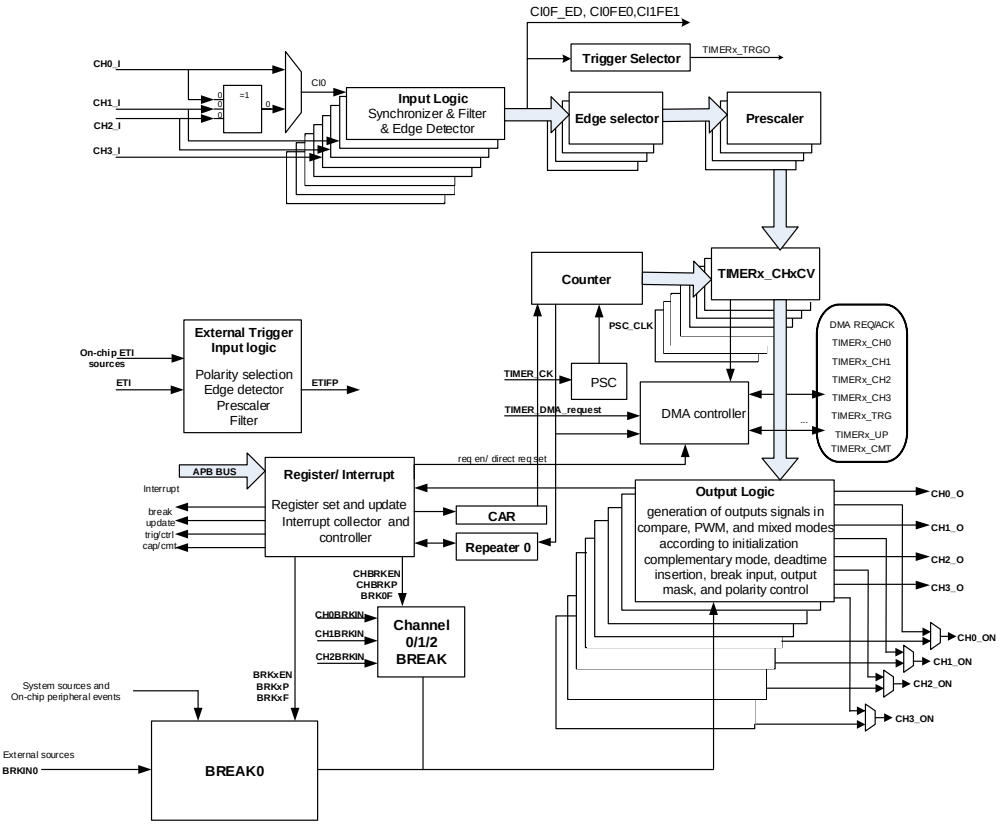
12.1.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is independent and user-configurable: input capture mode, output compare mode, programmable PWM mode, single pulse mode and trigger out.
- Programmable dead time insertion and separated dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input function: BREAK and channel 0/1/2 BREAK.
- Interrupt output or DMA request: update event, trigger event, compare/capture event and break input.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

12.1.3. Block diagram

Figure 12-1. Advanced timer block diagram provides details of the internal configuration of the advanced timer.

Figure 12-1. Advanced timer block diagram



12.1.4. Function overview

Clock selection

The clock source of the advanced timer can be either the CK_TIMER or an alternate clock source controlled by TSCFGy[4:0] (y=0..6) in SYSCFG_TIMERxCFG(x=0,7) registers.

- TSCFGy[4:0] (y=0..6) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

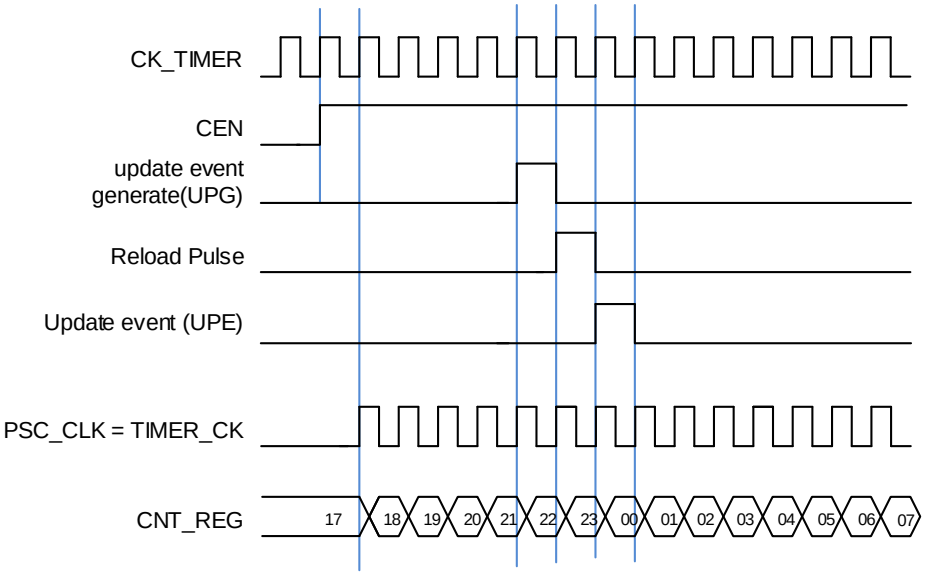
The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=0..6) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If TSCFGy[4:0] (y=0..2,6) in SYSCFG_TIMERxCFG(x=0,7) registers are setting to a nonzero value, the prescaler is clocked by other clock sources selected in the TSCFGy[4:0] (y=0..2,6) bit-filed, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5) are setting to

a nonzero value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 12-2. Normal mode, internal clock divided by 1



- TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CHn (n=0..3). This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7 and 0x9/0xA.

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 TSCFG6[4:0] to 0x1~0x4.

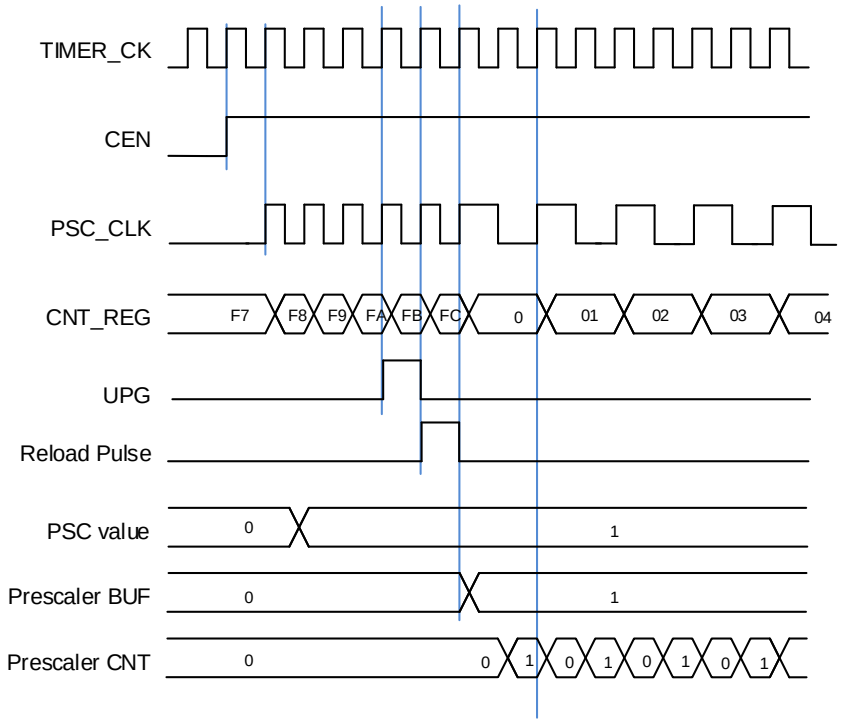
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is setting the TSCFG6[4:0] to 0x8. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Clock prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed ongoing, but it is adopted at the next update event.

Figure 12-3. Counter timing diagram with prescaler division change from 1 to 2



Up counting mode

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 restarts from 0. If the repetition counter is set, the update event will be generated after (TIMERx_CREP0+1) times of overflow. Otherwise the update event is generated each time when counter overflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 0 for the up-counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

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

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

Figure 12-4. Timing diagram of up counting mode, PSC=0/2 and ***Figure 12-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing*** show some examples of

the counter behavior for different clock prescaler factors when TIMERx_CAR=0x99.

Figure 12-4. Timing diagram of up counting mode, PSC=0/2

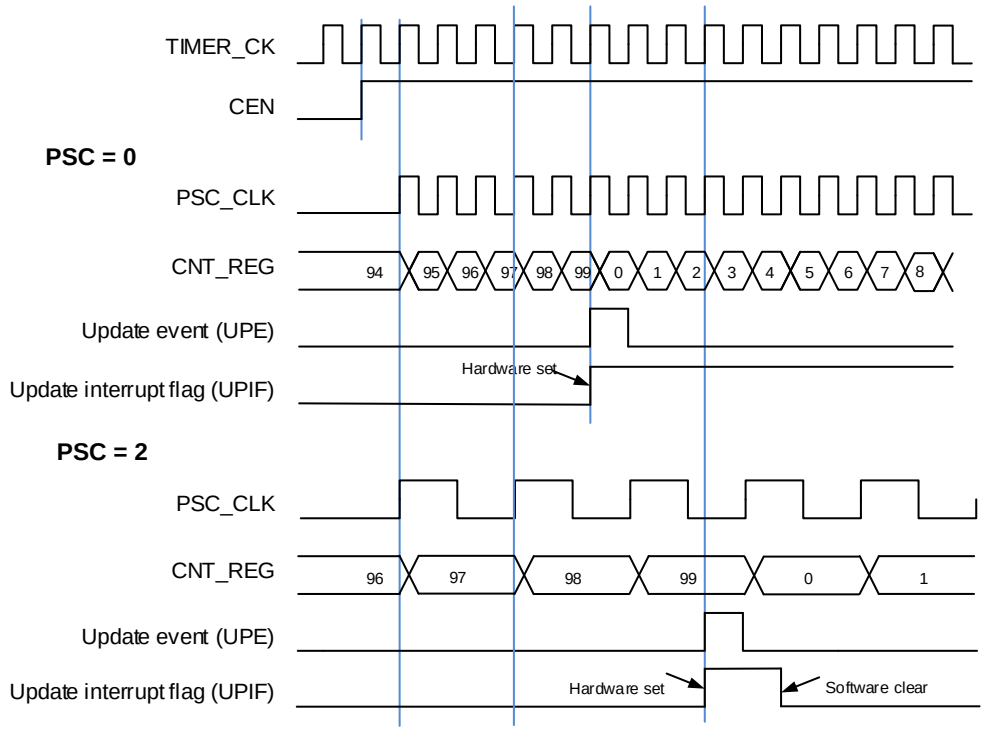
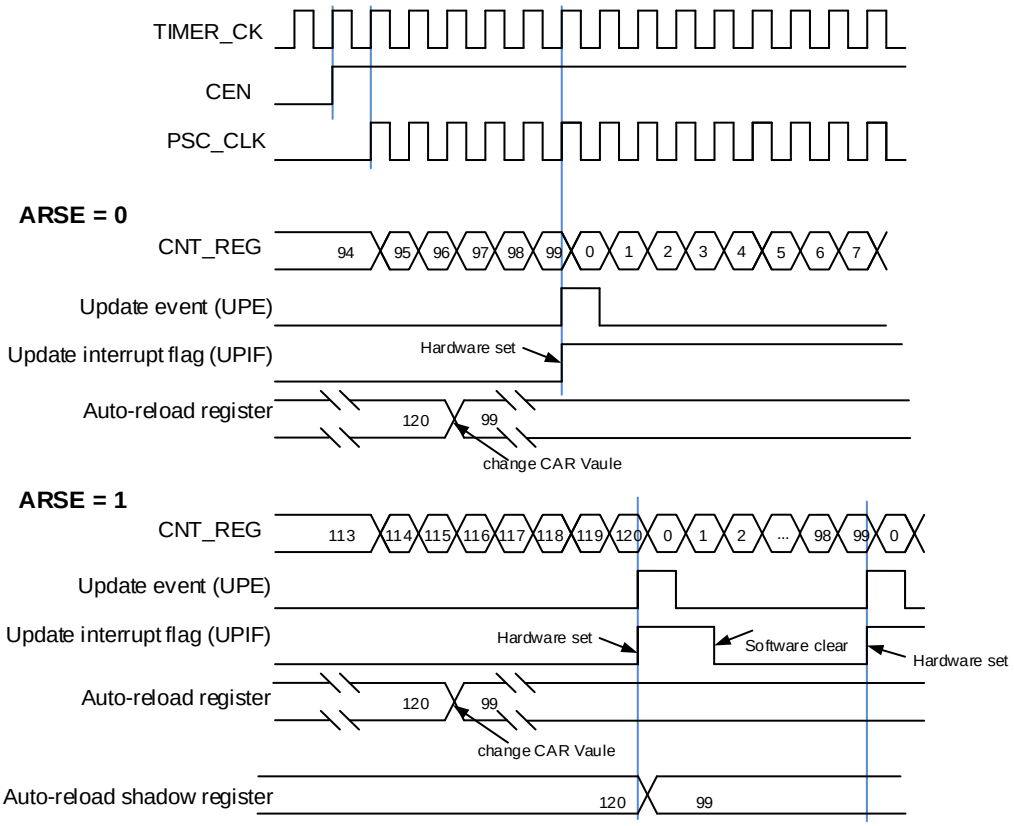


Figure 12-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the TIMERx_CAR register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. If the repetition counter is set, the update event will be generated after (TIMERx_CREP0+1) times of underflow. Otherwise, the update event is generated each time when counter underflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

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

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

Figure 12-6. Timing diagram of down counting mode, PSC=0/2 and *Figure 12-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior in different clock frequencies when TIMERx_CAR = 0x99.

Figure 12-6. Timing diagram of down counting mode, PSC=0/2

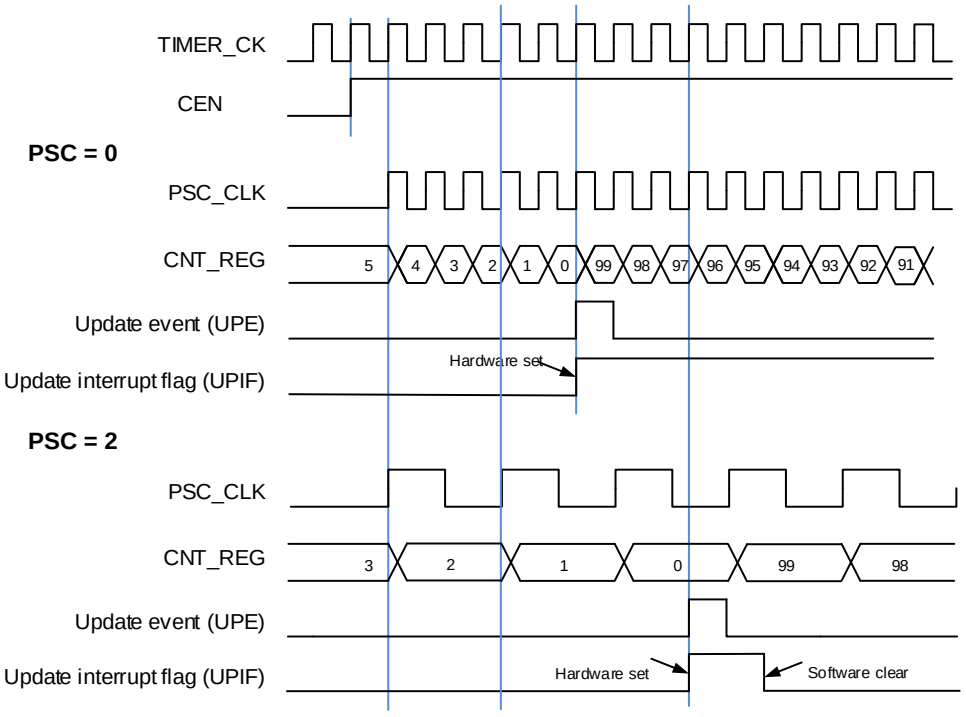
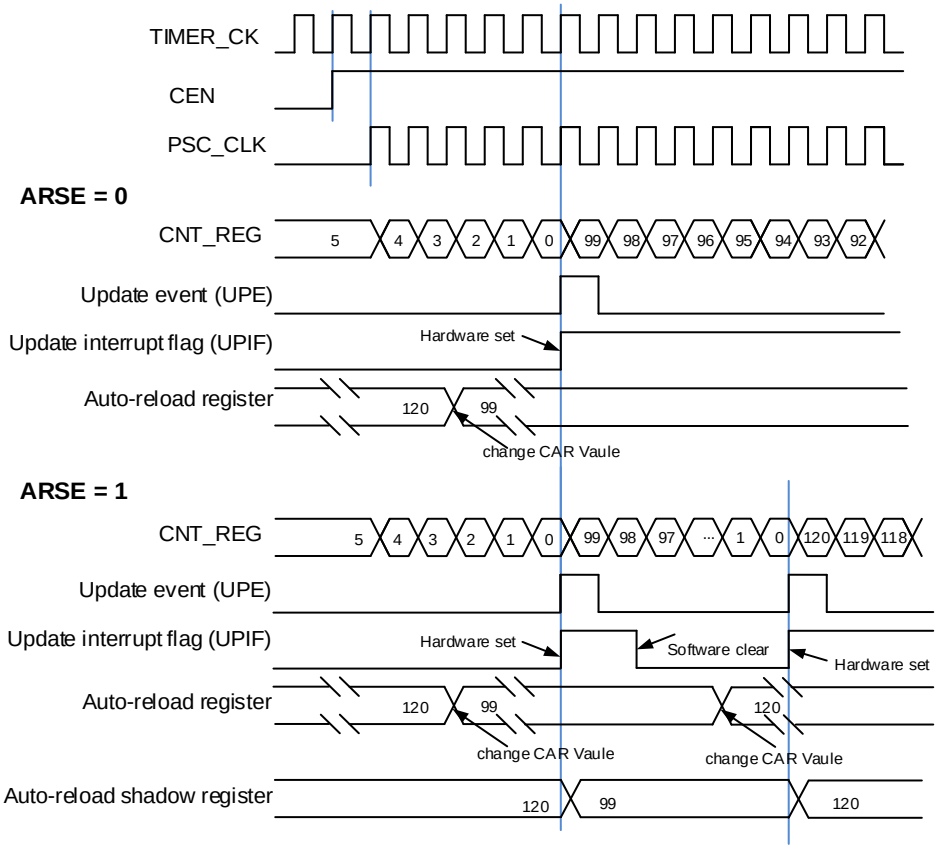


Figure 12-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing



Center-aligned counting mode

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 (TIMERx_CAR-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned counting mode. The counting direction is updated by hardware automatically.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

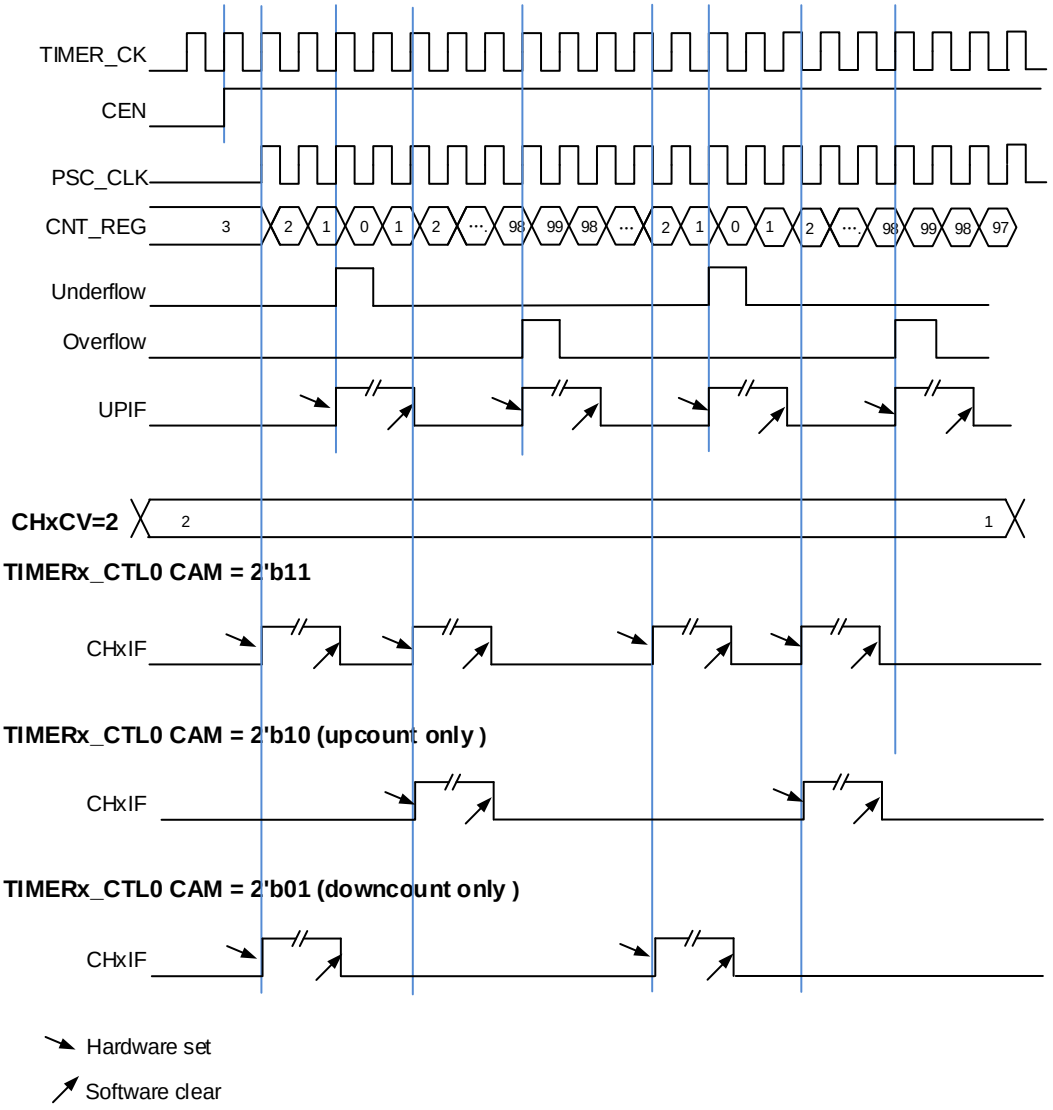
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM[1:0] in TIMERx_CTL0. The details refer to ***Figure 12-8. Timing diagram of center-aligned counting mode***.

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

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

Figure 12-8. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 12-8. Timing diagram of center-aligned counting mode



Counter repetition

Repetition counter is used to generate the update event or update the timer registers only after a given number (N+1) cycles of the counter, where N is the value of CREP0 bit in TIMERx_CREP0 register. The repetition counter is decremented at each counter overflow in up counting mode, at each counter underflow in down counting mode or at each counter overflow and at each counter underflow in center-aligned counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP0 in TIMERx_CREP0 register and generate an update event.

The new written CREP0 value will not take effect until the next update event. When the value of CREP0 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 CREP0 value takes effect. If an update event is generated by software after writing an odd number to CREP0, the update events will be generated on the underflow. If the next update event occurs on overflow after writing an odd number to CREP0, then the subsequent update events will be generated on the overflow.

Figure 12-9. Repetition counter timing diagram of center-aligned counting mode

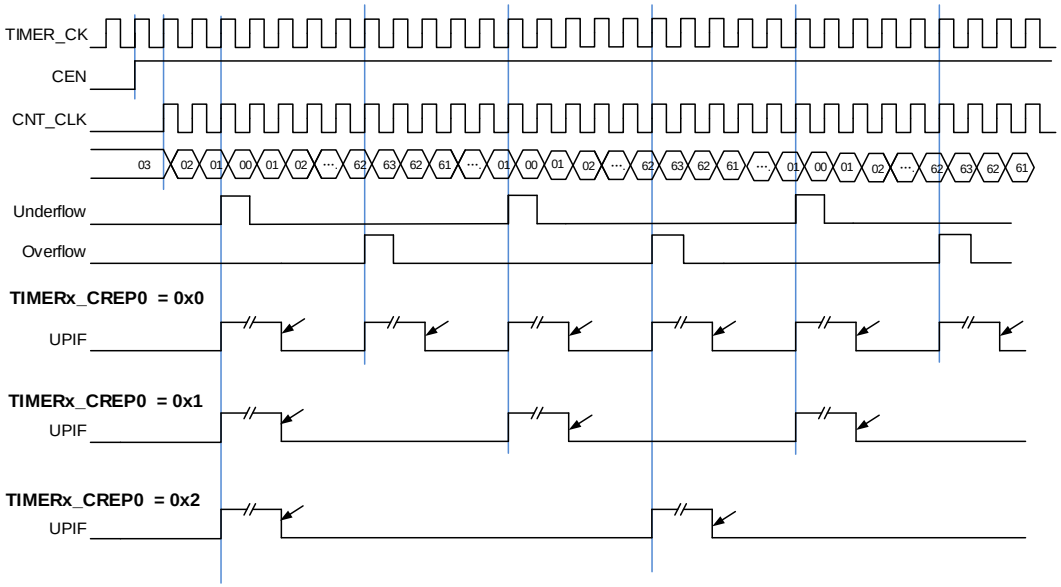


Figure 12-10. Repetition counter timing diagram of up counting mode

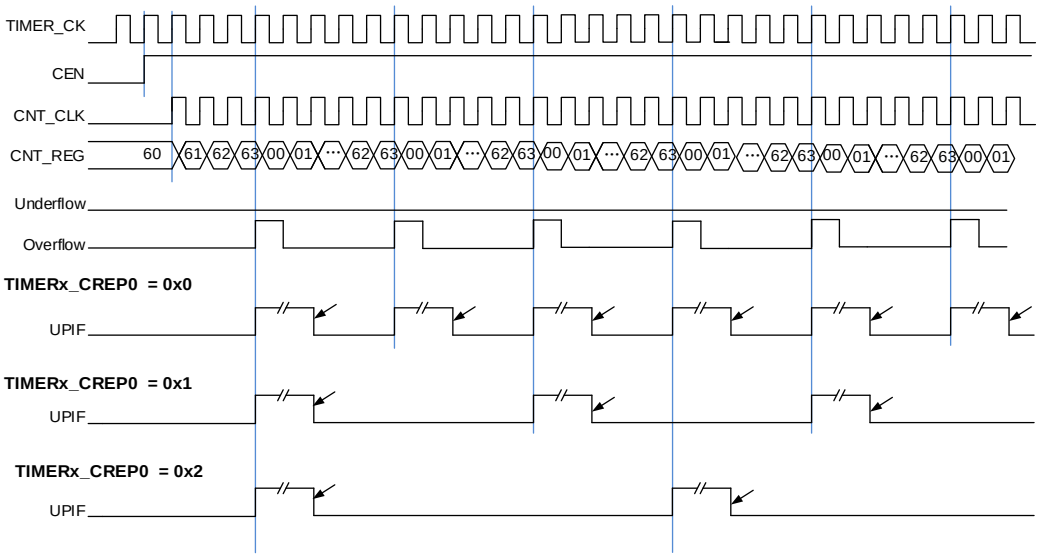
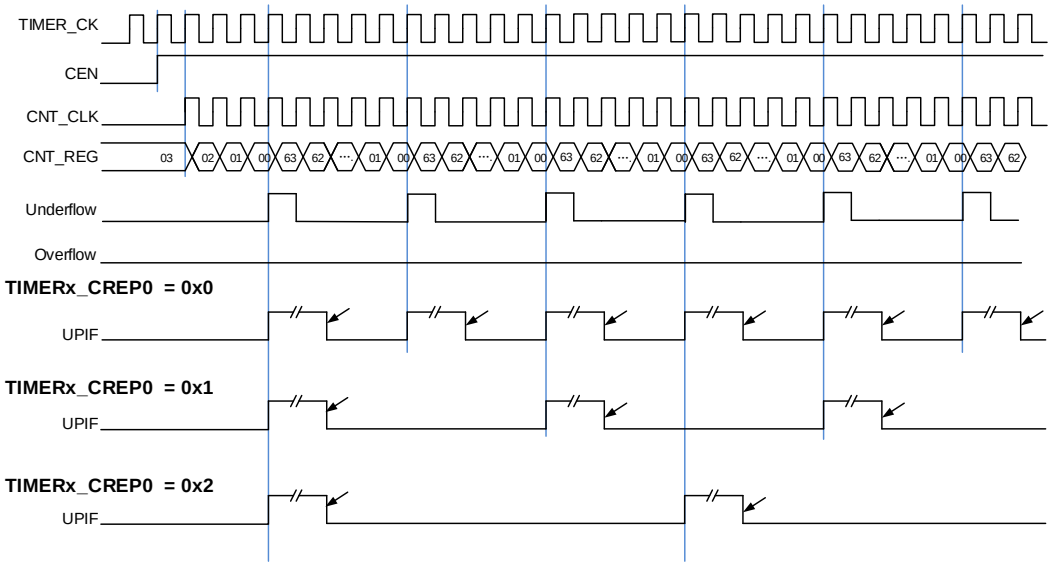


Figure 12-11. Repetition counter timing diagram of down counting mode



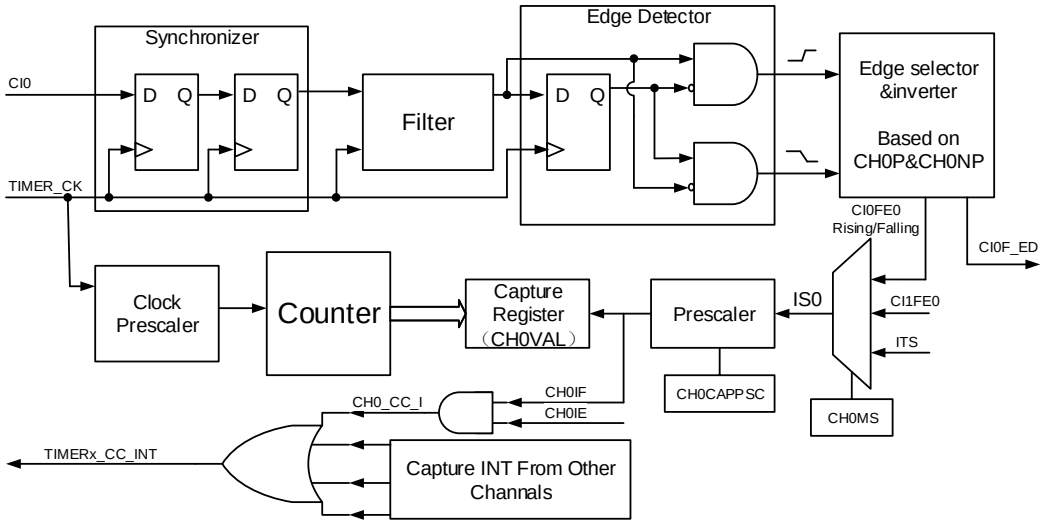
Capture/compare channels

The advanced timer has four independent channels which can be used as capture inputs or compare outputs. Each channel is built around a channel capture compare register including an input stage, a channel controller and an output stage.

■ Input capture mode

Input capture mode 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 (x=0...3) registers, at the same time the CHxIF (x=0..3) bits are set and the channel interrupt is generated if it is enabled when CHxIE =1 (x=0..3).

Figure 12-12. Input capture logic for channel 0



The input signals of channelx (CIx) can be the TIMERx_CHx signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals (just for CI0).

First, the input signal of channel (CIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP/ CHxNP bits. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT bit in TIMERx_CHCTL0 register).

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

Step2: Edge selection (CHxP and CHxNP bits in TIMERx_CHCTL2 register).

Rising edge or falling edge, choose one by configuring CHxP and CHxNP bits.

Step3: Capture source selection (CHxMS bit in TIMERx_CHCTL0 register).

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x000) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN bits in TIMERx_DMAINTEN).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN bit in TIMERx_CHCTL2).

Result: When the wanted input signal is captured, TIMERx_CHxCV will be set by counter's value and CHxIF bit is asserted. If the CHxIF bit is 1, the CHxOF bit will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN bits bits in TIMERx_DMAINTEN.

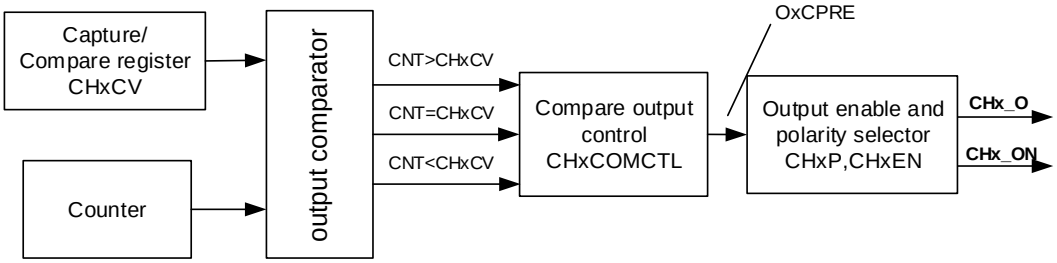
Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 3'b001 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 3'b010 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

- **Output compare mode**

Figure 12-13. Output compare logic showS the logic circuit of output compare mode.

Figure 12-13. Output compare logic



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 TIMERx_CHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV register, the CHxIF bit will be set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be asserted, if CHxDEN=1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP/CHxNP.
- Enable the output by CHxEN/ CHxNEN.

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

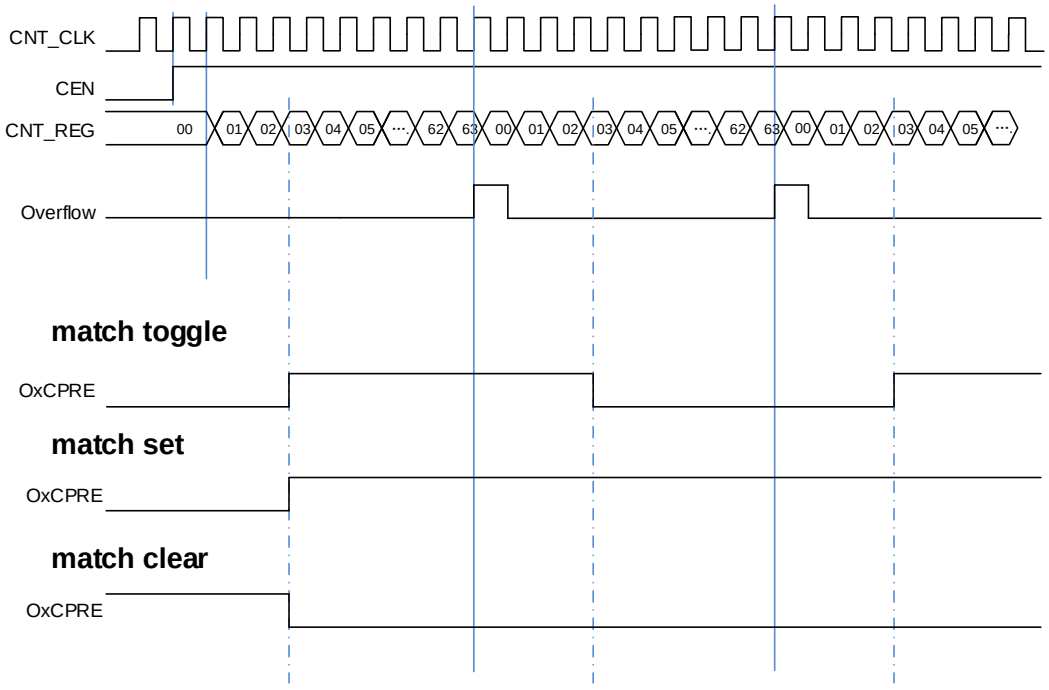
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.

The TIMERx_CHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 12-14. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 12-14. Output-compare in three modes



PWM mode

In the PWM output mode (by setting the CHxCOMCTL bit to 4'b0110 (PWM mode 0) or to 4'b0111 (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 12-15. Timing diagram of EAPWM*** shows the EAPWM output and interrupts waveform.

The CAPWM's period is determined by 2*TIMERx_CAR, and the duty cycle is determined by 2*TIMERx_CHxCV. ***Figure 12-16. Timing diagram 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 counter, the output will be always active in PWM mode 0 (CHxCOMCTL=4'b0110). And if the value of TIMERx_CHxCV is greater than the value of counter, the output will be always inactive in PWM mode 1 (CHxCOMCTL=4'b0111).

Figure 12-15. Timing diagram of EAPWM

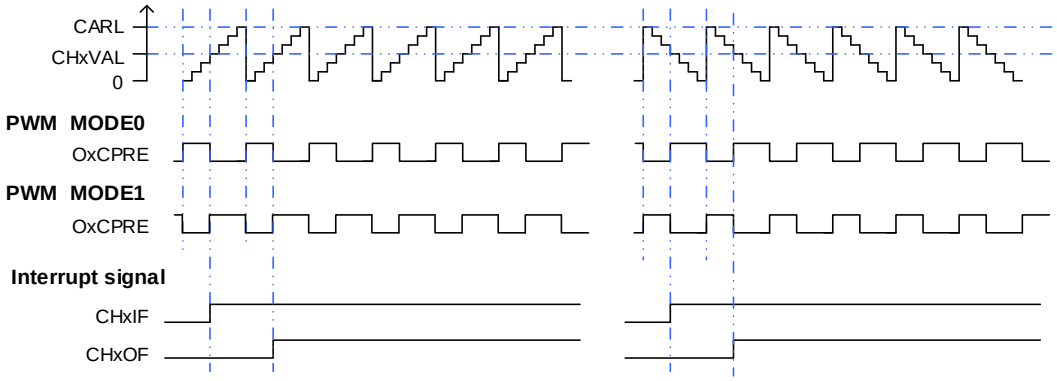
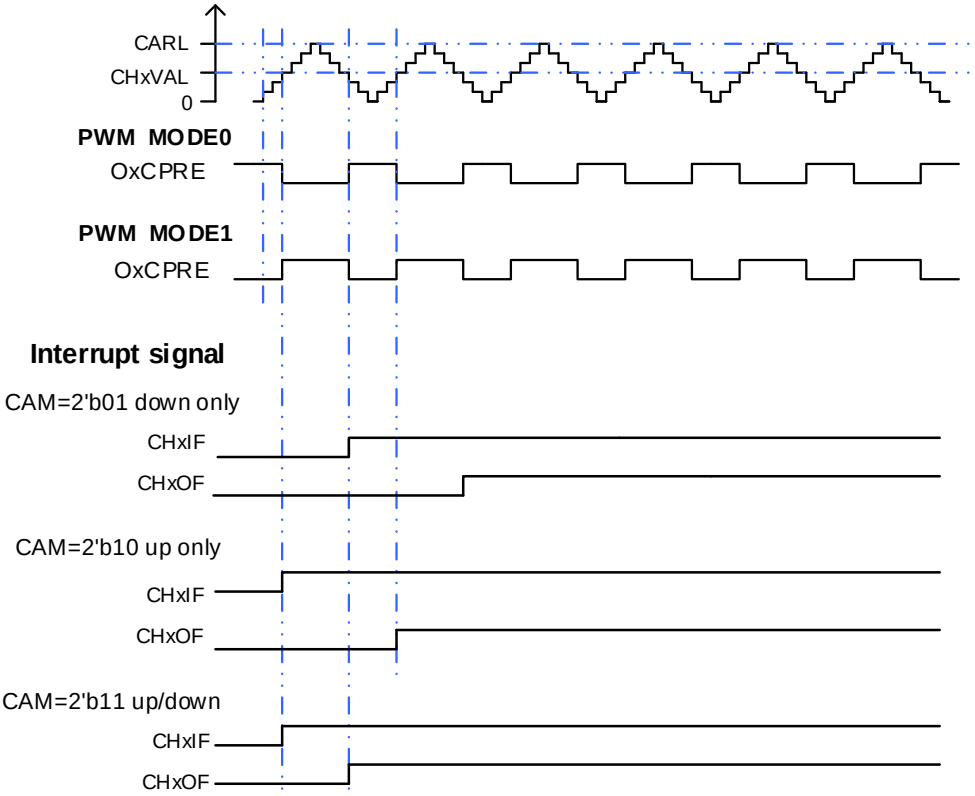


Figure 12-16. Timing diagram of CAPWM



Composite PWM mode

In the composite PWM mode (CHxCPWMEN = 1'b1, CHxMS[2:0] = 3'b000 and CHxCOMCTL = 4'b0110 or 4'b0111), the PWM signal output in channel x (x=0..3) is composited by CHxVAL and CHxCOMVAL_ADD bits.

If CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b0 (up counting mode), or CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b1 (Down counting mode), the channel x output is forced low when the counter matches the value of CHxVAL. It is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b0 (up counting mode), or

CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b1 (down counting mode) the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when the counter matches the value of CHxCOMVAL_ADD.

When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected composite PWM mode).

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the ***Table 12-2.The composite PWM pulse width***.

Table 12-2.The composite PWM pulse width

Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL – CHxCOMVAL_ADD)
	PWM mode 1	(CHxCOMVAL_ADD – CHxVAL)
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD – CHxVAL)
(CHxVAL = CHxCOMVAL_ADD ≤ CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down counting)	100%
	PWM mode 0 (down counting) or PWM mode 1 (up counting)	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

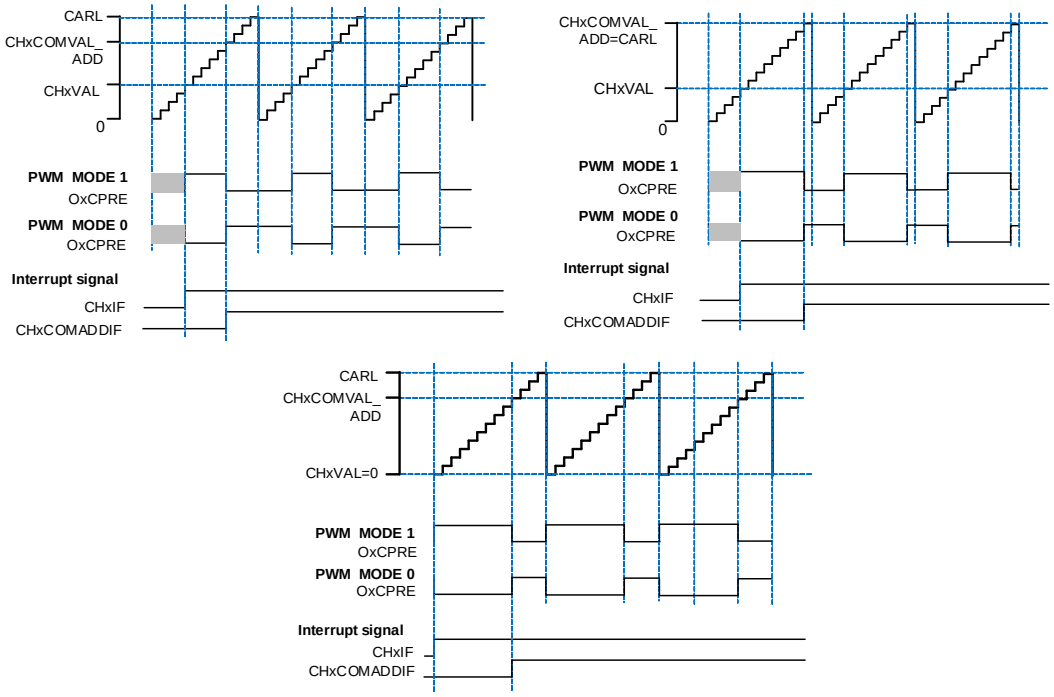
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when CHxCPWMEN=1) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be

divided into four situations:

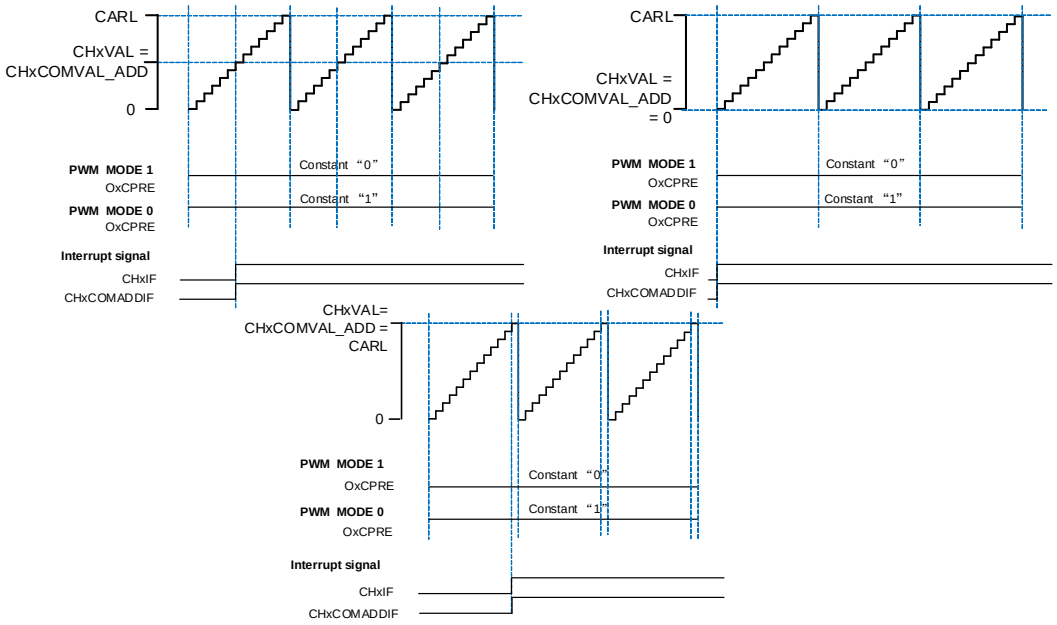
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-17. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



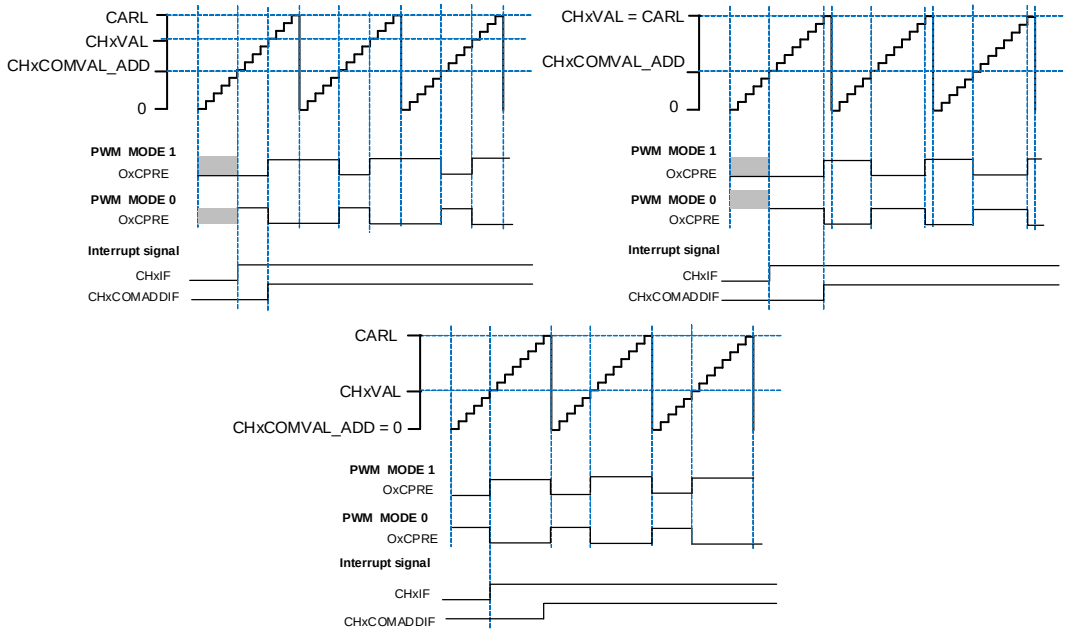
2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-18. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



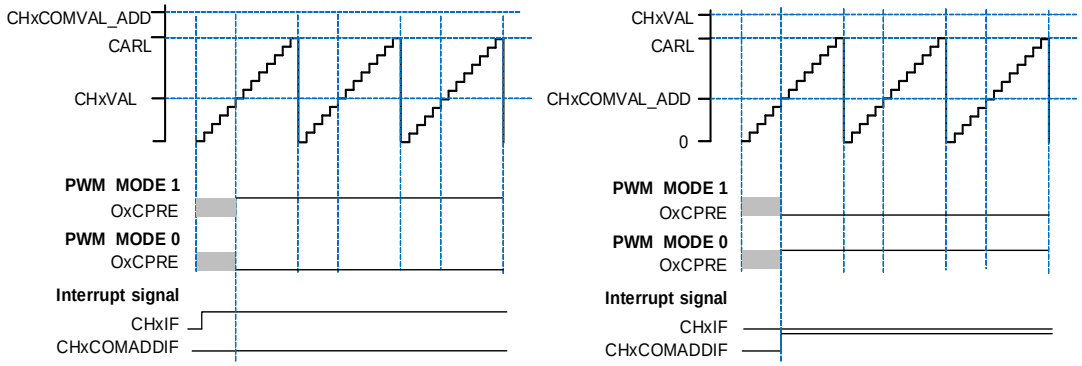
3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-19. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

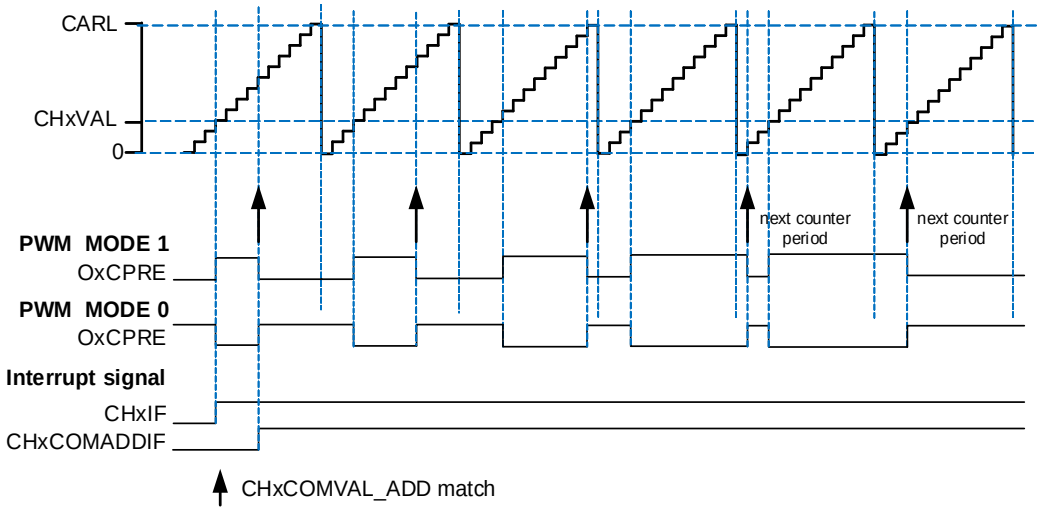
Figure 12-20. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



The composite PWM mode is intended to support the generation of PWM signals where the period is not modified while the signal is being generated, but the duty cycle will be varied. ***Figure 12-21. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD*** shows the PWM output and interrupts waveform.

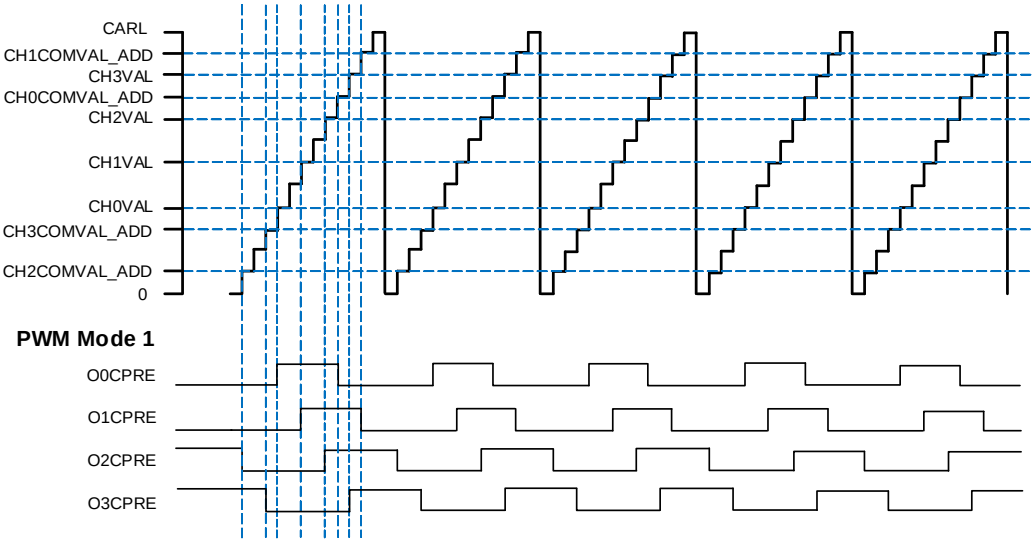
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 12-21. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD



If more than one channels are configured in composite PWM mode, it is possible to fix an offset for the channel x match edge of each pair with respect to other channels. This behavior is useful in the generation of lighting PWM control signals where it is desirable that edges are not coincident with each other pair to help eliminate noise generation. The CHxVAL register value is the shift of the PWM pulse with respect to the beginning of counter period.

Figure 12-22. Four Channels outputs in Composite PWM mode



Asymmetric PWM mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. In the asymmetric PWM mode (CHxCOMCTL[3:0]=4'b1010 / 4'b1011 and CHxMS[2:0] = 3'b000)，the PWM signal output in channel x (x=0..3) is composited by CHxVAL and CHxCOMVAL_ADD bits. The asymmetric PWM mode is only available in the center-aligned counting mode.

If CHxCOMCTL = 4'b1010 (asymmetric PWM mode 0) and DIR = 1'b0 (up counting mode), the channel x output is forced low when the counter matches the value of CHxVAL.

If CHxCOMCTL = 4'b1010 (asymmetric PWM mode 0) and DIR = 1'b1 (down counting mode), the channel x output is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b1011 (asymmetric PWM mode 1) and DIR = 1'b0 (up counting mode), the channel x output is forced high when the counter matches the value of CHxVAL.

If CHxCOMCTL = 4'b1011 (asymmetric PWM mode 1) and DIR = 1'b1 (down counting mode), the channel x output is forced low when the counter matches the value of CHxCOMVAL_ADD.

Compared to composite PWM mode, asymmetric PWM mode is only valid with center-aligned mode, and the CHxVAL is only valid when counting up and CHxCOMVAL_ADD is only valid when counting down. When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected asymmetric PWM mode).

When CHxVAL =0, the channel x output level is low in Asymmetric PWM mode 0. When CHxVAL =0, the channel x output level is high in Asymmetric PWM mode 1.

When CHxCOMVAL_ADD =0, the channel x output level is high in Asymmetric PWM mode 0. When CHxCOMVAL_ADD =0, the channel x output level is low in Asymmetric PWM mode 1.

When CHxVAL = CARL, the channel x output level is low in asymmetric PWM mode 0. When CHxVAL = CARL, the channel x output level is high in Asymmetric PWM mode 1.

When CHxCOMVAL_ADD = CARL, the channel x output level is high in asymmetric PWM mode 0. When CHxCOMVAL_ADD = CARL, the channel x output level is low in Asymmetric PWM mode 1.

The PWM period is determined by 2*CARL and the PWM pulse width is determined by the following table ***Table 12-3.The asymmetric PWM pulse width***.

Table 12-3.The asymmetric PWM pulse width

Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD ≤ CARL or CHxCOMVAL_ADD < CHxVAL ≤ CARL or CHxVAL = CHxCOMVAL_ADD	Asymmetric PWM mode 0	CHxCOMVAL_ADD + CHxVAL
	Asymmetric PWM mode 1	2*CARL – CHxCOMVAL_ADD – CHxVAL
CHxVAL > CARL > CHxCOMVAL_ADD	Asymmetric PWM mode 0	100%
	Asymmetric PWM mode 1	0%
CHxCOMVAL_ADD > CARL > CHxVAL	Asymmetric PWM mode 0	0%
	Asymmetric PWM mode 1	100%
(CHxVAL>CARL)and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

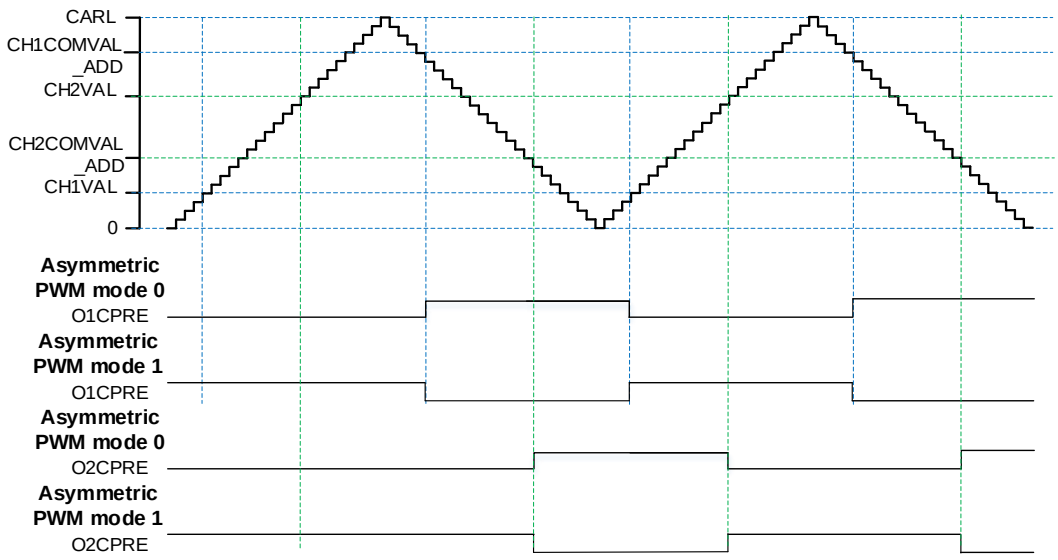
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in Composite PWM mode and Asymmetric PWM mode, when CHxCOMCTL[3:2]= 2'b10) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into three situations:

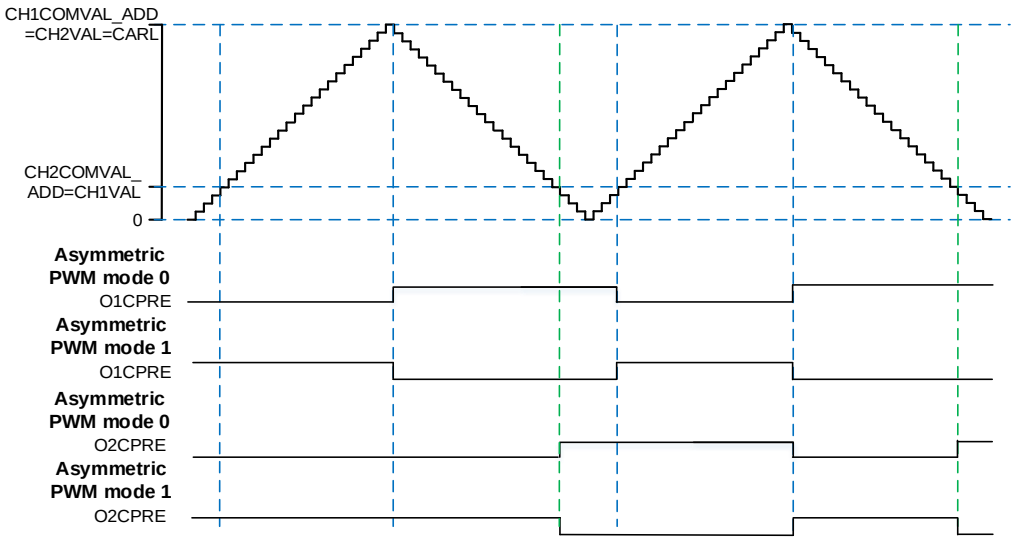
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-23. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL)

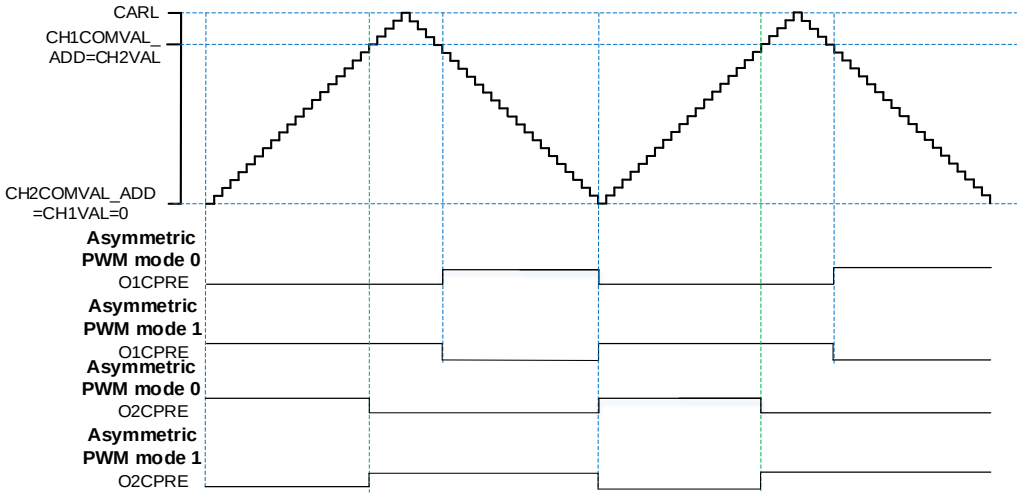
a. 0 < CH1VAL < CH1COMVAL_ADD < CARL / 0 < CH2COMVAL_ADD < CH2VAL < CARL



b. 0 < CH1VAL < CH1COMVAL_ADD = CARL / 0 < CH2COMVAL_ADD < CH2VAL = CARL, and CH1VAL = CH2COMVAL_ADD



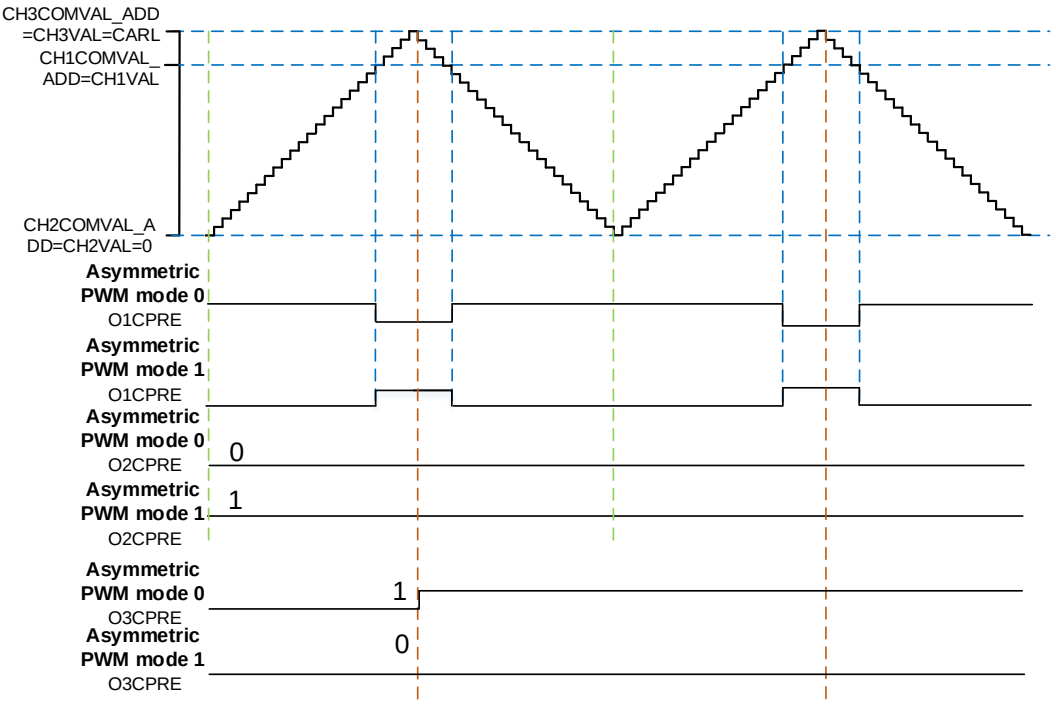
c. 0 = CH1VAL < CH1COMVAL_ADD < CARL / 0 = CH2COMVAL_ADD < CH2VAL < CARL, and CH1COMVAL_ADD = CH2VAL



2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-24. Channel x output PWM with（CHxVAL = CHxCOMVAL_ADD）

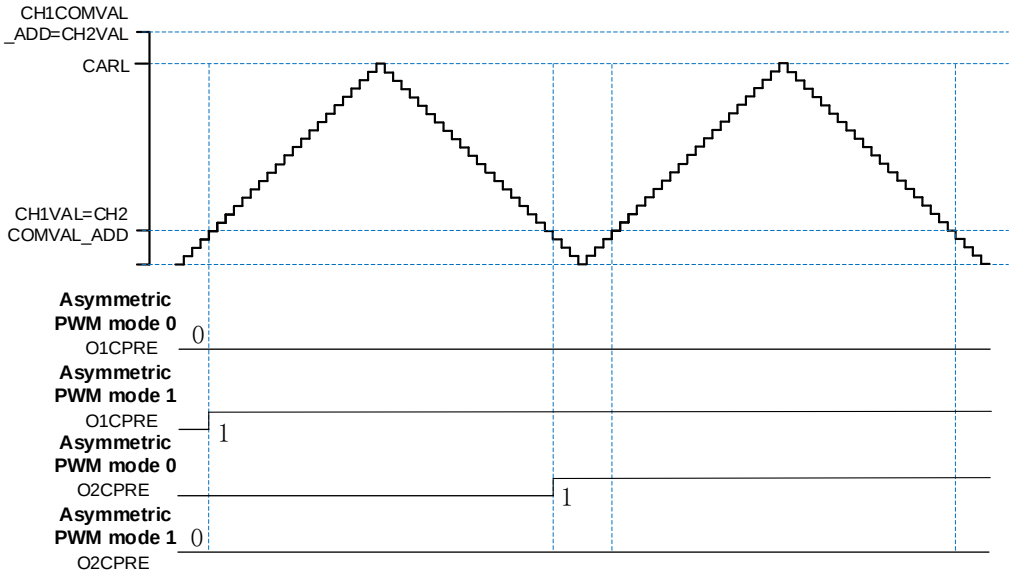
0 < CH1VAL = CH1COMVAL_ADD < CARL / CH2VAL = CH2COMVAL_ADD = 0 / CH3VAL = CH3COMVAL_ADD = CARL



3) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

Figure 12-25. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL

0 < CH1VAL < CARL < CH1COMVAL_ADD / 0 < CH2COMVAL_ADD < CARL < CH2VAL



Output match pulse select

Basing on that CHx_O (x=0..3) outputs are configured by CHxCOMCTL[3:0] (x=0..3) bits when the match events occur, the output signal is configured by CHxOMPSEL[1:0] (x=0..3) bit to be normal or a pulse.

When the match events occur, the CHxOMPSEL[1:0] (x=0..3) bits are used to select the output of OxCPRE which drives CHx_O:

- CHxOMPSEL = 2'b00, the OxCPRE signal is output normally with the configuration of CHxCOMCTL[3:0] bits;
- CHxOMPSEL = 2'b01, only the counter is counting up, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b10, only the counter is counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b11, both the counter is counting up and counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.

Figure 12-26. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00)

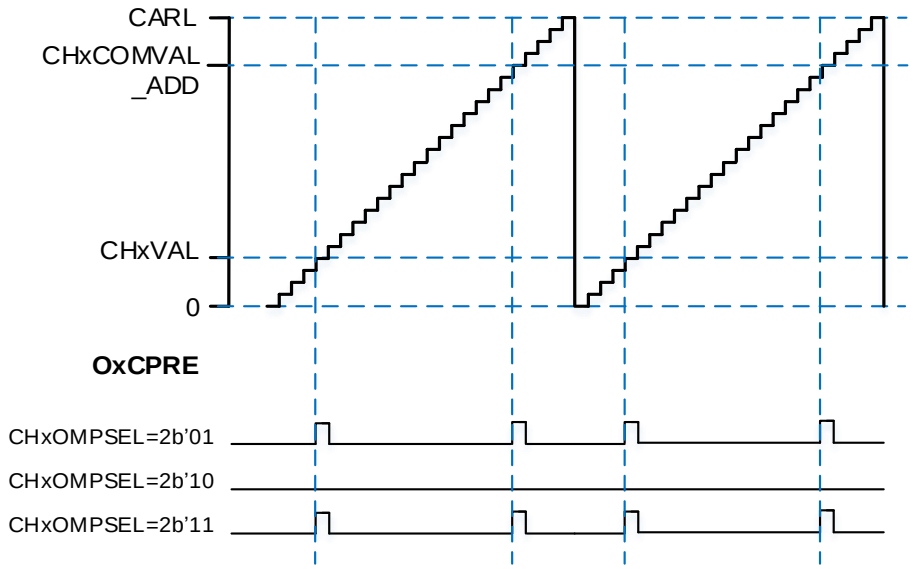
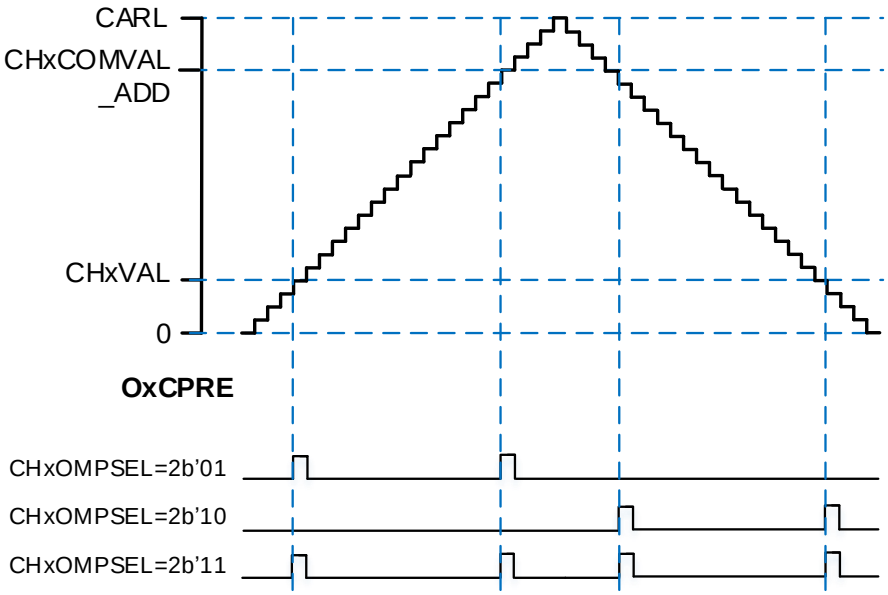


Figure 12-27. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00)



Channel output prepare signal

As is shown in ***Figure 12-13. Output compare logic***, when TIMERx is configured in compare match output mode, a middle signal named OxCPRE (channel x output prepare signal) will be generated before the channel outputs signal.

The OxCPRE signal has everal types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit.

Take OxCPRE as an example for description below, these include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the

TIMERx_CHxCV register.

The PWM mode 0/ PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

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

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next update event occurs (OCPRECIVM = 1) or underflow/overflow event occurs (OCPRECIVM = 0).

Outputs Complementary

Function of complementary is for a pair of channels, CHx_O and CHx_ON, the two output signals cannot be active at the same time. The TIMERx has 4 pairs of channels, all the four pairs have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register, the POEN&CHPOENx(x=0...2), ROS and IOS bits in the TIMERx_CCHP0 register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

When the the outputs of CHx_O and CHx_ON are complementary, there are three situations: output enable、output off-state and output disabled. The details are shown in ***Table 12-4. Complementary outputs controlled by parameters***.

Table 12-4. Complementary outputs controlled by parameters

Complementary Parameters					Output Status	
POEN &CHPOENx	**ROS**	**IOS**	**CHxEN**	**CHxNEN**	**CHx_O**	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable(1).	
				1	CHx_O/ CHx_ON output “off-state” (2): the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. (3)	
			1	0		
				1		
		1	x	x	CHx_O/ CHx_ON output “off-state”: the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.	
1	0	0/1	0	0	CHx_O/ CHx_ON = LOW CHx_O/ CHx_ON output disable.	
				1	CHx_O = LOW CHx_O output disable.	CHx_ON =OxCPRE⊕(4) CHxNP CHx_ON output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	CHx_ON = LOW CHx_ON output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	CHx_ON = (! OxCPRE)(5)⊕ CHxNP CHx_ON output enable.
	1		0	0	CHx_O = CHxP CHx_O output “off-state”.	CHx_ON = CHxNP CHx_ON output “off-state”.
				1	CHx_O = CHxP CHx_O output “off-state”	CHx_ON =OxCPRE⊕CHxNP CHx_ON output enable
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable	CHx_ON = CHxNP CHx_ON output “off-state”.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable	CHx_ON = (! OxCPRE)⊕ CHxNP CHx_ON output enable.

Note:

(1) output disable: the CHx_O / CHx_ON are disconnected to corresponding pins, the pin is floating

with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) “off-state”: CHx_O / CHx_ON output with inactive state (e.g., CHx_O = (0⊕CHxP) = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (! OxCPRE)：the complementary output of the OxCPRE signal.

Dead time insertion

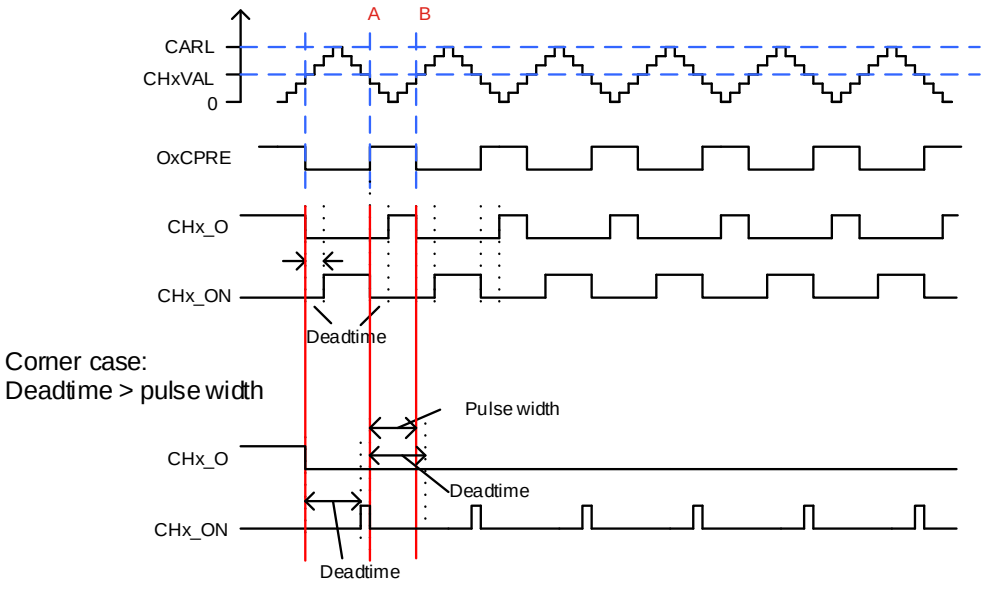
The dead time insertion is enabled when both CHxEN and CHxNEN are configured to 1'b1, it is also necessary to configure POEN&CHPOENx(x=0...2) to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register 0 (TIMERx_CCHP0)*** for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event (TIMERx_CNT = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 12-28. Complementary output with dead time insertion***, CHx_O signal remains at the low level until the end of the dead time delay, while CHx_ON signal will be cleared at once. Similarly, at point B when the channelx match event (TIMERx_CNT = CHxVAL) occurs again, OxCPRE is cleared, and CHx_O signal will be cleared at once, while CHx_ON signal remains at the low level until the end of the dead time delay.

Sometimes, we can see corner cases about the dead time insertion. For example: the dead time delay is greater than or equal to the duty cycle of the CHx_O signal, then the CHx_O signal is always inactive (As shown in ***Figure 12-28. Complementary output with dead time insertion***).

Figure 12-28. Complementary output with dead time insertion



By configuring the DTIENCHx (x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of dead-time insertion function for each pair of channels. When the DTIENCHx (x=0...3) bit is "0", the corresponding channels CHx_O and CHx_ON will not be inserted into the dead-time.

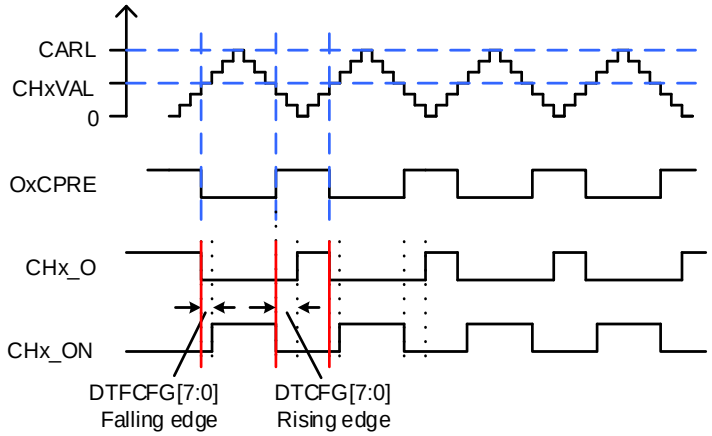
Different dead time insertion

The CHx_O and CHx_ON signal can output with different dead time when the DTDIFEN bit (in the TIMERx_CCHP1register) is set to 1. As shown in ***Figure 12-29. Complementary output with different dead time(DTDIFEN=1)***.

The rising edge dead time of the channel output prepare signal OxCPRE is configured by the DTCFG[7:0] bit-field in the TIMERx_CCHP0 register. And the falling edge dead time of the OxCPRE signal is configured by the DTFCFG[7:0] bit-field in the TIMERx_CCHP1 register.

The DTDIFEN bit must be written before enabling the counter and cannot be modified while CEN=1.

Figure 12-29. Complementary output with different dead time(DTDIFEN=1)



Break function

In this mode, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP0 register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot be set both to active level when break occurs.

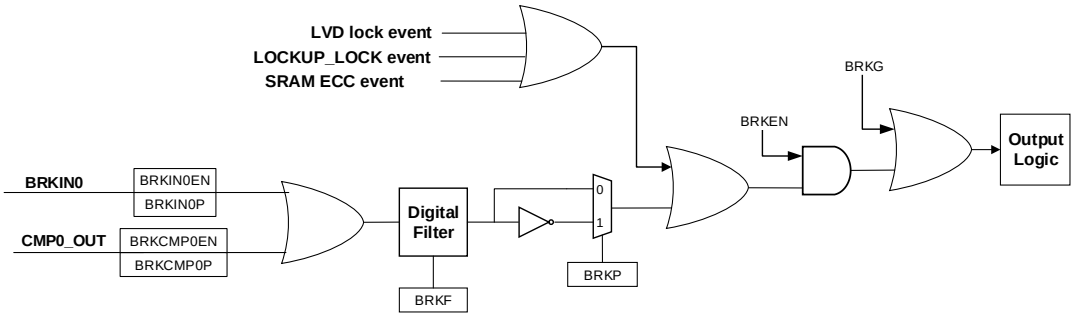
The break event is the result of logic ORed of all sources. The break functions can handle two types of event sources:

- External sources: coming from BRKIN0 inputs;
- System sources: LVD lock event, Cortex®-M33 LOCKUP_LOCK event or SRAM ECC event;
- On-chip peripheral events: input by comparator output.

Break events can also be generated by software using BRKG bit in the TIMERx_SWEVG register.

Refer to ***Figure 12-30. BREAK function logic diagram***, BRKIN0 can select GPIO pins from the TRIGSEL module, which can select by ***Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN)*** and ***Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN).***

Figure 12-30. BREAK function logic diagram



BREAK can be used to handle the faults of system sources, on-chip peripheral events and external sources. When a BREAK event occurs, the outputs are force at an inactive level, or at a predefined level (either active or inactive) after a deadtime duration.

When a break occurs, the POEN bit is cleared asynchronously. As soon as POEN is 0, the level of the CHx_O and CHx_ON outputs are determined by the ISOx and ISOxN bits in the TIMERx_CTL1 register. If IOS = 0, the timer releases the enable output, otherwise, the enable output remains high. When IOS=1, the output behavior of the channel is shown in ***Figure 12-31. Output behavior of the channel in response to BREAK (the break input high active and IOS=1)***. The complementary outputs are first in the reset state, and then the dead time generator is reactivated to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead time.

Figure 12-31. Output behavior of the channel in response to BREAK (the break input high active and IOS=1)

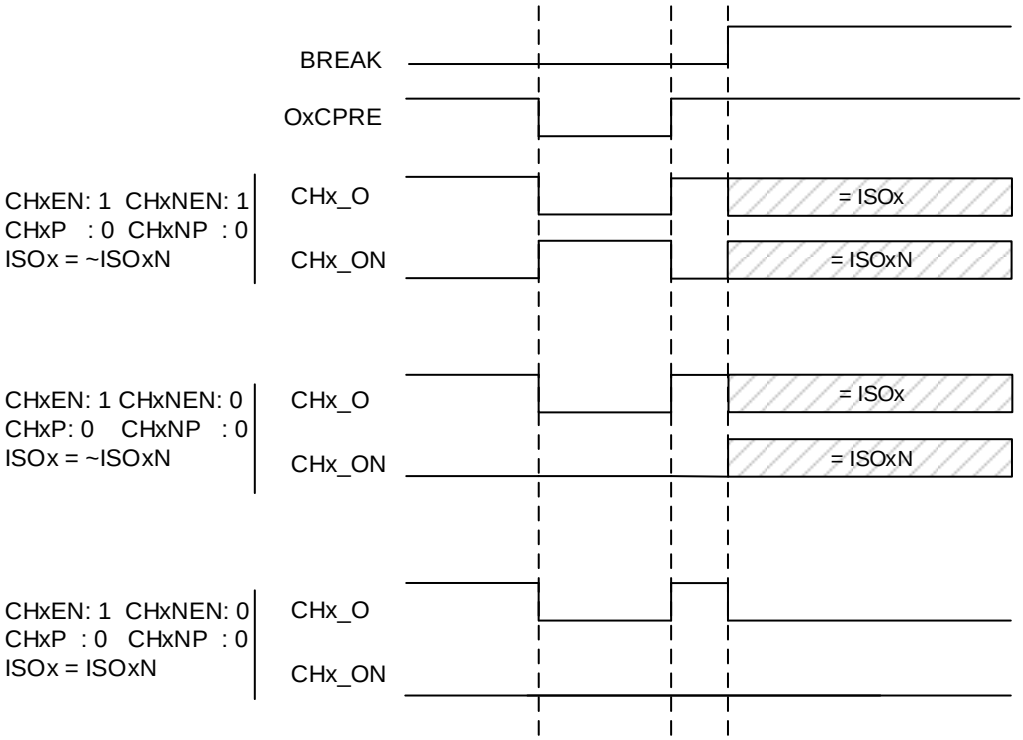


Table 12-5. Output behavior of the channel in response to a BREAK (the break input is

high active)

BREAK inputs	Output Status	
	CHx_O	CHx_ON
High	IOS=1: CHx_O output is inactive and then output to idle level (by IOSx bit) after a deadtime time. IOS=0: CHx_O output disable (inactive).	IOS=1: CHx_ON output is inactive and then output to level (by IOSxN bit) after a deadtime time. IOS=0: CHx_ON output disable (inactive).
Low	CHx_O output disable (inactive).	CHx_ON output disable (inactive).

When a break occurs, the BRKIF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

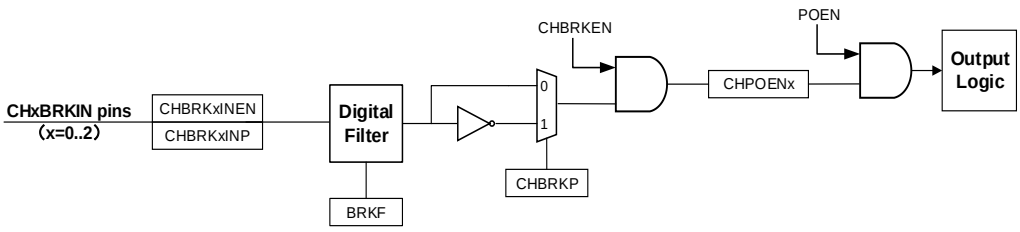
By configuring the BEKENCHx (x=0...3) bit in the TIMERx_CTL2 register to realize the independent control of break function for each pair of channels. When the BEKENCHx(x=0...3) bit is “0” and a break event occurs, the corresponding channels CHx_O and CHx_ON will not be changed and the outputs is keeping.

To enhance the flexibility of the break function, for the advanced timers, each pair of channels has a separate break input CHBRKxIN (x=0...2), and can be individually broken when the corresponding CHBRKxIN (x=0...2) input is active. The channel break can be enabled by setting the CHBRKEN bit in the TIMERx_CCHP0 register. Additionally, the polarity of channel break is configured by setting the CHBRKP bit in the TIMERx_CCHP0 register.

Furthermore, if the channel x break is enabled, CHx_O and CHx_ON (x=0...2) are controlled not only by POEN but also by CHPOENx (x=0...2). The output of CHx and CHx_ON (x=0...2) after being broken is the same as described for the BREAK functionality. Each channel break input has its own enable bit CHBRKxINEN (x=0...2) and polarity control bit CHBRKxINP (x=0...2), and shares the same filter configuration field BRKF[3:0] with BREAK. Please refer to ***Figure 12-32. Channel break function logic diagram***.

When the channel x (x=0...2) break input is active, the corresponding channel break flag bit CHxBRKIF (x=0...2) is set to 1. If the channel x (x=0...2) break input is active for multiple periods, flag bit CHxBRKMIF(x=0...2) will be set to 1. The period counts can be determined by configuring the TIMERx_CHBRKPER register.

Figure 12-32. Channel break function logic diagram



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 with each other to generate the counter value.

Setting TSCFGy[4:0] (y=0, 1, 2) != 5'b00000 to select that the counting direction of timer is determined only by the CI0, only by the CI1, or by the CI0 and the CI1.

The DIR bit is modified by hardware automatically during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in ***Table 12-6. Counting direction in different quadrature decoder signals***. The CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection. The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 12-6. Counting direction in different quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 TSCFG0[4:0]!= 5'b00000	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 TSCFG1[4:0]!= 5'b00000	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 TSCFG2[4:0]!= 5'b00000	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 12-33. Example of counter operation in quadrature decoder interface mode

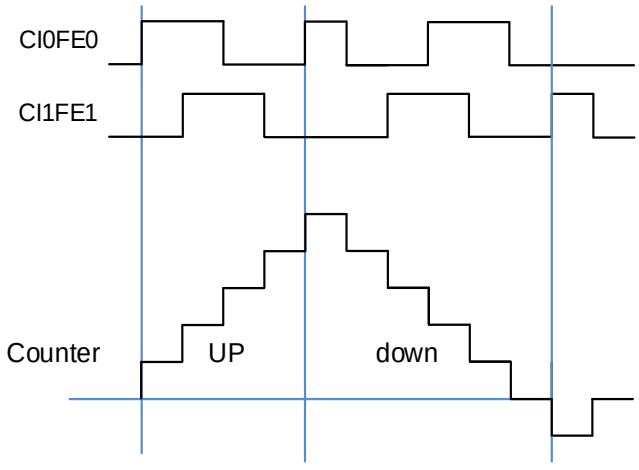
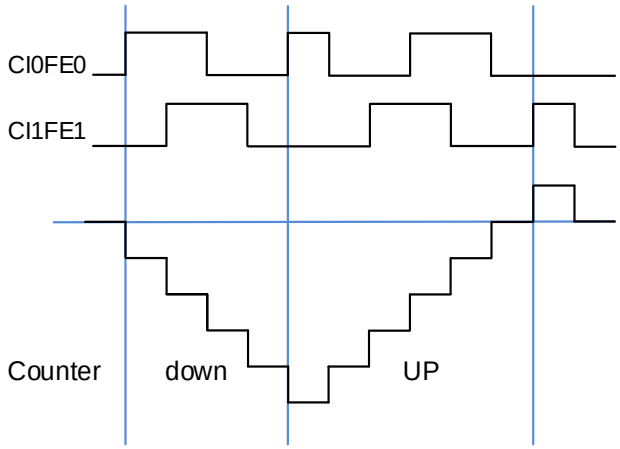


Figure 12-34. Example of quadrature decoder interface mode with CI0FE0 polarity

inverted



Quadrature decoder and non-quadrature decoder clock output

TIMERs can output the decoder clock output signal as TRGO. This function can be enabled by setting the MMC[3:0] bit-field to 4'b1000 in the TIMERx_CTL1 register, and just used in quadrature decoder mode 0~2.

Hall sensor function

Hall sensor is generally used to control BLDC motor; the advanced timer supports this function.

Figure 12-35. Hall sensor is used for BLDC motor shows how to connect the timer and the motor. And two timers are needed. TIMER_in(Advanced/General L0 TIMER) is used to accept three rotor position signals of motor from hall sensors.

Each of the 3 hall sensors provides a pulse which is applied to an input capture pin, then both the speed and position of rotor can be calculated by analyzing the hall sensor signals.

By the internal connection function (TRGO-ITIx), TIMER_in and TIMER_out can be connected. TIMER_out will generate PWM signals to control the speed of BLDC motor based on the ITIx. Then, the feedback circuit is finished, you can change the configuration to fit your request.

Because the advanced/general L0 TIMER has the input XOR function, they can be used as the TIMER_in timer. And the advanced timer has the functions of complementary output and dead time, so it can be used as the TIMER_out timer. Else, based on the timer's 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)

After appropriate interconnected timers are selected and wires are connected, the timers need to be configured. Some key settings are as follows:

- Enable XOR by setting TI0S, then, the change of each input signal will make the CI0

toggle. CH0VAL will record the current value of counter.

- Choose ITIx to trigger commutation by configuring CCUC and CCSE.
- Configure PWM parameters based on the requests.

Figure 12-35. Hall sensor is used for BLDC motor

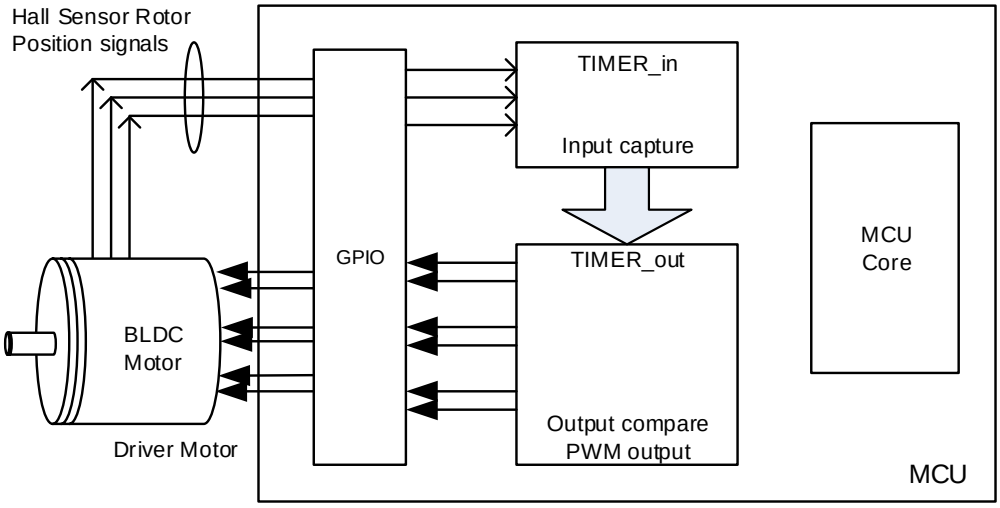
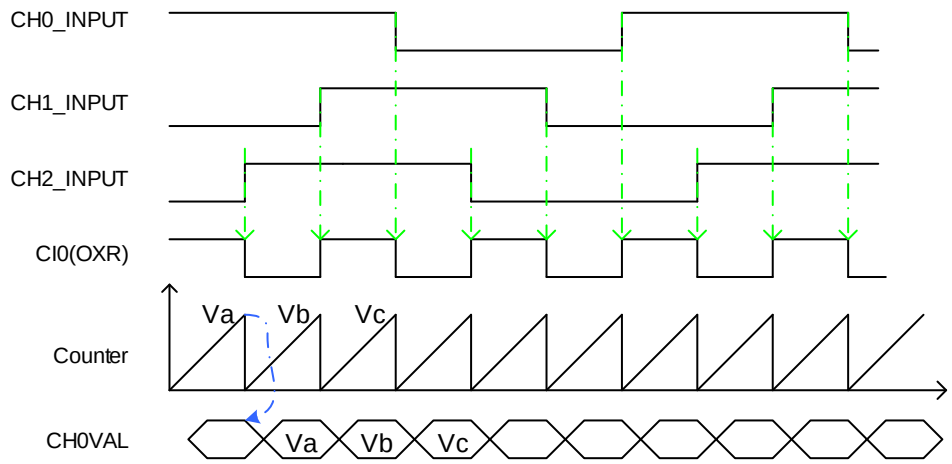
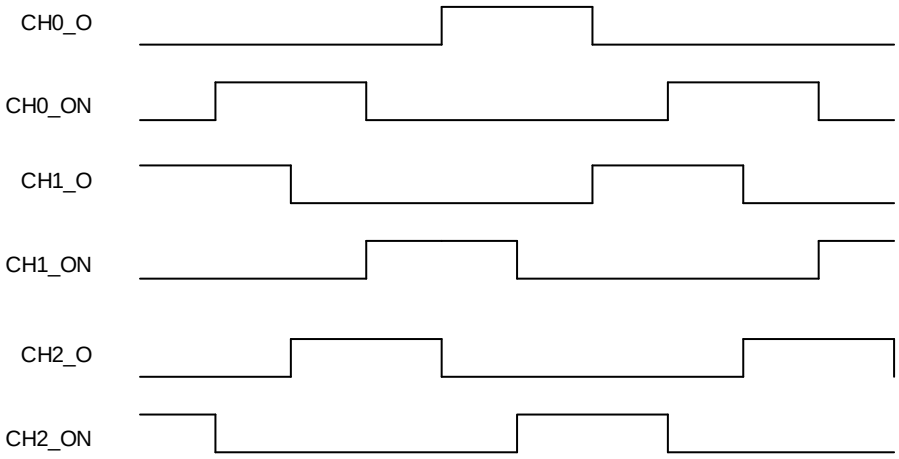


Figure 12-36. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



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

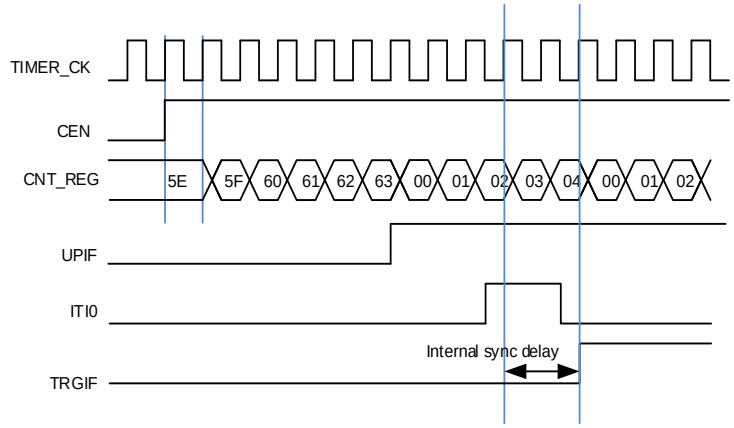
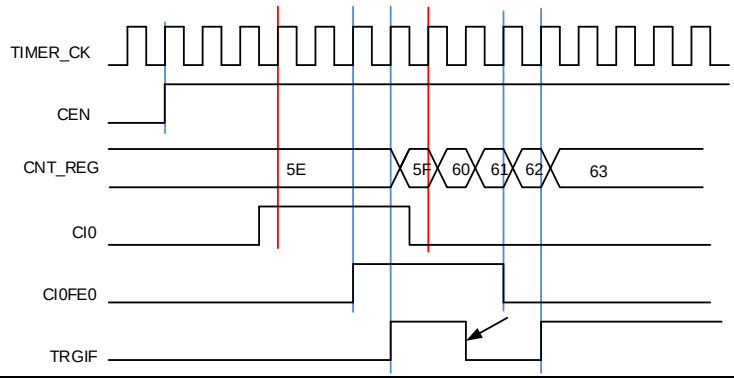


Master-slave management

The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode and so on, which is selected by the TSCFGy[4:0] (y=3..6) in SYSCFG_TIMERxCFG(x=0,7).

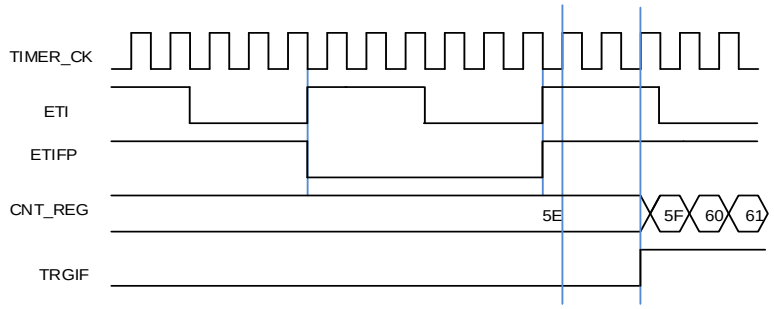
Table 12-7. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode y=4: pause mode y=5: event mode y=6: external clock mode 0	TSCFGy[4:0] 00000:Mode disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED	If CIxFEx (x=0...3) are selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion. If ETIFP (the filtered	For the ITIx, no filter and prescaler can be used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used. For the ETIFP, filter

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
		00110: CI0FE0 00111: CI1FE1 01000: ETIFP(1) 01001: CI2FE2 01010: CI3FE3	output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of trigger input comes.	TSCFG3[4:0] 5'b00001, ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 12-37. 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.	TSCFG4[4:0] =5'b00110, 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 12-38. Pause mode 			
Exam3	**Event mode** The counter will start to	TSCFG5[4:0] =5'b01000,	ETP = 0, the polarity of ETI does not change.	ETPSC = 1, ETI is divided by 2.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	count when a rising edge of trigger input comes.	ETIFP is selected.		ETFC = 0, ETI does not filter.

Figure 12-39. Event mode



Single pulse mode

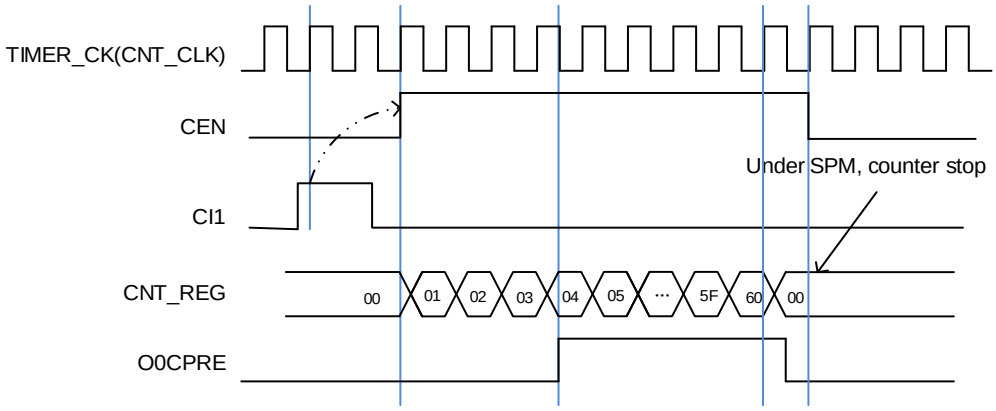
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL bit.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the CEN bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signals will change to, as the compare match event occurs without taking the comparison result into account.

Single pulse mode is also applicable to composite PWM mode (CHxCPWMEN = 1’b1 and CHxMS[2:0] = 3’b000).

Figure 12-40. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60



Timers interconnection

The timers can be internally connected for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the master mode while configuring another timer to be in the slave mode. The following figures show several examples of trigger selection for the master mode and slave mode.

Some interconnection examples:

- TIMER2 as the prescaler for TIMER0

TIMER2 is configured as a prescaler for TIMER0, steps are shown as follows:

1. Configure TIMER2 in master mode and select its update event (UPE) as trigger output (MMC=4'b0010 in the TIMER2_CTL1 register). Then TIMER2 drives a periodic signal on each counter overflow.
2. Configure TIMER2 period (TIMER2_CAR register).
3. Configure TIMER0 in external clock mode 0 and select the TIMER2 as TIMER0 input trigger source (TSCFG6[4:0] = 5'b00011 in the SYSCFG_TIMER0CFG1 register).
4. Start TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
5. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

- Start TIMER0 with TIMER2's enable/update signal

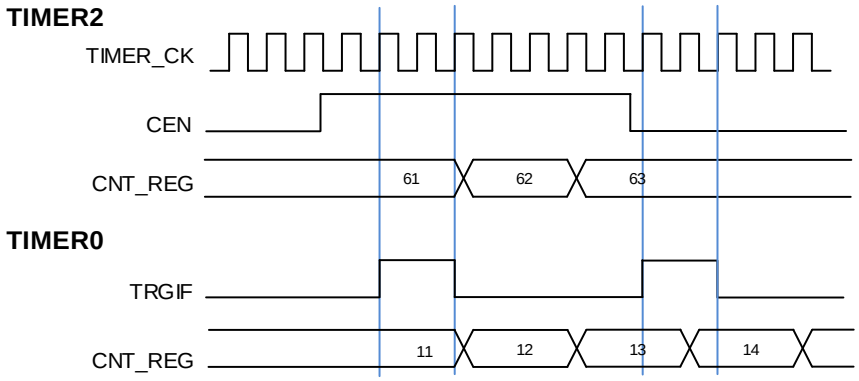
First, enable TIMER0 with the enable signal of TIMER2. Refer to ***Figure 12-41. Trigger mode of TIMER0 controlled by enable signal of TIMER2***. TIMER0 starts counting from its current value with the divided internal clock after being triggered by TIMER2 enable signal output.

When TIMER0 receives the trigger signal, its CEN bit is set automatically and the counter counts until TIMER0 is disabled. Both clock frequency of the counters is divided by 3 from TIMER_CK ($f_{PSC_CLK} = f_{TIMER_CK}/3$). Steps are shown as follows:

1. Configure TIMER2 in master mode to send its enable signal as trigger output (MMC=4'b0001 in the TIMER2_CTL1 register).
2. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TSCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).

3. Start TIMER2 by writing 1 to the CEN bit (TIMER2_CTL0 register).

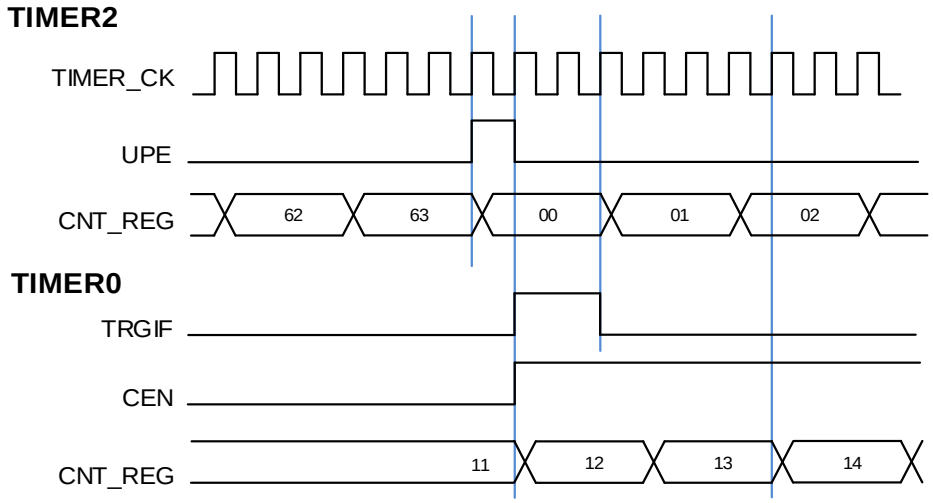
Figure 12-41. Trigger mode of TIMER0 controlled by enable signal of TIMER2



In this example, the update event can also be used as trigger source instead of enable signal. Refer to ***Figure 12-42. Trigger mode of TIMER0 controlled by update signal of TIMER2***. Steps are shown as follows:

1. Configure TIMER2 in master mode to send its update event (UPE) as trigger output (MMC=4'b0010 in the TIMER2_CTL1 register).
2. Configure the TIMER2 period (TIMER2_CARL registers).
3. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TSCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 12-42. Trigger mode of TIMER0 controlled by update signal of TIMER2



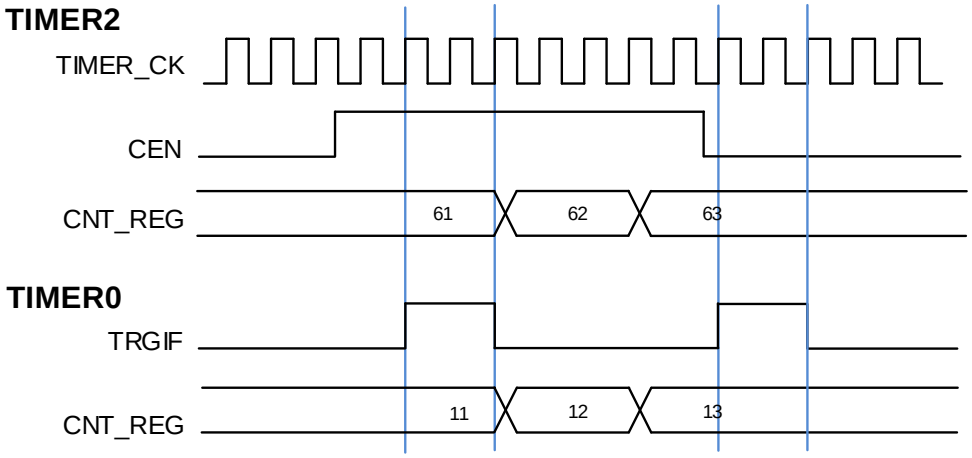
- Enable TIMER0 to count with the enable/O0CPRE signal of TIMER2.

In this example, TIMER0 is enabled with the enable signal of TIMER2. Refer to ***Figure 12-43. Pause mode of TIMER0 controlled by enable signal of TIMER2***. TIMER0 counts with the divided internal clock only when TIMER2 is enabled. Both clock frequency of the counters is

divided by 3 from TIMER_CK ($f_{PSC_CLK} = f_{TIMER_CK}/3$). Steps are shown as follows:

1. Configure TIMER2 in master mode and output enable signal as trigger output (MMC=4'b0001 in the TIMER2_CTL1 register).
2. Configure TIMER0 in pause mode and select the TIMER2 as TIMER0 input trigger source (TSCFG4[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
3. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
4. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).
5. Stop TIMER2 by writing '0' to the CEN bit (TIMER2_CTL0 register).

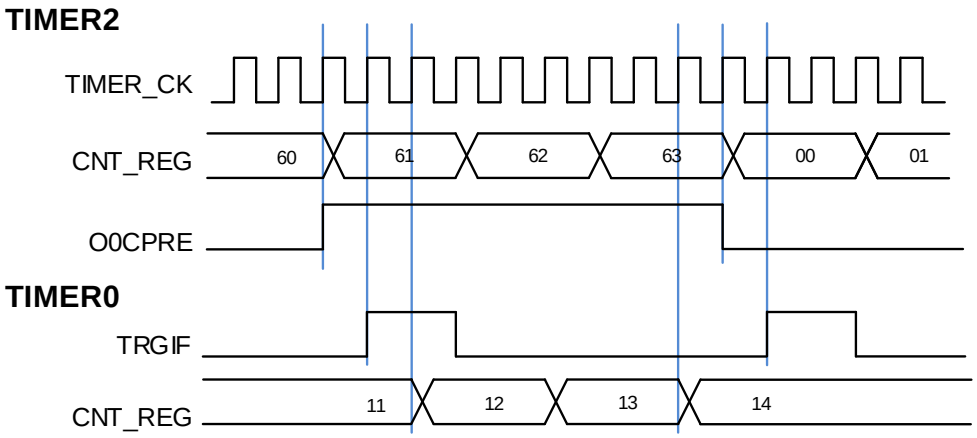
Figure 12-43. Pause mode of TIMER0 controlled by enable signal of TIMER2



In this example, O0CPRE can also be used as trigger source instead of enable signal output. Steps are shown as follows:

1. Configure TIMER2 in master mode and O0CPRE as trigger output (MMS=3'b100 in the TIMER2_CTL1 register).
2. Configure the TIMER2 O0CPRE waveform (TIMER2_CHCTL0 register).
3. Configure TIMER0 in pause mode and select the TIMER2 as TIMER0 input trigger source (TSCFG4[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).
4. Enable TIMER0 by writing '1' to the CEN bit (TIMER0_CTL0 register).
5. Start TIMER2 by writing '1' to the CEN bit (TIMER2_CTL0 register).

Figure 12-44. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2



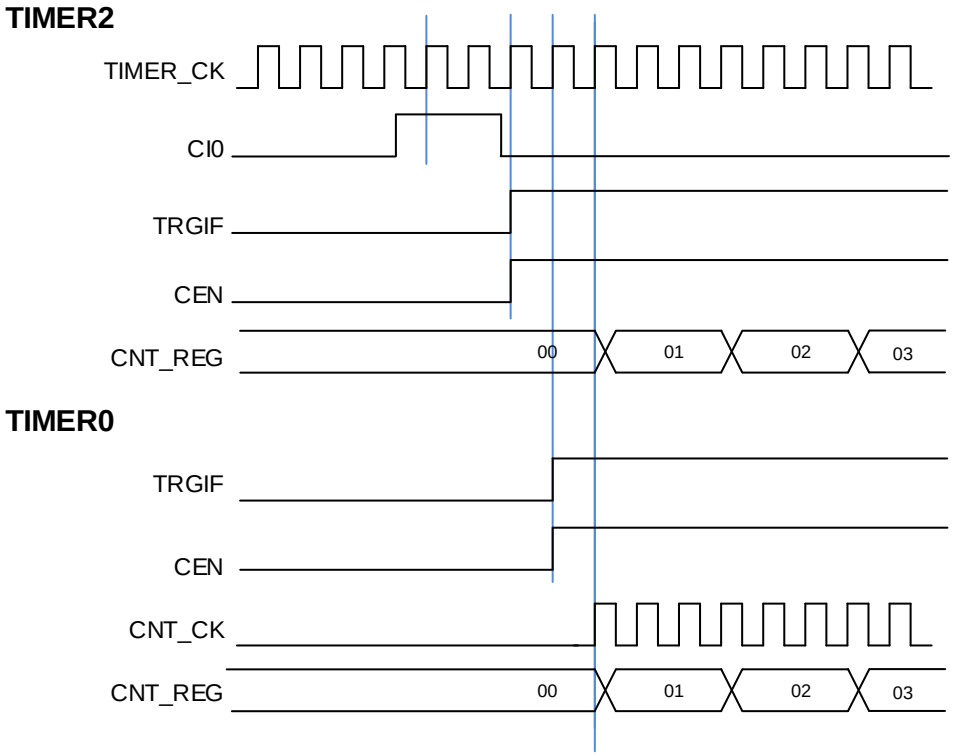
- Using an external trigger to start two timers synchronously.

The start of TIMER0 is triggered by the enable signal of TIMER2, and TIMER2 is triggered by its CI0 input rising edge. To ensure that two timers start synchronously, TIMER2 must be configured in master/slave mode. Steps are shown as follows:

1. Configure TIMER2 in event mode and select the CI0F_ED as TIMER2 input trigger source (TSCFG5[4:0] = 5'b00101 in the_SYSCFG_TIMER2CFG0 register).
2. Configure TIMER2 in master/slave mode by writing MSM=1 (TIMER2_SMCFG register).
3. Configure TIMER0 in event mode and select the TIMER2 as TIMER0 input trigger source (TSCFG5[4:0] = 5'b00011 in the_SYSCFG_TIMER0CFG0 register).

When the CI0 signal of TIMER2 generates a rising edge, two timer counters start counting synchronously with the internal clock and both TRGIF flags are set.

Figure 12-45. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2



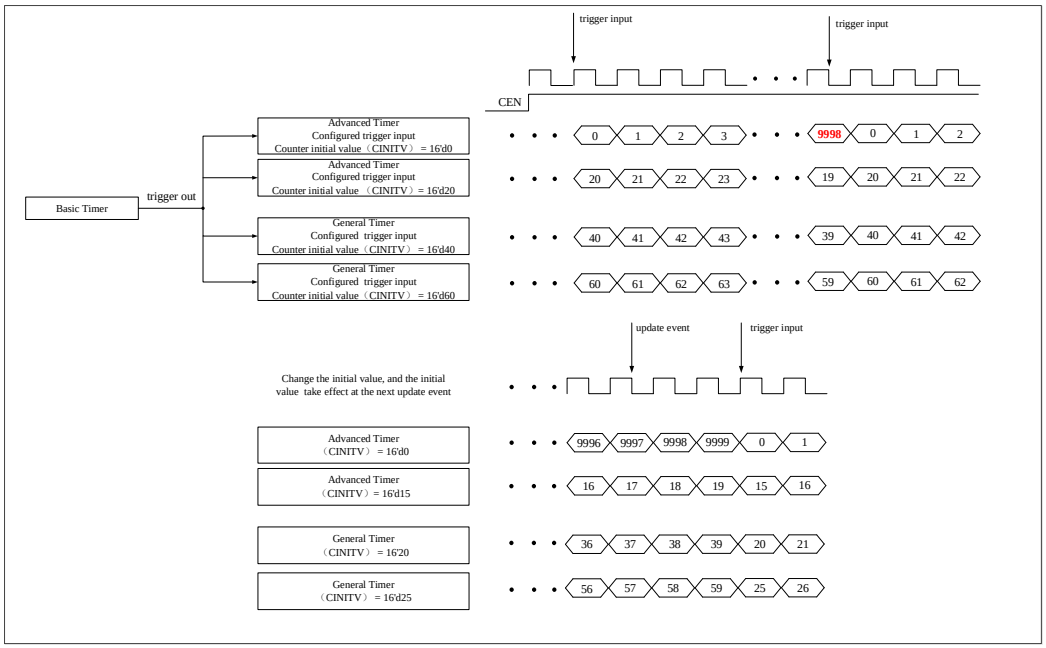
Counter synchronization and counter initial direction and value refresh

In some interconnection configurations, several timers are triggered and synchronized to start counting at the same time. several timers may have phase offsets after long counts and the phase offsets is eliminated by periodically refreshing the synchronization counter with an external trigger, it is possible to control the initial value of the phase between several timers by configuring counter initial control register(TIMERx_CINITV).

Four slave timers are configured to exist in the same trigger input source. The initial values of the timer can be preload by setting the CINITVEN bit of the TIMERx_CINITCTL register. In the following figure, the initial load of the four slave timers is 0,20,40,60 respectively.So there's an arithmetic shift of 20 CK_TIMER.

The 9998 count in the figure below shows the phase shift between the counter buses of different timers after a long count. The four timers count value is refreshed synchronization by trigger output of basic timer, and the phase offset is avoided.

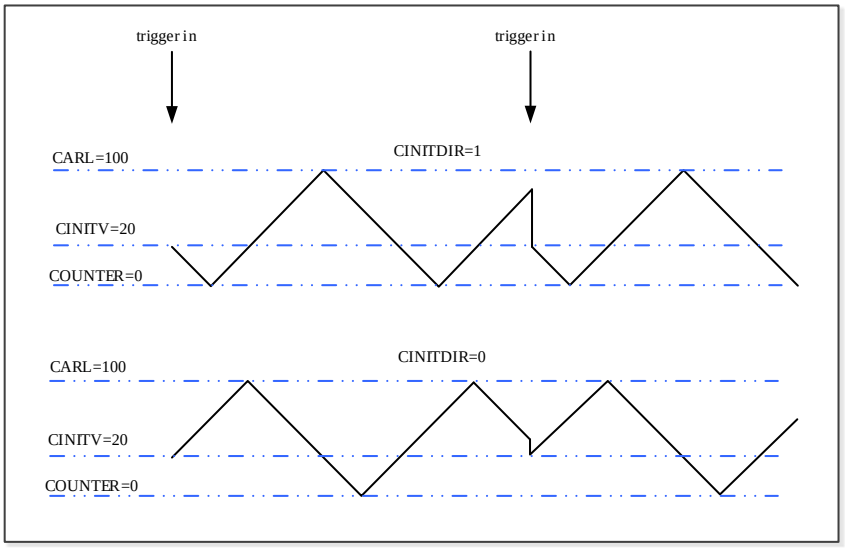
Figure 12-46. Triggering four timers by trigger out of basic timer



The counting direction after resetting the counter can be configured by CINITDIR bit in TIMERx_CINITCTL register when the counter is in center-aligned counting mode. CINITDIR only takes effect when CINITVEN in TIMERx_CINITCTL register is enabled, and CINITDIR does not take effect when the counter is in up/down counting mode.

When CINITDIR is 0, the counting direction is up after resetting the counter. When CINITDIR is 1, the count direction is down after resetting the counter.

Figure 12-47. Counting direction in CINITDIR is 0 or 1



The counter initial direction and value also can refresh by soft synchronization event. When SWSYNCG bit in TIMERx_CINITCTL register is set, a soft synchronization event generated, and TIMERx can refresh the counter initial direction and value.

When the advanced TIMERx is used in master mode to synchronize the other advanced timers, the MMC[3:0] bit-field (in TIMERx_CTL1 register) should set to 4’b1001. The soft

synchronization event (generated by setting the SWSYNCG to 1) of TIMERx which can output as the TRGO signal, is used to synchronize the other advanced timers.

Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register is set to 1, the TIMERx counter stops.

12.1.5. Registers definition (TIMERx, x=0, 7)

TIMER0 base address: 0x4001 2C00

TIMER7 base address: 0x4001 3400

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 ratio between CK_TIMER (the timer clock) and DTS (the dead time and sampling clock) which is used for the dead time generator and the digital filter. 00: f_{DTS}= f_{CK_TIMER} 01: f_{DTS}= f_{CK_TIMER} /2 10: f_{DTS}= f_{CK_TIMER} /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when the counter is counting down, compare interrupt flag of channels can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when the counter is counting up, compare interrupt flag of channels can be set. 11: Center-aligned and counting up/down assert mode. The counter counts in

		center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Both when the counter is counting up and counting down, compare interrupt flag of channels can be set. After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode or decoder mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generates an update interrupt or a DMA request: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Only counter overflow/underflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UPG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

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
CCUC[2:1]		Reserved				MMC[3]	Reserved								
rw						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISO3N	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC[0]	Reserved	CCSE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

Bits	Fields	Descriptions
31:30	CCUC[2:1]	Commutation control shadow register update control Refer to CCUC [0] description.
29:26	Reserved	Must be kept at reset value.
25	MMC[3]	Master mode control Refer to MMC[2:0] description.
24:16	Reserved	Must be kept at reset value.
15	ISO3N	Idle state of multi mode channel 3 complementary output. Refer to ISO0N bit.
14	ISO3	Idle state of channel 3 output. Refer to ISO0 bit.
13	ISO2N	Idle state of multi mode channel 2 complementary output. Refer to ISO0N bit
12	ISO2	Idle state of channel 2 output Refer to ISO0 bit
11	ISO1N	Idle state of multi mode channel 1 complementary output Refer to ISO0N bit
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of multi mode channel 0 complementary output 0: When POEN&CHPOENx(x=0...2) bit is reset, CH0_ON is set low. 1: When POEN&CHPOENx(x=0...2) bit is reset, CH0_ON is set high. This bit can be modified only when PROT[1:0] bits in TIMERx_CCHP0 register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN&CHPOENx(x=0...2) bit is reset, CH0_O is set low. 1: When POEN&CHPOENx(x=0...2) 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_CCHP0 register is 00.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input.

		1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 0000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 0001: Enable. This mode is used to start several timers at the same time or control a slave timer to be enabled in a period. In this mode, the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 0010: Update. In this mode, the master mode controller selects the update event as TRGO. 0011: Capture/compare pulse. In this mode, the master mode controller generates a TRGO pulse when a capture or a compare match occurs in channel 0. 0100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO. 0101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO. 0110: Compare. In this mode, the master mode controller selects the O2CPRE signal as TRGO. 0111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO. 1000: Decoder clock output. In this mode, the master mode controller selects the decoder clock signal as TRGO. This value just used in quadrature decoder mode 0~2. 1001: In this mode, the master mode controller selects the soft synchronization event signal (generated by setting the SWSYNCG to 1) as TRGO. Others: Reserved.
3	DMAS	DMA request source selection 0: DMA request of CHx is sent when capture/compare event occurs. 1: DMA request of channel CHx is sent when update event occurs.
2	CCUC[0]	Commutation control shadow register update control The CCUC[2:1] and CCUC[0] field are used to control the commutation control shadow register update. When the commutation control shadow registers (for CHxEN, CHxNEN and CHxCOMCTL bits) are enabled (CCSE=1), the update control of the shadow registers with the CCUC[2:0] bit-field are shown as below: 000: The shadow registers update when CMTG bit is set. 001: The shadow registers update when CMTG bit is set or a rising edge of TRGI occurs.

		100: The shadow registers update when the counter generates an overflow event. 101: The shadow registers update when the counter generates an underflow event. 110: The shadow registers update when the counter generates an overflow or underflow event. Others: Reserved When a channel does not have a complementary output, this bit has no effect. **Note**: When CCUC[2:0] bit-field are set to 100, 101 and 110, the update of the shadow registers also considers the value the CCUSEL bit in the TIMERx_CFG register.
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 when commutation event comes. When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal. 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of slave mode controller is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TSCFGy[4:0](y=3,4,5) bits must not be

		5'b01000 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in TSCFG6[4:0] bit-field in SYSCFG_TIMERxCFG1 register.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: ETIFP frequency divided by 2 10: ETIFP frequency divided by 4 11: ETIFP frequency divided by 8
11:8	ETFC[3:0]	External trigger filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample ETIFP signal and the length of the digital filter applied to ETIFP. 0000: Filter disabled. f_{SAMP}= f_{DTS}, N=1. 0001: f_{SAMP}= f_{CK_TIMER}, N=2. 0010: f_{SAMP}= f_{CK_TIMER}, N=4. 0011: f_{SAMP}= f_{CK_TIMER}, N=8. 0100: f_{SAMP}=f_{DTS}/2, N=6. 0101: f_{SAMP}=f_{DTS}/2, N=8. 0110: f_{SAMP}=f_{DTS}/4, N=6. 0111: f_{SAMP}=f_{DTS}/4, N=8. 1000: f_{SAMP}=f_{DTS}/8, N=6. 1001: f_{SAMP}=f_{DTS}/8, N=8. 1010: f_{SAMP}=f_{DTS}/16, N=5. 1011: f_{SAMP}=f_{DTS}/16, N=6. 1100: f_{SAMP}=f_{DTS}/16, N=8. 1101: f_{SAMP}=f_{DTS}/32, N=5. 1110: f_{SAMP}=f_{DTS}/32, N=6. 1111: f_{SAMP}=f_{DTS}/32, N=8.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIE	CH2COMADDIE	CH1COMADDIE	CH0COMADDIE	Reserved											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3COMADDIE	Channel 3 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode.
30	CH2COMADDIE	Channel 2 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode.
29	CH1COMADDIE	Channel 1 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode.
28	CH0COMADDIE	Channel 0 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode.
27:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable 0: Disabled 1: Enabled
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: Disabled 1: Enabled

11	CH2DEN	Channel 2 capture/compare DMA request enable 0: Disabled 1: Enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: Disabled 1: Enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7	BRKIE	Break interrupt enable 0: Disabled 1: Enabled
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5	CMTIE	Commutation interrupt enable 0: Disabled 1: Enabled
4	CH3IE	Channel 3 capture/compare interrupt enable 0: Disabled 1: Enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: Disabled 1: Enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: Disabled 1: Enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDIF	CH2COMADDIF	CH1COMADDIF	CH0COMADDIF	Reserved											
rc_w0	rc_w0	rc_w0	rc_w0												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SYSBIF	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	rc_w0

Bits	Fields	Descriptions
31	CH3COMADDIF	Channel 3 additional compare interrupt flag. Refer to CH0COMADDIF description.
30	CH2COMADDIF	Channel 2 additional compare interrupt flag. Refer to CH0COMADDIF description.
29	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
28	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred **Note:** This flag just used in composite PWM mode.
27:14	Reserved	Must be kept at reset value.
13	SYSBIF	System source break interrupt flag This flag is set by hardware when the system sources are active, and cleared by software if the system sources are inactive. 0: No system source break interrupt occurred 1: System source break interrupt occurred **Note**: When this bit is set, this bit must be cleared by software before the channel outputs are restored.
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 This flag is set by hardware when the BREAK input is active, and cleared by software if the BREAK input is not at active level. 0: No active level on break input has been detected. 1: An active level on break input has been detected.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when the commutation event of channel occurs, and cleared by software. 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4	CH3IF	Channel 3 capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred

1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3COMADDG	CH2COMADDG	CH1COMADDG	CH0COMADDG	Reserved											
w	w	w	w												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31	CH3COMADDG	Channel 3 additional compare event generation. Refer to CH0COMADDG description.
30	CH2COMADDG	Channel 2 additional compare event generation. Refer to CH0COMADDG description.
29	CH1COMADDG	Channel 1 additional compare event generation. Refer to CH0COMADDG description.
28	CH0COMADDG	Channel 0 additional compare event generation. This bit is set by software to generate a compare event in channel 0 additional, it is automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event 1: Generate a channel 0 additional compare event **Note:** This bit just used in composite PWM mode.
27:8	Reserved	Must be kept at reset value.
7	BRKG	BREAK event generation This bit is set by software to generate an event and cleared by hardware automatically. When this bit is set, the POEN bit will be cleared and BRKIF flag will be set. 0: No generate a BREAK event 1: Generate a BREAK event
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register will be set, related interrupt or DMA transfer can occur if enabled.

0: No generate a trigger event
1: Generate a trigger event

5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel commutation update event
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS [2]	CH0MS [2]	CH1COM ADDSEN	CH0COM ADDSEN	CH1ADDU PS	CH0ADDU PS	Reserved	CH1COM CTL[3]	Reserved							CH0COM CTL[3]
		Reserved					Reserved								Reserved

rw	rw	rw	rw	rw	rw	rw									rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000.
27	CH1ADDUPS	Channel 1 additional update source 0: TIMERx_CH1COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH1COMV_ADD register is updated when the counter matches the value of CH1VAL.
26	CH0ADDUPS	Channel 0 additional update source 0: TIMERx_CH0COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH0COMV_ADD register is updated when the counter matches the value of CH0VAL.
25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description
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[2:0] 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. The CH1MS[2:0] bit-field is writable only when the channel is not active (CH1EN bit in TIMERx_CHCTL2 register is reset). 000: Channel 1 is configured as output. 001: Channel 1 is configured as input, IS1 is connected to CI1FE1. 010: Channel 1 is configured as input, IS1 is connected to CI0FE1. 011: Channel 1 is configured as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100~111: Reserved.
7	CH0COMCEN	Channel 0 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the O0CPRE signal will be cleared. 0: Channel 0 output compare clear disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[3] and CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O and CH0_ON. The active level of O0CPRE is high, while the active level of CH0_O and CH0_ON depend on CH0P and CH0NP bits. 0000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 0010: Clear the channel output on match. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 0011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 0100: Force low. O0CPRE is forced low level. 0101: Force high. O0CPRE is forced high level. 0110: PWM mode 0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV, otherwise it is inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV, otherwise it is active. 0111: PWM mode 1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV, otherwise it is active. When counting down,

O0CPRE is active as long as the counter is larger than TIMERx_CH0CV, otherwise it is inactive.

1000: Reserved.

1001: Reserved.

1010: Asymmetric PWM mode 0. Only for Center aligned mode, when counting up, the channel output is inactive when the counter is matched with the value of TIMERx_CH0CV, when counting down, the channel output is active when the counter is matched with the value of TIMERx_CH0CV_ADD.

1011: Asymmetric PWM mode 1. Only for Center aligned mode, when counting up, the channel output is active when the counter is matched with the value of TIMERx_CH0CV, when counting down, the channel output is inactive when the counter is matched with the value of TIMERx_CH0CV_ADD.

1100~1111: Reserved.

Note: In the composite PWM mode (CH0CPWMEN = 1'b1 and CH0MS = 3'b000), the PWM signal output in channel 0 is composited by TIMERx_CH0CV and TIMERx_CH0COMV_ADD. Please refer to ***Composite PWM mode*** for more details.

If configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from "Timing" mode to "PWM" mode or the result of the comparison changes.

When the outputs of CH0_O and CH0_ON are complementary, this bit-field is preloaded.

If CCSE =1, this bit-field will only be updated when a channel commutation event is generated.

This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000 (compare mode).

3 CH0COMSEN Channel 0 compare output shadow enable

When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled.

0: Channel 0 output compare shadow disabled

1: Channel 0 output compare shadow enabled

The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).

This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000.

2 CH0COMFEN Channel 0 output compare fast enable

0: Channel 0 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and CH0_O is set to the compare level.The minimum delay from an edge on the result of the comparison to activate CH0_O output is 5 clock cycles.

1: Channel 0 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and CH0_O is set to the compare level.The minimum delay from an edge on the trigger input to

		activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (CH0EN bit in TIMERx_CHCTL2 register is reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Same as output compare mode.
30	CH0MS[2]	Channel 0 I/O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8. 1010: $f_{SAMP}=f_{DTS}/16$, N=5.

1011: $f_{SAMP}=f_{DTS}/16$, N=6.

1100: $f_{SAMP}=f_{DTS}/16$, N=8.

1101: $f_{SAMP}=f_{DTS}/32$, N=5.

1110: $f_{SAMP}=f_{DTS}/32$, N=6.

1111: $f_{SAMP}=f_{DTS}/32$, N=8.

3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection Same as output compare mode.

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3MS [2]	CH2MS [2]	CH3COM ADDSEN	CH2COM ADDSEN	CH3ADDU PS	CH2ADDU PS	Reserved	CH3COM CTL[3]	Reserved							CH2COM CTL[3]
		Reserved					Reserved								Reserved
rw	rw	rw	rw	rw	rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	CH3MS[2]	Channel 3 I/O mode selection Refer to CH3MS[1:0]description.
30	CH2MS[2]	Channel 2 I/O mode selection Refer to CH2MS[1:0] description.
29	CH3COMADDSEN	Channel 3 additional compare output shadow enable Refer to CH2COMADDSEN description.
28	CH2COMADDSEN	Channel 2 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2COMV_ADD register which updates at each update event will be enabled.

		0: Channel 2 additional compare shadow disabled 1: Channel 2 additional compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH2MS bit-field is 000.
27	CH3ADDUPS	Channel 3 additional update source 0: TIMERx_CH3COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH3COMV_ADD register is updated when the counter matches the value of CH3VAL.
26	CH2ADDUPS	Channel 2 additional update source 0: TIMERx_CH2COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH2COMV_ADD register is updated when the counter matches the value of CH2VAL.
25	Reserved	Must be kept at reset value.
24	CH3COMCTL[3]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH2COMCTL[3]	Channel 2 compare output control Refer to CH2COMCTL[2:0] description
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH2COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH2COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH2COMFEN description.
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[2:0] bit-field is writable only when the channel is not active (CH3EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 3 is configured as output. 01: Channel 3 is configured as input, IS3 is connected to CI3FE3. 10: Channel 3 is configured as input, IS3 is connected to CI2FE3. 11: Channel 3 is configured as input, IS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register).

		100~111: Reserved.
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, the O2CPRE signal is cleared when high level is detected on ETIFP input. 0: Channel 2 output compare clear disabled 1: Channel 2 output compare clear enabled
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field controls the behavior of O2CPRE which drives CH2_O and CH2_ON. The active level of O2CPRE is high, while the active level of CH2_O and CH2_ON depend on CH2P and CH2NP bits. 0000: Timing mode. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 0001: Set the channel output on match. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV. 0010: Clear the channel output on match. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV. 0011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV. 0100: Force low. O2CPRE is forced low level. 0101: Force high. O2CPRE is forced high level. 0110: PWM mode 0. When counting up, O2CPRE is active as long as the counter is smaller than TIMERx_CH2CV, otherwise it is inactive. When counting down, O2CPRE is inactive as long as the counter is larger than TIMERx_CH2CV, otherwise it is active. 0111: PWM mode 1. When counting up, O2CPRE is inactive as long as the counter is smaller than TIMERx_CH2CV, otherwise it is active. When counting down, O2CPRE is active as long as the counter is larger than TIMERx_CH2CV, otherwise it is inactive. 1000: Reserved. 1001: Reserved. 1010: Asymmetric PWM mode 0. Only for Center aligned mode, when counting up, the channel output is inactive when the counter is matched with the value of TIMERx_CH2CV, when counting down, the channel output is active when the counter is matched with the value of TIMERx_CH2CV_ADD. 1011: Asymmetric PWM mode 1. Only for Center aligned mode, when counting up, the channel output is active when the counter is matched with the value of TIMERx_CH2CV, when counting down, the channel output is inactive when the counter is matched with the value of TIMERx_CH2CV_ADD. 1100~1111: Reserved. **Note**: In the composite PWM mode (CH2CPWMEN = 1'b1 and CH2MS = 3'b000), the PWM signal output in channel 2 is composited by TIMERx_CH2CV and TIMERx_CH2COMV_ADD. Please refer to ***Composite PWM mode*** for more

		details. If configured in PWM mode, the O2CPRE level changes only when the output compare mode switches from “Timing” mode to “PWM” mode or the result of the comparison changes. When the outputs of CH2_O and CH2_ON are complementary, this bit-field is preloaded. If CCSE =1, this bit-field will only be updated when a channel commutation event is generated. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH2MS bit-field is 000(compare mode).
3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disabled 1: Channel 2 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH2MS bit-field is 000.
2	CH2COMFEN	Channel 2 output compare fast enable 0: Channel 2 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and CH2_O is set to the compare level.The minimum delay from an edge on the result of the comparison to activate CH2_O output is 5 clock cycles. 1: Channel 2 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and CH2_O is set to the compare level.The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active (CH2EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 2 is configured as output. 01: Channel 2 is configured as input, IS2 is connected to CI2FE2. 10: Channel 2 is configured as input, IS2 is connected to CI3FE2. 11: Channel 2 is configured as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=0,7) register). 100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions

31	CH3MS[2]	Channel 3 I/O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I/O mode selection Same as output compare mode.
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8. 1010: $f_{SAMP}=f_{DTS}/16$, N=5. 1011: $f_{SAMP}=f_{DTS}/16$, N=6. 1100: $f_{SAMP}=f_{DTS}/16$, N=8. 1101: $f_{SAMP}=f_{DTS}/32$, N=5. 1110: $f_{SAMP}=f_{DTS}/32$, N=6. 1111: $f_{SAMP}=f_{DTS}/32$, N=8.
3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode.

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3PERFOREN	CH2PERFOREN	CH1PERFOREN	CH0PERFOREN	Reserved											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	CH3NEN	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	rw	rw

Bits	Fields	Descriptions
31	CH3PERFOREN	Channel 3 in the composite PWM or asymmetric PWM mode, the period point match moment O3CPRE level setting Refer to CH0PERFOREN description.
30	CH2PERFOREN	Channel 2 in the composite PWM or asymmetric PWM mode, the period point match moment O2CPRE level setting Refer to CH0PERFOREN description.
29	CH1PERFOREN	Channel 1 in the composite PWM or asymmetric PWM mode, the period point match moment O1CPRE level setting Refer to CH0PERFOREN description.
28	CH0PERFOREN	Channel 0 in the composite PWM or asymmetric PWM mode, the period point match moment O0CPRE level setting 0: disable 1: In the composite PWM0 (with PWM mode 0) or asymmetric PWM 0 mode, the level will be forced to high at the time of period point matching; In the composite PWM1 (with PWM mode 1) or asymmetric PWM 1 mode, the level at the matching time of the period point is forced to low. **Note:** In the central alignment mode, the period point is the counter underflow time; In the edge-aligned mode, the period point is the counter flow moment.
27:16	Reserved	Must be kept at reset value.
15	CH3NP	Channel 3 complementary output polarity Refer to CH0NP description
14	CH3NEN	Channel 3 complementary output enable Refer to CH0NEN description
13	CH3P	Channel 3 capture/compare polarity Refer to CH0P description.
12	CH3EN	Channel 3 capture/compare 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 polarity Refer to CH0P description.
8	CH2EN	Channel 2 capture/compare 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 polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare 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_CCHP0 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 channel 0. 0: Channel 0 complementary output disabled 1: Channel 0 complementary output enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, these bits specifie the channel 0 input signal's polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for channel 0 input signals.

		00: channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted/ both channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 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
Reserved															

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register 0 (TIMERx_CREP0)

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								CREP0[7:0]							

rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition

counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only.

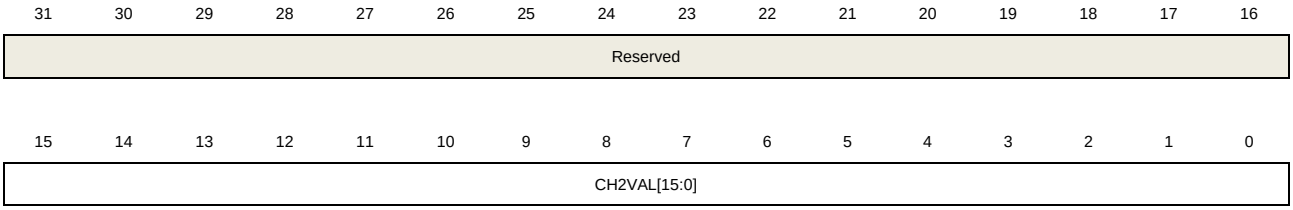
When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

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



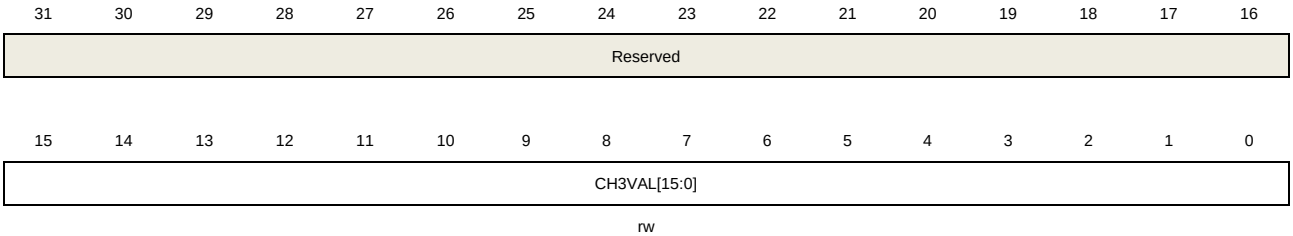
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only.

When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register 0 (TIMERx_CCHP0)

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
CHBRKP	CHBRKEN	Reserved										BRKF[3:0]			
rw	rw											rw			
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	CHBRKP	Channel BREAK input signal polarity This bit specifies the polarity of the channel BREAK input signal. 0: Channel BREAK input active low 1: Channel BREAK input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
30	CHBRKEN	Channel BREAK input signal enable This bit can be set to enable the channel BREAK input signal. 0: Channel BREAK input disabled 1: Channel BREAK input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
29:20	Reserved	Must be kept at reset value.
19:16	BRKF[3:0]	BREAK input signal filter An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BREAK input signal and the length of the digital filter applied to BREAK. 0000: Filter disabled. BREAK act asynchronously, N=1 0001: $f_{SAMP} = f_{CK_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK_TIMER}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8

1000: $f_{SAMP} = f_{DTS}/8$, N=6
1001: $f_{SAMP} = f_{DTS}/8$, N=8
1010: $f_{SAMP} = f_{DTS}/16$, N=5
1011: $f_{SAMP} = f_{DTS}/16$, N=6
1100: $f_{SAMP} = f_{DTS}/16$, N=8
1101: $f_{SAMP} = f_{DTS}/32$, N=5
1110: $f_{SAMP} = f_{DTS}/32$, N=6
1111: $f_{SAMP} = f_{DTS}/32$, N=8
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

15 POEN

Primary output enable
This bit is set by software or automatically set by hardware depending on the OAEN bit. It is cleared asynchronously by hardware as soon as the break input is active. 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: Channel outputs are disabled or forced to idle state.
1: Channel outputs are enabled.

14 OAEN

Output automatic enable
This bit specifies whether the POEN&CHPOENx(x=0...2) bit can be set automatically by hardware.
0: POEN&CHPOENx(x=0...2) cannot be set by hardware.
1: POEN&CHPOENx(x=0...2) can be set by hardware automatically at the next update event, if the break input is not active.
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

13 BRKP

BREAK input signal polarity
This bit specifies the polarity of the BREAK input signal.
0: BREAK input active low
1: BREAK input active high
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

12 BRKEN

BREAK input signal enable
This bit can be set to enable the BREAK input signal
0: BREAK input disabled
1: BREAK input enabled
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

11 ROS

Run mode “off-state” enable
When POEN&CHPOENx(x=0...2) 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 12-4. Complementary outputs controlled by parameters***.

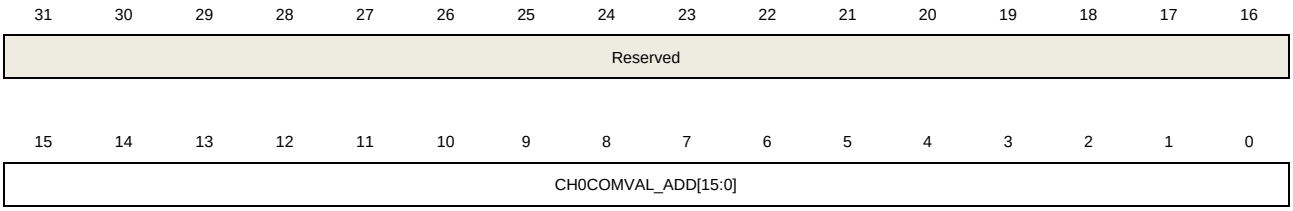
		0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled. 1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
10	IOS	Idle mode “off-state” enable When POEN&CHPOENx(x=0...2) 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 12-4. 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_CCHP0 register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx / ISOxN bits in TIMERx_CTL1 register, the BRKEN /BRKP/OAEN/DTCFG bits in TIMERx_CCHP0 register, are writing protected. 10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode), the ROS/IOS bits in TIMERx_CCHP0 register are writing protected. 11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL /CHxCOMSEN/CHxCOMADDSEN bits in TIMERx_CHCTL0/1 and registers (if the related channel is configured in output) are writing protected. This bit-field can be written only once after the system reset. Once the TIMERx_CCHP0 register has been written, this bit-field will be writing protected.
7:0	DTCFG[7:0]	Dead time configuration This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between the value of DTCFG and the duration of dead-time is as follow: DTCFG[7:5] = 3’b0xx: DT value = DTCFG[7:0] * t_{DT}, t_{DT} = t_{DTS}. DTCFG[7:5] = 3’b10x: DT value = (64+DTCFG[5:0]) * t_{DT}, t_{DT} = t_{DTS}*2. DTCFG[7:5] = 3’b110: DT value = (32+DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS}*8. DTCFG[7:5] = 3’b111: DT value = (32+DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS}*16. This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Channel 0 additional compare value register (TIMERx_CH0COMV_ADD)

Address offset: 0x64

Reset value: 0x0000 0000

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



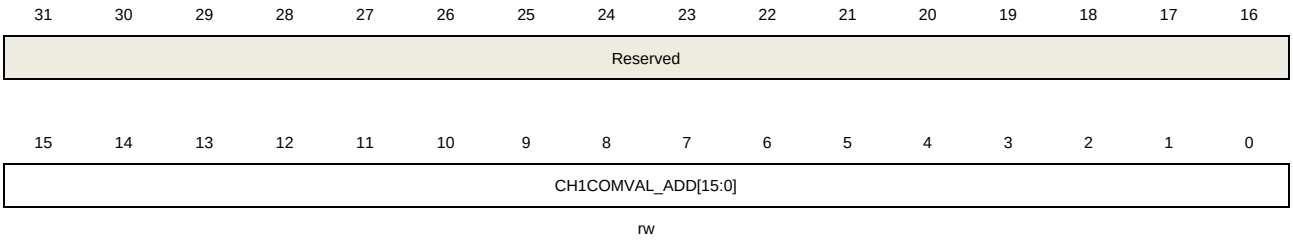
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0COMVAL_ADD [15:0]	Additional compare value of channel 0 When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode (when CH0CPWMEN=1).

Channel 1 additional compare value register (TIMERx_CH1COMV_ADD)

Address offset: 0x68

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode (when CH1CPWMEN=1).

Channel 2 additional compare value register (TIMERx_CH2COMV_ADD)

Address offset: 0x6C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2COMVAL_ADD [15:0]	Additional compare value of channel 2 When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode (when CH2CPWMEN=1).

Channel 3 additional compare value register (TIMERx_CH3COMV_ADD)

Address offset: 0x70

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3COMVAL_ADD [15:0]	Additional compare value of channel 3 When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode (when CH3CPWMEN=1).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0000 00FF

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3C	CH2C	CH1C	CH0C	Reserved											

PWMEN	PWMEN	PWMEN	PWMEN												
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3OMPSEL[1:0]		CH2OMPSEL[1:0]		CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		BRKEN CH3	BRKEN CH2	BRKEN CH1	BRKEN CH0	DTIEN CH3	DTIEN CH2	DTIEN CH1	DTIEN CH0
rw		rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	CH3CPWMEN	Channel 3 composite PWM mode enable 0: Disabled 1: Enabled
30	CH2CPWMEN	Channel 2 composite PWM mode enable 0: Disabled 1: Enabled
29	CH1CPWMEN	Channel 1 composite PWM mode enable 0: Disabled 1: Enabled
28	CH0CPWMEN	Channel 0 composite PWM mode enable 0: Disabled 1: Enabled
27:16	Reserved	Must be kept at reset value.
15:14	CH3OMPSEL[1:0]	Channel 3 output match pulse select When the match events occur, this bit is used to select the output of O3CPRE which drives CH3_O. 00: The O3CPRE signal is output normally with the configuration of CH3COMCTL[2:0] bits. 01: Only the counter is counting up, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle. 10: Only the counter is counting down, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle. 11: Both the counter is counting up and counting down, the O3CPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
13:12	CH2OMPSEL[1:0]	Channel 2 output match pulse select When the match events occur, this bit is used to select the output of O2CPRE which drives CH2_O. 00: The O2CPRE signal is output normal with the configuration of CH2COMCTL[2:0] bits. 01: Only when the counter is counting up, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O2CPRE signal is output a pulse

		when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O2CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
11:10	CH1OMPSEL[1:0]	Channel 1 output match pulse select When the match events occur, this bit is used to select the output of O1CPRE which drives CH1_O. 00: The O1CPRE signal is output normal with the configuration of CH1COMCTL[2:0] bits. 01: Only when the counter is counting up, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
9:8	CH0OMPSEL[1:0]	Channel 0 output match pulse select When the match events occur, this bit is used to select the output of O0CPRE which drives CH0_O. 00: The O0CPRE signal is output normal with the configuration of CH0COMCTL[2:0] bits. 01: Only when the counter is counting up, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
7	BRKENCH3	Break control enable for channel 3 0: Disabled 1: Enabled
6	BRKENCH2	Break control enable for channel 2 0: Disabled 1: Enabled
5	BRKENCH1	Break control enable for channel 1 0: Disabled 1: Enabled
4	BRKENCH0	Break control enable for channel 0 0: Disabled 1: Enabled
3	DTIENCH3	Dead time inserted enable for channel 3

		Enables the deadtime insertion in the outputs of CH3_ON and CH3_O. 0: Disabled 1: Enabled
2	DTIENCH2	Dead time inserted enable for channel 2 Enables the deadtime insertion in the outputs of CH2_ON and CH2_O. 0: Disabled 1: Enabled
1	DTIENCH1	Dead time inserted enable for channel 1 Enables the deadtime insertion in the outputs of CH1_ON and CH1_O. 0: Disabled 1: Enabled
0	DTIENCH0	Dead time inserted enable for channel 0 Enables the deadtime insertion in the outputs of CH0_ON and CH0_O. 0: Disabled 1: Enabled

Alternate function control register 0 (TIMERx_AFCTL0)

Address offset: 0x8C

Reset value: 0x0000 0001

This register 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						BRKCMP0P	Reserved								BRKIN0P
						rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						BRKCMP0EN	Reserved								BRKIN0EN
						rw									rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	BRKCMP0P	BREAK CMP0 input polarity This bit is used to configure the CMP0 input polarity, and the specific polarity is determined by this bit and the BRKP bit. 0: CMP0 input signal will not be inverted (BRKP =0, the input signal is active low; BRKP =1, the input signal is active high) 1: CMP0 input signal will be inverted (BRKP=0, the input signal is active high; BRKP =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

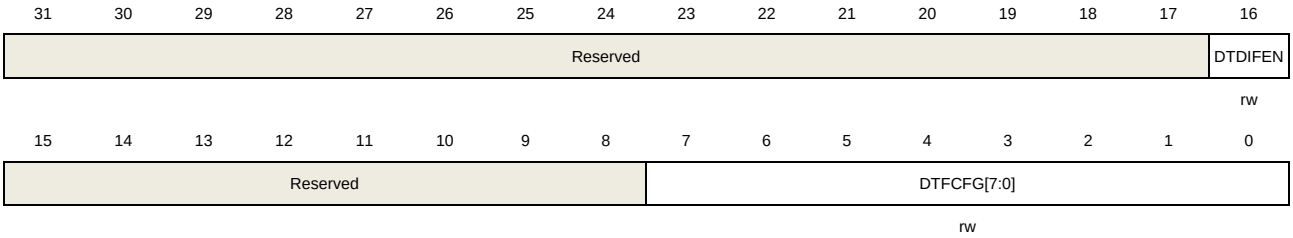
24:17	Reserved	Must be kept at reset value.
16	BRKIN0P	BREAK BRKIN0 alternate function input polarity This bit is used to configure the BRKIN0 input polarity, and the specific polarity is determined by this bit and the BRKP bit. 0: BRKIN0 input signal will not be inverted (BRKP =0, the input signal is active low; BRKP =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRKP=0, the input signal is active high; BRKP =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
15:10	Reserved	Must be kept at reset value.
9	BRKCMP0EN	BREAK CMP0 enable 0: CMP0 input disabled 1: CMP0 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
8:1	Reserved	Must be kept at reset value.
0	BRKIN0EN	BREAK BRKIN0 alternate function input enable 0: BRKIN0 alternate function input disabled 1: BRKIN0 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Complementary channel protection register 1 (TIMERx_CCHP1)

Address offset: 0x09C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	DTDIFEN	Dead time configure different enable 0: The dead time for both rising and falling edges are same, which is defined in DTCFG[7:0] bit-field in TIMERx_CCHP0 register or register. 1: The dead time on rising edge is defined in DTCFG[7:0] bit-field in

		TIMERx_CCHP0 register, and the dead time on falling edge is defined in DTFCFG [7:0] bit-field in TIMERx_CCHP1 register. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.
15:8	Reserved	Must be kept at reset value.
7:0	DTFCFG[7:0]	Dead time falling edge configure This bit-field controls the value of the dead-time on the falling edge of OxCPRE, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3'b0xx: DTvalue = DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b10x: DTvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b110: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b111: DTvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 00.

Counter initial control register (TIMERx_CINITCTL)

Address offset: 0xA4

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													SWSYNCG	CINITDIR	CINITVEN
													w	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	SWSYNCG	Soft synchronization event generation This bit is set by software and cleared by hardware automatically. Reads this bit always return a 0. 0: No affect 1: Generate soft synchronization event
1	CINITDIR	Counter initial count direction 0: When the synchronization event occurs, the counter count up. 1: When the synchronization event occurs, the counter count down. This bit indicates the initial direction of the counter after a synchronization event occurs and the counter initial value is loaded from the TIMERx_CINITV register.

Note: This bit is only used when the CAM[1:0] ≠ 00.

0	CINITVEN	Counter initial value register enable 0: Counter initial value register disable. The counter register can't load from the counter initial value register. 1: Counter initial value register enable. When a synchronization event (or soft synchronization event generated by setting the SWSYNCG bit) occurs, the counter register can load from the counter initial value register.

Counter initial value register (TIMERx_CINITV)

Address offset: 0xA8

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
CINITVAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CINITVAL [15:0]	Counter initial value This bits-field indicates the counter initial value. When the CINITVEN bit is 0, the counter register can't load the initial value which are set in CINITVAL bits-field. When the CINITVEN bit is 1, when the synchronization event occurs, the counter register can load the initial value which are set in CINITVAL bits-field.

Channel break control register (TIMERx_CHBRKCTL)

Address offset: 0xAC

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									CHPOEN2	CHPOEN1	CHPOEN0	Reserved			
									rw	rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH2BRKINEN	CH1BRKINEN	CH0BRKINEN	Reserved	CH2BRKIE	CH1BRKIE	CH0BRKIE	Reserved	CH2BRKINP	CH1BRKINP	CH0BRKINP	Reserved	CH2BRKMIE	CH1BRKMIE	CH0BRKMIE
	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	CHPOEN2	Channel primary output enable 2 It is cleared asynchronously by hardware as soon as channel 2 BREAK input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH2_O and CH2_ON) if the corresponding enable bits (CH2EN, CH2NEN) have been set. 0: CH2_O / CH2_ON outputs are disabled or forced to idle state. 1: CH2_O / CH2_ON outputs are enabled.
21	CHPOEN1	Channel primary output enable 1 It is cleared asynchronously by hardware as soon as channel 1 BREAK input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH1_O and CH1_ON) if the corresponding enable bits (CH1EN, CH1NEN) have been set. 0: CH1_O / CH1_ON outputs are disabled or forced to idle state. 1: CH1_O / CH1_ON outputs are enabled.
20	CHPOEN0	Channel primary output enable 0 It is cleared asynchronously by hardware as soon as channel 0 BREAK input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CH0_O and CH0_ON) if the corresponding enable bits (CH0EN, CH0NEN) have been set. 0: CH0_O / CH0_ON outputs are disabled or forced to idle state. 1: CH0_O / CH0_ON outputs are enabled.
19:15	Reserved	Must be kept at reset value.
14	CH2BRKINEN	Channel 2 BREAK input signal enable This bit can be set to enable the channel 2 BREAK input signal. 0: Channel 2 BREAK input disabled 1: Channel 2 BREAK input enabled
13	CH1BRKINEN	Channel 1 BREAK input signal enable This bit can be set to enable the channel 1 BREAK input signal. 0: Channel 1 BREAK input disabled 1: Channel 1 BREAK input enabled
12	CH0BRKINEN	Channel 0 BREAK input signal enable This bit can be set to enable the channel 0 BREAK input signal. 0: Channel 0 BREAK input disabled 1: Channel 0 BREAK input enabled
11	Reserved	Must be kept at reset value.
10	CH2BRKIE	Channel 2 BREAK interrupt enable 0: Disabled

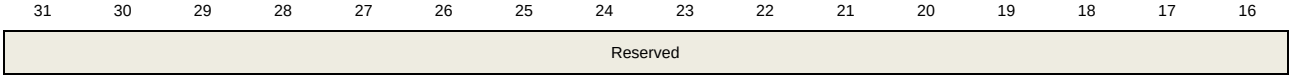
		1: Enabled
9	CH1BRKIE	Channel 1 BREAK interrupt enable 0: Disabled 1: Enabled
8	CH0BRKIE	Channel 0 BREAK interrupt enable 0: Disabled 1: Enabled
7	Reserved	Must be kept at reset value.
6	CH2BRKINP	Channel 2 BREAK input signal polarity This bit specifies the polarity of the channel 2 BREAK input signal. 0: Channel 2 BREAK input signal will not be inverted 1: Channel 2 BREAK input signal will be inverted
5	CH1BRKINP	Channel 1 BREAK input signal polarity This bit specifies the polarity of the channel 1 BREAK input signal. 0: Channel 1 BREAK input signal will not be inverted 1: Channel 1 BREAK input signal will be inverted
4	CH0BRKINP	Channel 0 BREAK input signal polarity This bit specifies the polarity of the channel 0 BREAK input signal. 0: Channel 0 BREAK input signal will not be inverted 1: Channel 0 BREAK input signal will be inverted
3	Reserved	Must be kept at reset value.
2	CH2BRKMIE	Channel 2 BREAK multi-period interrupt enable 0: Disabled 1: Enabled
1	CH1BRKMIE	Channel 1 BREAK multi-period interrupt enable 0: Disabled 1: Enabled
0	CH0BRKMIE	Channel 0 BREAK multi-period interrupt enable 0: Disabled 1: Enabled

Channel break period register (TIMERx_CHBRKPER)

Address offset: 0xB0

Reset value: 0x0000 0000

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



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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PERCNT[15:0]	Channel BREAK period count In Edge-aligned, this field represents PERCNT PWM periods. In Center-aligned, this field represents PERCNT/2 PWM periods.

Channel break interrupt flag register (TIMERx_CHBRKINTF)

Address offset: 0xB4

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									CH2BRKIF	CH1BRKIF	CH0BRKIF	Reserved	CH2BRKMIF	CH1BRKMIF	CH0BRKMIF
									rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	CH2BRKIF	Channel 2 BREAK interrupt flag This flag is set by hardware as soon as the channel 2 BREAK input is active, and cleared by software if the channel 2 BREAK input is not at active level. 0: No active level on channel 2 BREAK inputs has been detected. 1: An active level on channel 2 BREAK inputs has been detected. **Note:** When CH2BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
5	CH1BRKIF	Channel 1 BREAK interrupt flag This flag is set by hardware as soon as the channel 1 BREAK input is active, and cleared by software if the channel 1 BREAK input is not at active level. 0: No active level on channel 1 BREAK inputs has been detected. 1: An active level on channel 1 BREAK inputs has been detected. **Note:** When CH1BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
4	CH0BRKIF	Channel 0 BREAK interrupt flag This flag is set by hardware as soon as the channel 0 BREAK input is active, and cleared by software if the channel 0 BREAK input is not at active level.

0: No active level on channel 0 BREAK inputs has been detected.
1: An active level on channel 0 BREAK inputs has been detected.
Note: When CH0BRKIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.

3	Reserved	Must be kept at reset value.
2	CH2BRKMIF	Channel 2 BREAK multi-period interrupt flag This flag is set by hardware when the channel 2 BREAK input is active for multiple consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 2 BREAK input is not at active level. 0: No active level on channel 2 BREAK inputs for multiple consecutive periods has been detected. 1: An active level on channel 2 BREAK inputs for multiple consecutive periods has been detected. **Note:** When CH2BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
1	CH1BRKMIF	Channel 1 BREAK multi-period interrupt flag This flag is set by hardware when the channel 1 BREAK input is active for multiple consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 1 BREAK input is not at active level. 0: No active level on channel 1 BREAK inputs for multiple consecutive periods has been detected. 1: An active level on channel 1 BREAK inputs for multiple consecutive periods has been detected. **Note:** When CH1BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.
0	CH0BRKMIF	Channel 0 BREAK multi-period interrupt flag This flag is set by hardware when the channel 0 BREAK input is active for multiple consecutive periods (determined by the TIMERx_CHBRKPER register), and cleared by software if the channel 0 BREAK input is not at active level. 0: No active level on channel 0 BREAK inputs for multiple consecutive periods has been detected. 1: An active level on channel 0 BREAK inputs for multiple consecutive periods has been detected. **Note:** When CH0BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0
Reset value: 0x0000 0000

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

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

Reserved

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6’b000000: transfer 1 time 6’b000001: transfer 2 times … 6'b111111: transfer 63 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 6’b000000: TIMERx_CTL0 6’b000001: TIMERx_CTL1 … In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will

be accessed.

The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											OCPRECIVM	CCUSEL	Reserved	CHVSEL	OUTSEL
											rw	rw		rw	rw

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	OCPRECIVM	Output prepare signal clear invalid mode select This bit determines when the output prepare signal clear is set to invalid by the hardware. 0：The next counter overflow or underflow event 1：The next counter update event
3	CCUSEL	Commutation control shadow register update select This bit is valid only when the CCUC[2:0] bit-field are set to 100, 101 and 110. 0: The shadow registers update when the counter generates an overflow/ underflow event. 1: The shadow registers update when the counter generates an overflow/ underflow event and the repetition counter value is zero.
2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	OUTSEL	The output value selection bit This bit-field is set and reset by software. 1: If POEN bit and IOS bit are 0, the output is disabled. 0: No effect.

12.2. General level0 timer (TIMERx, x=1,2,3,4)

12.2.1. Overview

The general level0 timer module (TIMER1/2/3/4) is a four-channel timer that supports input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level0 timer has a 16-bit or 32-bit counter that can be used as an unsigned counter.

In addition, the general level0 timers can be programmed and be used for counting, their external events can be used to drive other timers.

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

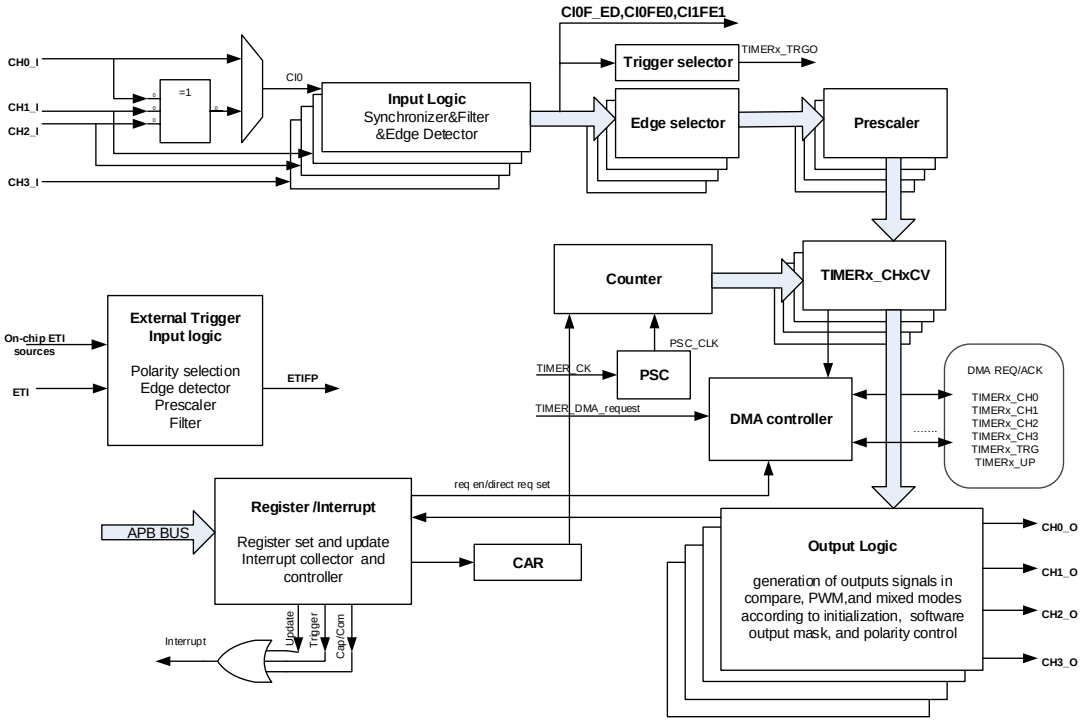
12.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits (TIMER2/3/4) or 32 bits (TIMER1).
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode and single pulse mode.
- Auto reload function.
- Interrupt output or DMA request: update event, trigger event and compare/capture event.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

12.2.3. Block diagram

Figure 12-48. General Level 0 timer block diagram provides details on the internal configuration of the general level 0 timer.

Figure 12-48. General Level 0 timer block diagram



12.2.4. Function overview

Clock selection

The general level0 TIMER has the capability of being clocked by either the CK_TIMER or an alternate clock source controlled by TSCFGy[4:0] (y=0..6,9..15) in SYSCFG_TIMERxCFG (x=1..4) registers.

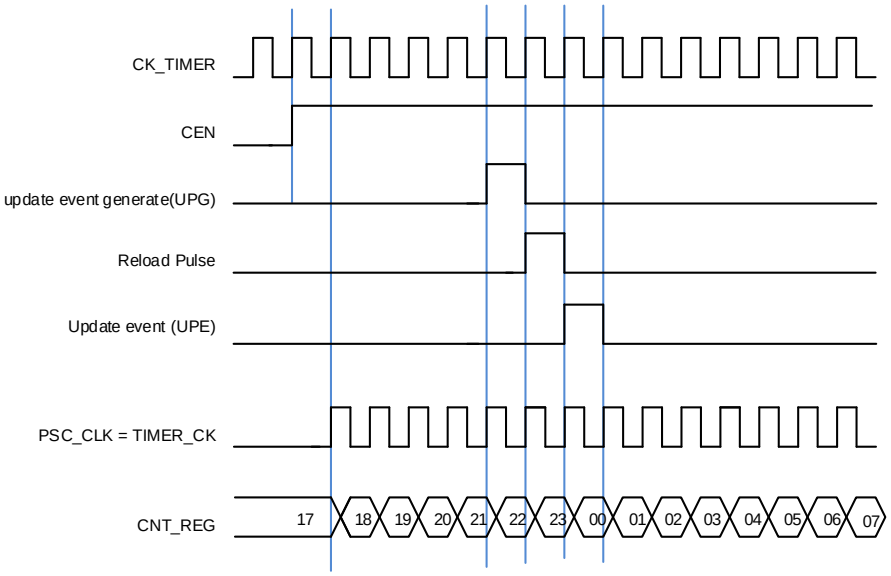
- TSCFGy[4:0] (y=0..6,9..15) in SYSCFG_TIMERxCFG(x=1..4) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=0..6,9..15) = 5'b00000 in SYSCFG_TIMERxCFG(x=1..4) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter's prescaler to count is equal to CK_TIMER which is from RCU module.

If TSCFGy[4:0] (y=0..2,6,9..14) in SYSCFG_TIMERxCFG(x=1..4) registers are setting to an available value, the prescaler is clocked by other clock sources selected in the TSCFGy[4:0] (y=0..2,6, 9..14) bit-filed, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5) are setting to an available value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 12-49. Normal mode, internal clock divided by 1



- TSCFG6[4:0] are setting to a nonzero value (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 TSCFG6[4:0] to 0x5~0x7 .

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0~ITI3. This mode can be selected by setting TSCFG6[4:0] to 0x1~0x4.

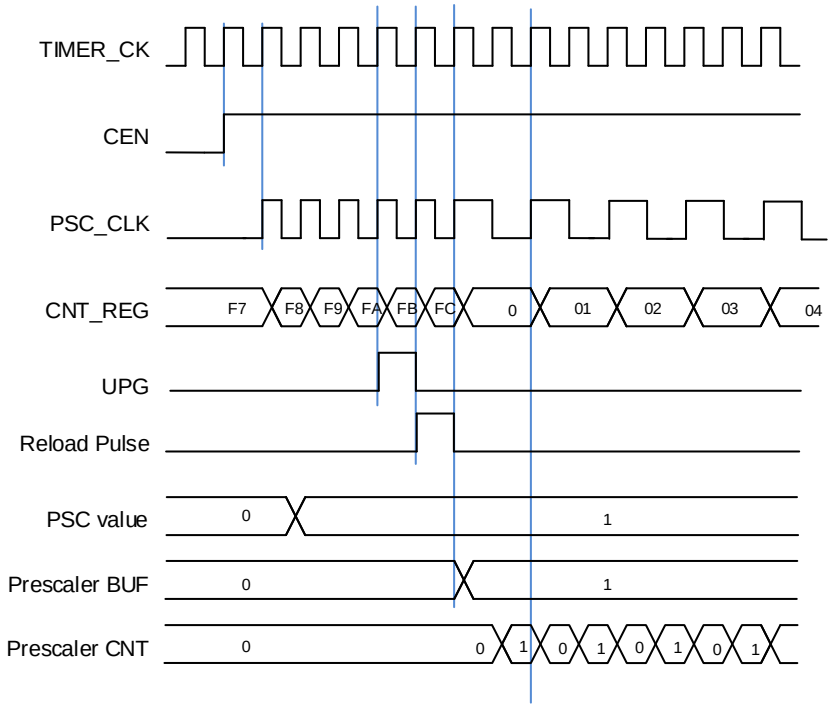
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is setting the TSCFG6[4:0] to 0x8. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Clock prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed ongoing, but it is adopted at the next update event.

Figure 12-50. Counter timing diagram with prescaler division change from 1 to 2



Up counting mode

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 restarts from 0. The update event is generated each time when counter overflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

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

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

Figure 12-51. Timing chart of up counting mode, PSC=0/2 and *Figure 12-52. Timing chart of up counting, change TIMERx_CAR ongoing* show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 12-51. Timing chart of up counting mode, PSC=0/2

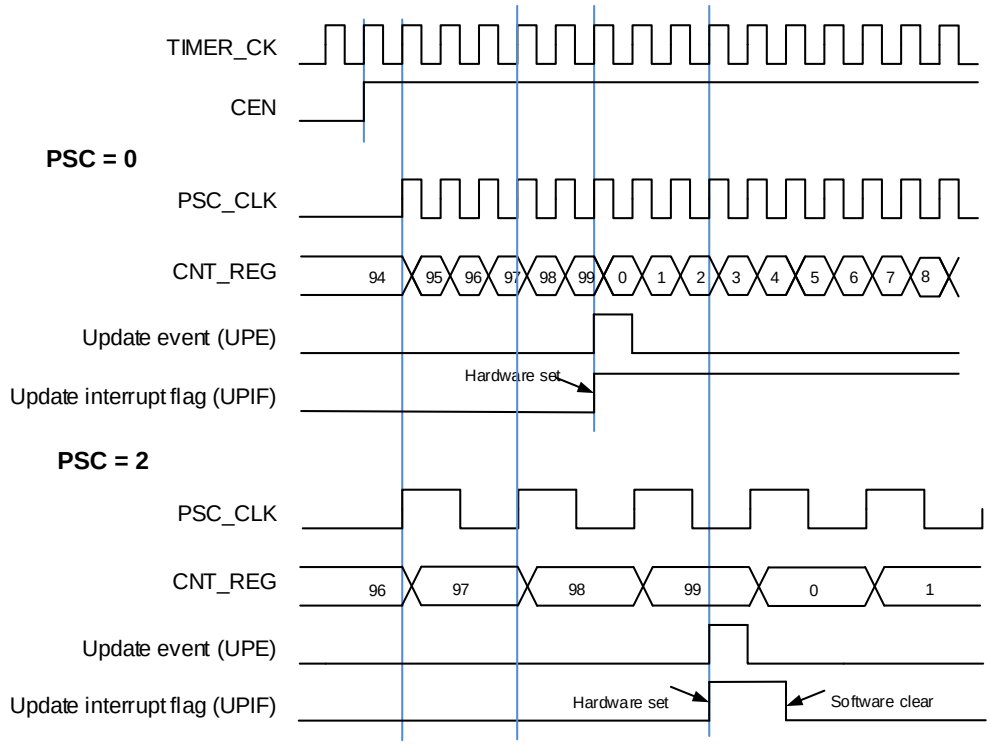
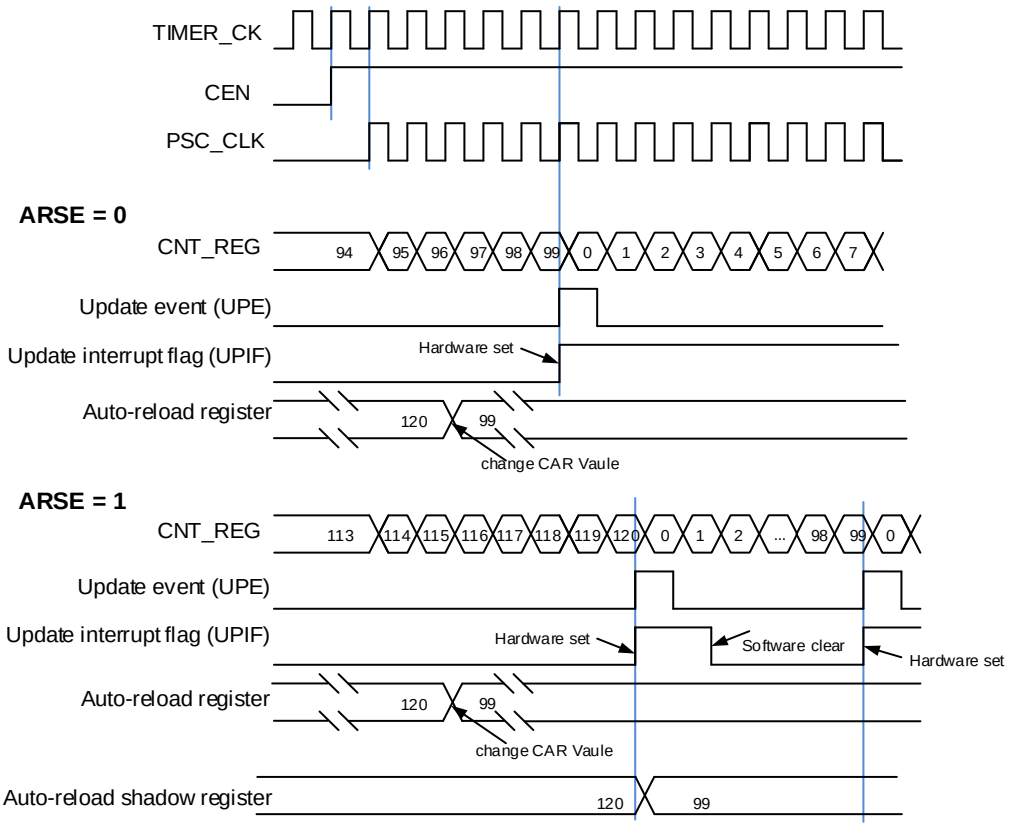


Figure 12-52. Timing chart of up counting, change TIMERx_CAR ongoing



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the TIMERx_CAR register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

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

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

Figure 12-53. Timing chart of down counting mode, PSC=0/2 and *Figure 12-54. Timing chart of down counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior for different clock frequencies when TIMERx_CAR = 0x99.

Figure 12-53. Timing chart of down counting mode, PSC=0/2

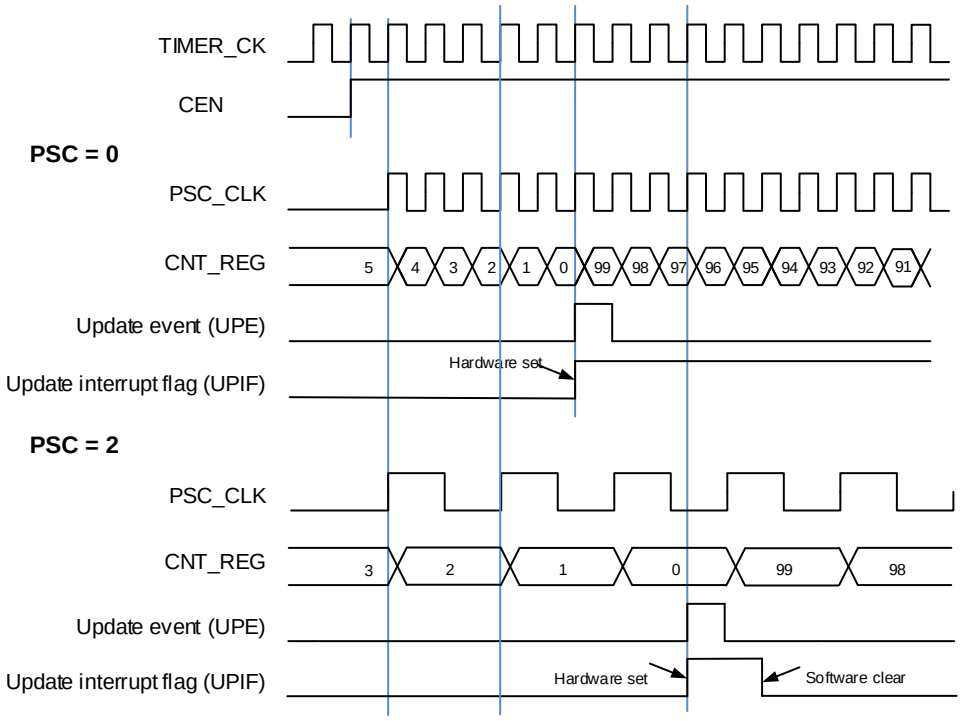
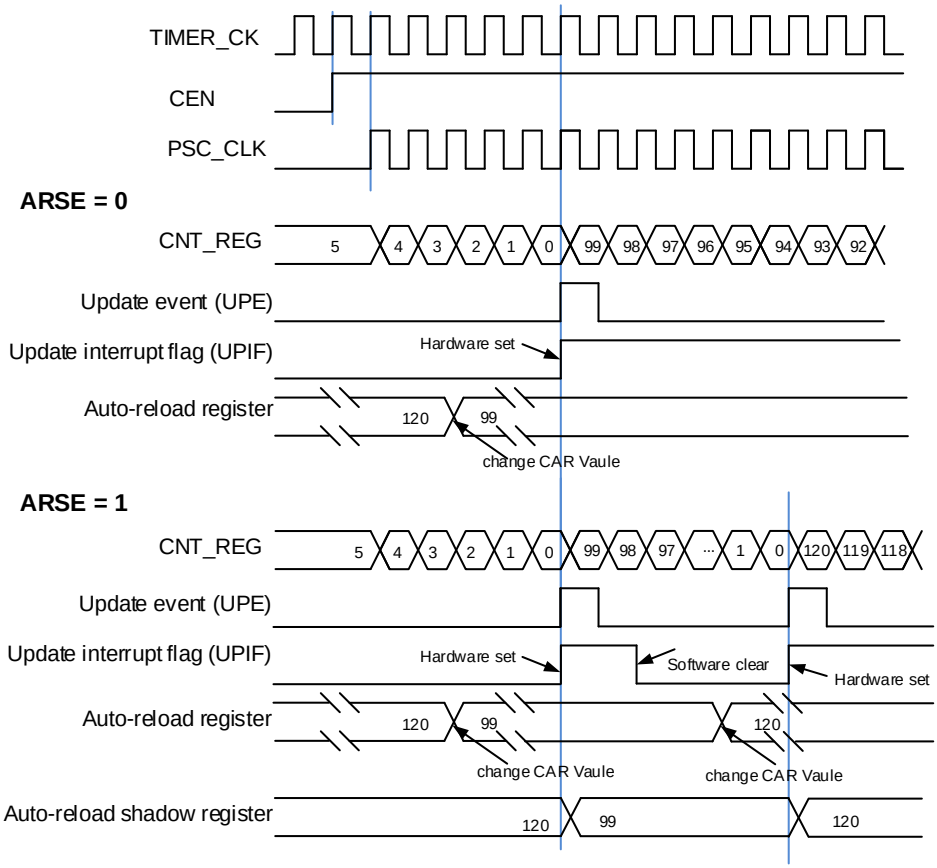


Figure 12-54. Timing chart of down counting mode, change TIMERx_CAR ongoing



Center-aligned counting mode

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 (TIMERx_CAR-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned counting mode. The counting direction is updated by hardware automatically.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

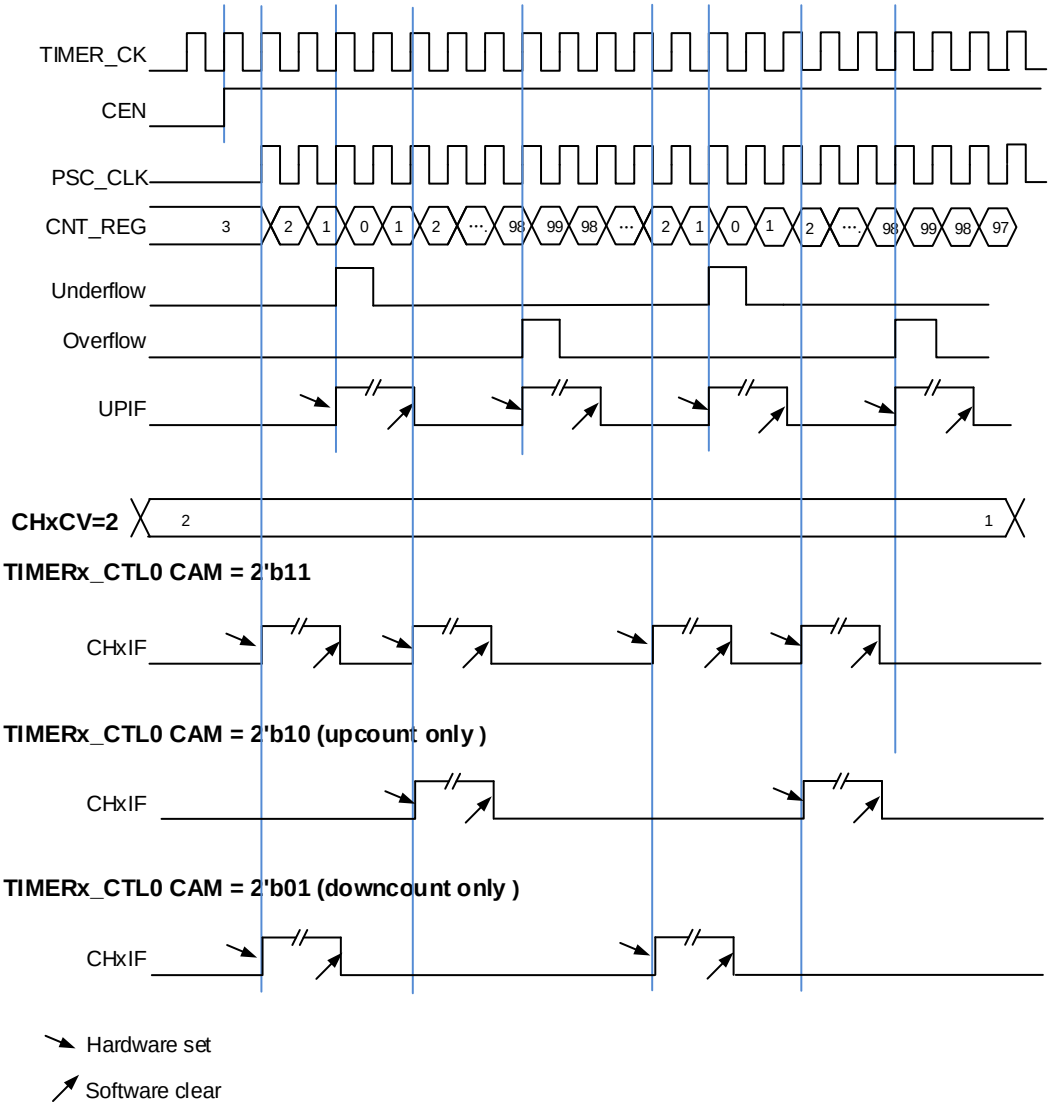
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 12-55. Timing chart of center-aligned counting mode***.

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

When an update event occurs, all the registers (auto-reload register, prescaler register) are updated. ***Figure 12-55. Timing chart of center-aligned counting mode*** shows the example

of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0.

Figure 12-55. Timing chart of center-aligned counting mode



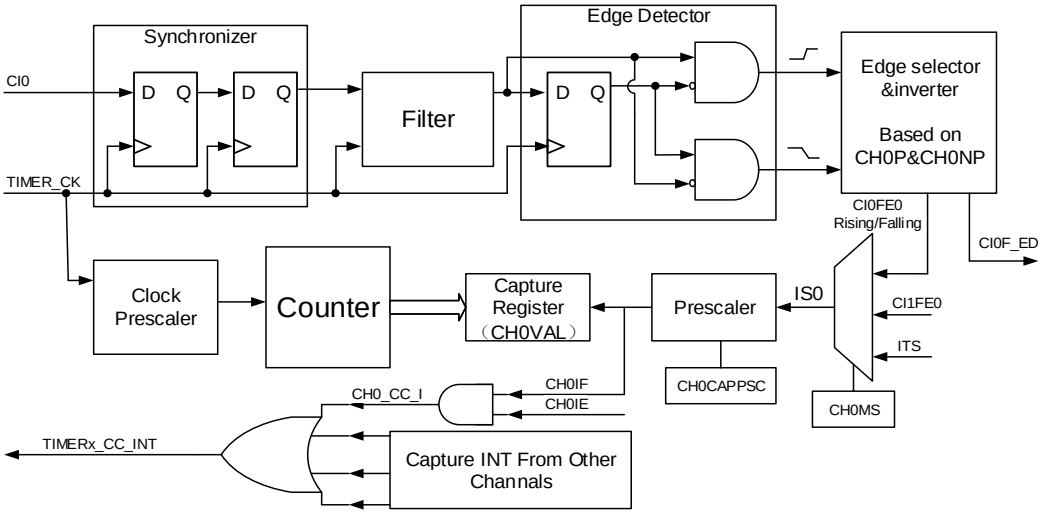
Capture/compare channels

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.

■ Input capture mode

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the CHxIF bit is set and the channel interrupt is generated if it is enabled when CHxIE=1.

Figure 12-56. Input capture logic



The input signals of channelx (CIx) can be the TIMERx_CHx signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals (just for CI0). First, the input signal of channel (CIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP bit. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT in TIMERx_CHCTL0).

Based on the input signal and quality of requested signal, configure compatible

CHxCAPFLT.

Step2: Edge selection (CHxP and CHxNP bits in TIMERx_CHCTL2).

Rising edge or falling edge, choose one by configuring CHxP and 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 input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 3'b001 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 3'b010 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

- **Output compare mode**

Figure 12-57. Output compare logic (x=0,1,2,3)

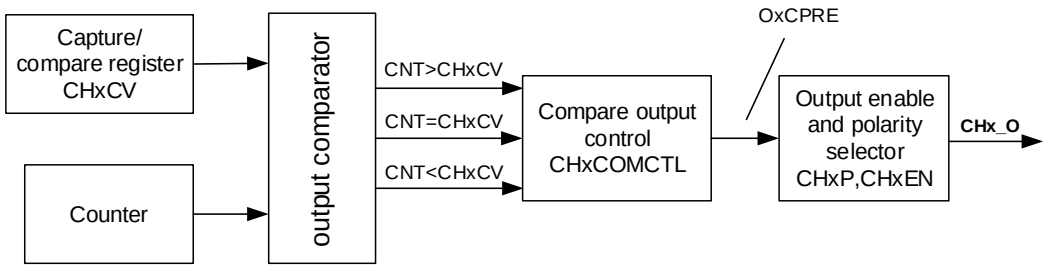


Figure 12-57. Output compare logic (x=0,1,2,3) shows the logic circuit of output compare mode. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output prepare signal*) is described as blew: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE signal, CHxP bit and CH0P bit (please refer to the TIMERx_CHCTL2 register for more details).For example, configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxEN=1 (the output of CHx_O is enabled),

If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV register, the CHxIF bit will be set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be asserted, if CHxDEN=1.

So, the process can be divided into several steps as below:

Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP.
- Enable the output by CHxEN.

Step3: Interrupt/DMA-request enables configuration by CHxIE/CHxDEN.

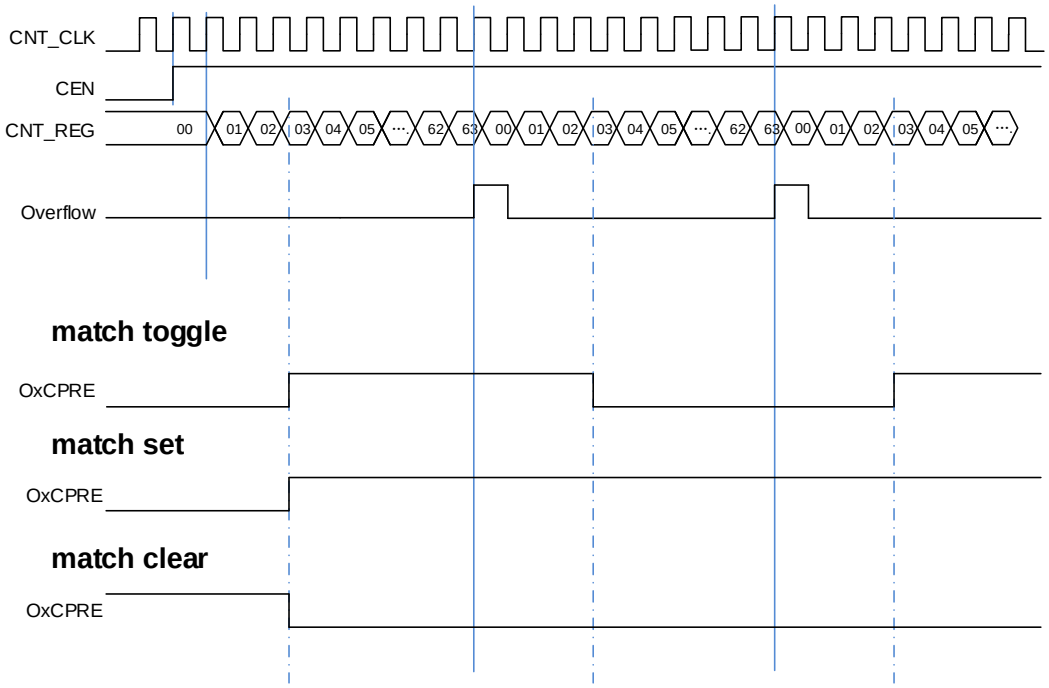
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.

The TIMERx_CHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 12-58. Output-compare under three modes shows the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 12-58. Output-compare under three modes



PWM mode

In the PWM output mode (by setting the CHxCOMCTL bit to 4'b0110 (PWM mode 0) or to 4'b0111 (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 12-59. 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 12-60. 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 counter, the output will be always active in PWM mode 0 (CHxCOMCTL=4'b0110). And if the value of TIMERx_CHxCV is greater than the value of counter, the output will be always inactive in PWM mode 1 (CHxCOMCTL=4'b0111).

Figure 12-59. Timing chart of EAPWM

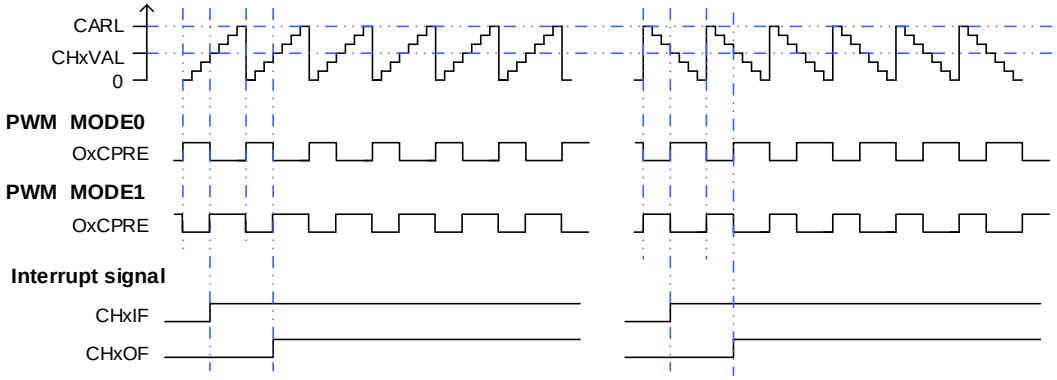
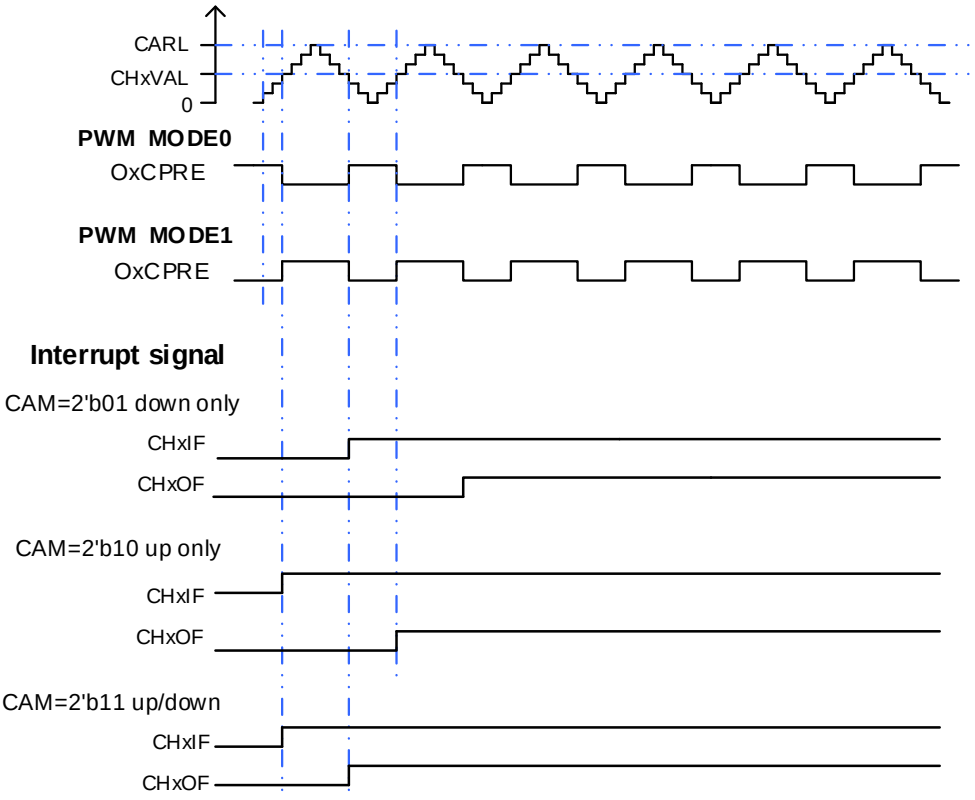


Figure 12-60. Timing chart of CAPWM



Channel output prepare signal

As is shown in ***Figure 12-57. Output compare logic (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. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit. 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.

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 with each other to generate the counter value.

Setting TSCFGy[4:0] (y=0, 1, 2, 13, 14) != 5'b00000 to select that the counting direction of timer is determined only by the CI0, only by the CI1, or by the CI0 and the CI1.

The DIR bit is modified by hardware automatically during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in ***Table 12-8. Counting direction in different quadrature decoder signals***. The CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection. The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 12-8. Counting direction in different quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 TSCFG0[4:0]!= 5'b00000	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 TSCFG1[4:0]!= 5'b00000	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 TSCFG2[4:0]!= 5'b00000	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up
Quadrature decoder mode 3 TSCFG13[4:0]!= 5'b00000	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	-	-	-	-
Quadrature decoder mode 4 TSCFG14[4:0]!= 5'b00000	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	-	-

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 12-61. Example of counter operation in decoder interface mode

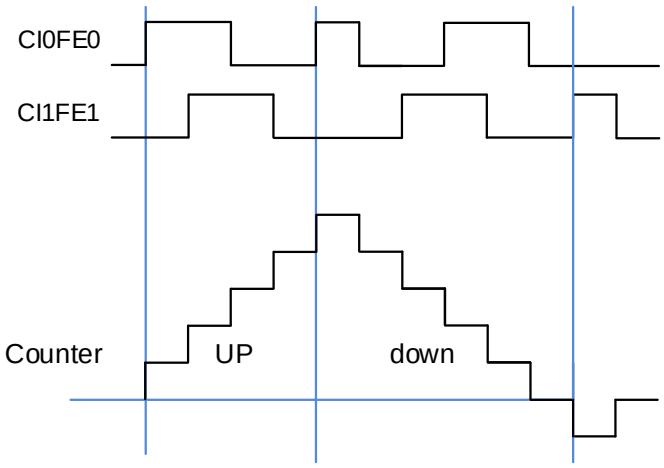
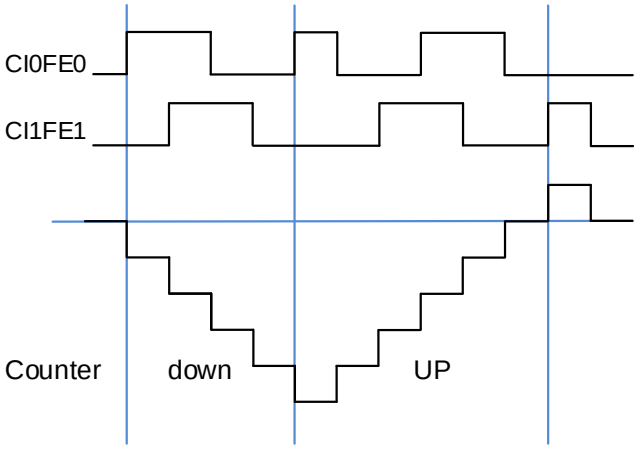


Figure 12-62. Example of decoder interface mode with CI0FE0 polarity inverted



When the counter direction changes in the quadrature decoder modes, the DIR bit in TIMERx_CTL0 register within a change, at the same time the DIRTRANIF bit in TIMERx_INTF is set to 1. And the corresponding interrupt is generated if the DIRTRANIE bit in TIMERx_DMAINTEN register is set to 1.

Quadrature decoder signal detection

The quadrature decoder signal jump detection function can be enabled by setting the DECJDEN bit (in TIMERx_CTL2 register) to 1, which can be used to detect whether the level jump edges (rising or falling) of the CI0 and CI1 signals occur at the same time.

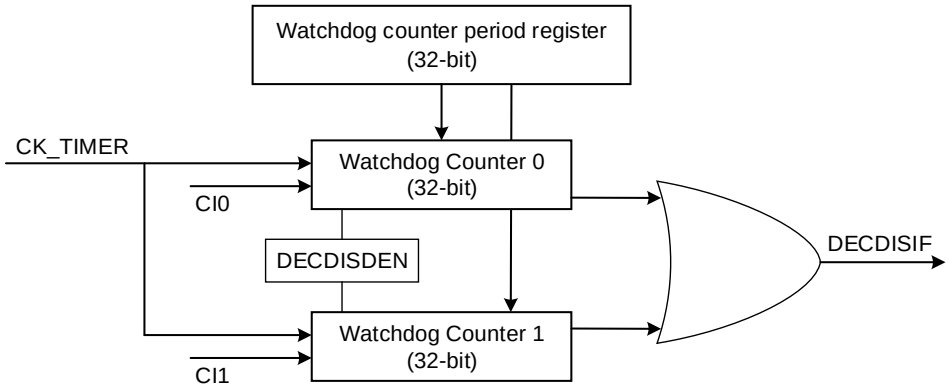
When DECJDEN=1, if the level transitions of the two quadrature signals CI0 and CI1 occur simultaneously, the interrupt flag DECJIF is set, if DECJIE=1, the corresponding interrupt is generated.

The quadrature decoder signal disconnection detection function can be enabled by setting the DECDISDEN bit (in TIMERx_CTL2 register) to 1, which can be used to detect the signal conditions of the CI0 and CI1 signals.

As shown in ***Figure 12-63. Quadrature decoder signal disconnection detection block diagram***. The signal detection module includes two 32-bit watchdog counters and a period register. The CI0 and CI1 signals are used to reset the two watchdog counters respectively.

When DECDISDEN=1, two watchdog counters start counting up at the same time. If the counter continues to count to the watchdog period value (this value is determined by the WDGPER[31:0] bit-field in the TIMERx_WDGPER register), the counter is timeout and the interrupt flag DECDISIF is set. If DECDISIE=1, the corresponding interrupt is generated.

Figure 12-63. Quadrature decoder signal disconnection detection block diagram



Non-quadrature decoder

The non-quadrature decoder function has four modes: non-quadrature decoder mode 0~3, which can be selected by setting TSCFGy[4:0] (y=9, 10, 11, 12) != 5'b00000.There are two input sources in these four modes: CI0 and CI1, and the CI0FE0 and CI1FE1 are the signals of the CI0 and CI1 after the filtering and polarity selection.

When the non-quadrature decoder mode 0 and non-quadrature decoder mode 1 are enabled,

the CI0FE0 is used as the count direction signal and the CI1FE1 signal is used as the count pulse.

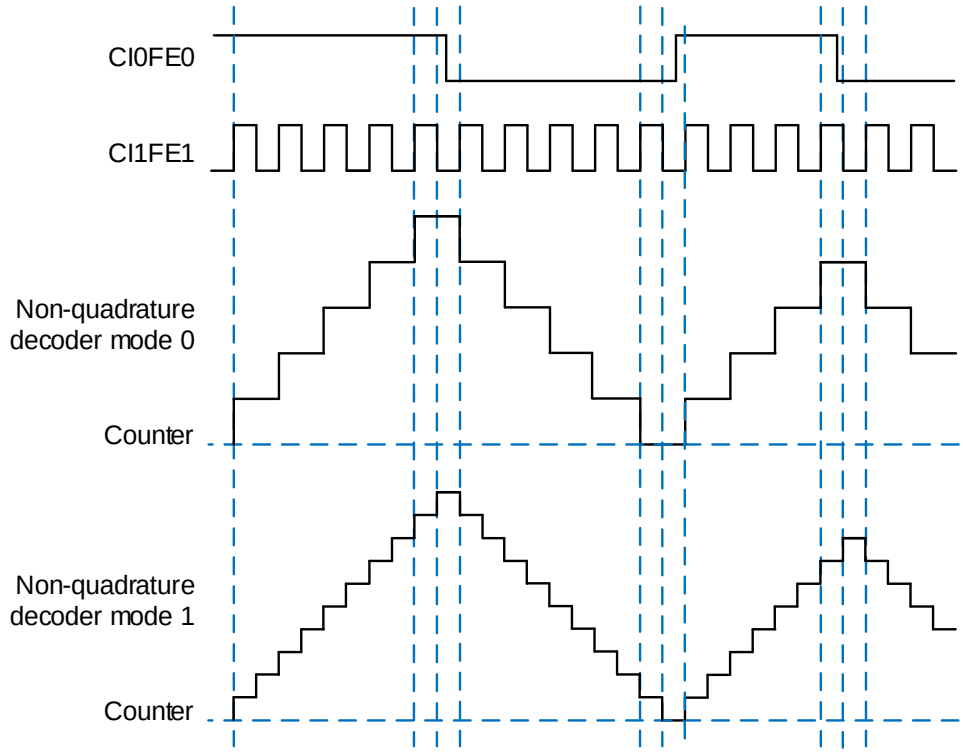
The CH0P is used to configure the count direction selection: When CH0P =0, the counter will count up when the CI0FE0 signal is high and will count down when the CI0FE0 signal is low; When CH0P =1, the counter will count down when the CI0FE0 signal is high and will count up when the CI0FE0 signal is low.

The CH1P is used to configure the count pulse edge selection: In non-quadrature decoder mode 0, the counter will count on both rising and falling edges of the CI1FE1 signal. In non-quadrature decoder mode 1, when CH1P=0, the counter will count on the rising edge of the CI1FE1 signal; When CH1P=1, the counter will count on the falling edge of the CI1FE1 signal. The more details for non-quadrature decoder mode 1 is shown in ***Table 12-9. the counter operation in non-quadrature mode 1*** and ***Figure 12-64. Example of counter operation in non-quadrature mode 0 / 1 with CH1P=0***.

Table 12-9. the counter operation in non-quadrature mode 1

CH1P	level	counter operation
0	CI0FE0 is high	the counter will count up on the rising edge of the CI1FE1 input signal
	CI0FE0 is low	the counter will count down on the rising edge of the CI1FE1 input signal
1	CI0FE0 is high	the counter will count up on the falling edge of the CI1FE1 input signal
	CI0FE0 is low	the counter will count down on the falling edge of the CI1FE1 input signal

Figure 12-64. Example of counter operation in non-quadrature mode 0 / 1 with CH1P=0



The non-quadrature decoder mode 2/3 uses two inputs CI0FE0 and CI1FE1 respectively to interact with each other to generate the counter value, and the counting direction DIR bit modified by hardware automatically.

When in non-quadrature decoder mode 2, the counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.

When in non-quadrature decoder mode 3, the counter will count on rising or falling edge of CI0FE0 and CI1FE1 signals. When CHxP=0, the counter will counter on the high level or the falling edge or the CIxFEx signal; When CHxP=1, the counter will counter on the low level or the rising edge or the CIxFEx signal. The more details for non-quadrature decoder mode 2/3 is shown in ***Figure 12-65. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0)*** and ***Table 12-10. the counter operation in non-quadrature decoder mode 2 / 3.***

Figure 12-65. Example of counter operation in non-quadrature decoder mode 2 / 3 (CH0P / CH1P=0)

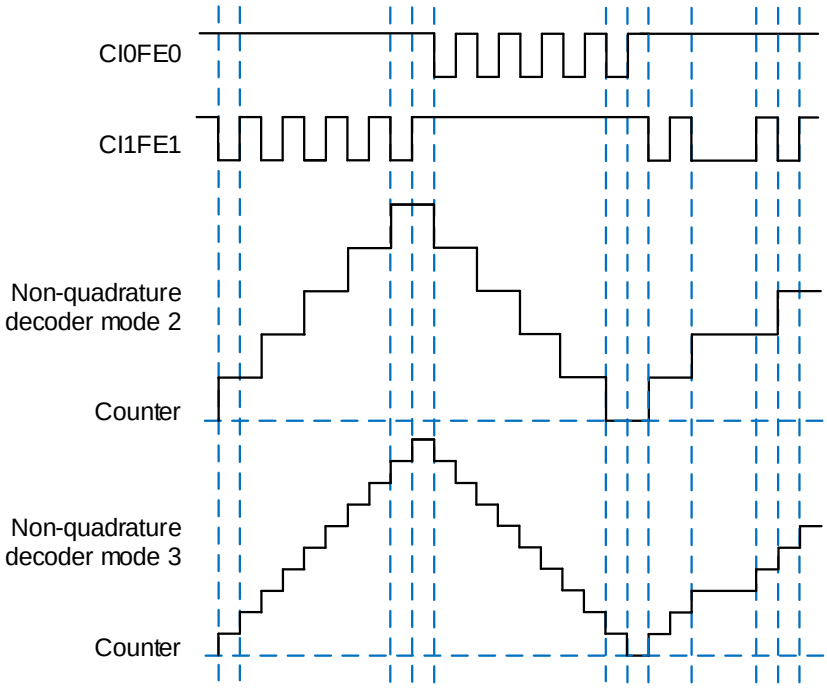


Table 12-10. the counter operation in non-quadrature decoder mode 2 / 3

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
non-quadrature decoder mode 2 TSCFG11[4:0] != 5'b00000	CHxP=0 (x=0 or 1)	CI1FE1=1	Down	Down	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	Up	Up
		CI0FE0=0	x	x	-	-
	CHxP=1 (x=0 or 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	Down	Down	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	Up	Up
non-quadrature decoder mode 3	CHxP=0 (x=0 or 1)	CI1FE1=1	-	Down	x	x
		CI1FE1=0	-	-	x	x

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
TSCFG12[4:0] != 5'b00000		CI0FE0=1	x	x	-	Up
		CI0FE0=0	x	x	-	-
	CHxP=1 (x=0 or 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	Down	-	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	Up	-

When the counter direction changes in the non-quadrature decoder modes, the DIR bit in TIMERx_CTL0 register within a change, at the same time the DIRTRANIF bit in TIMERx_INTF is set to 1. And the corresponding interrupt is generated if the DIRTRANIE bit in TIMERx_DMAINTEN register is set to 1.

Quadrature decoder and non-quadrature decoder clock output

TIMERs can output the decoder clock output signal as TRGO. This function can be enabled by setting the MMC[3:0] bit-field to 4'b1000 in the TIMERx_CTL1 register, and just used in quadrature decoder mode 0~4 and non-quadrature decoder mode 0~3.

Hall sensor function

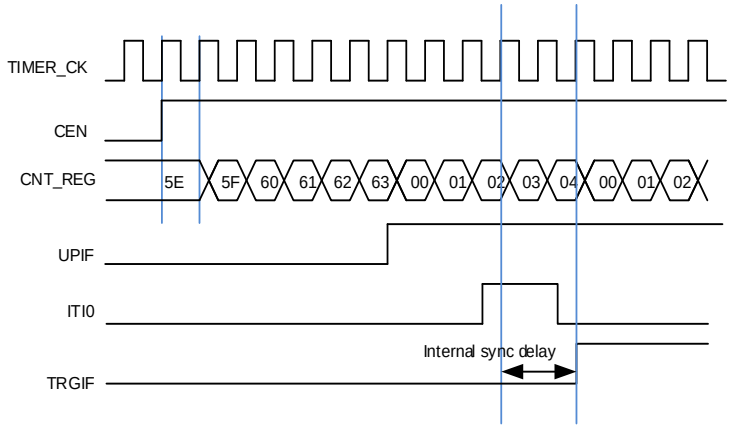
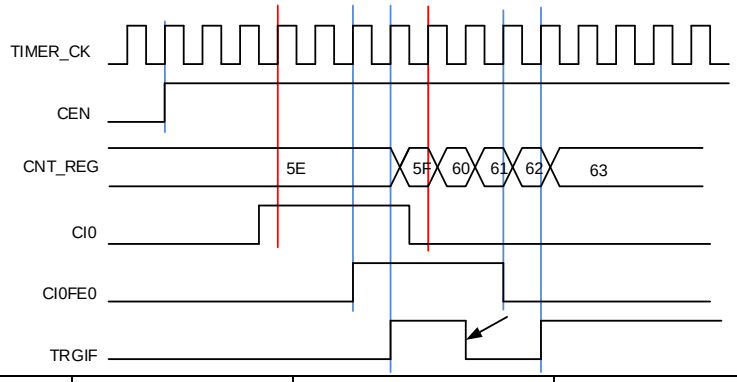
Refer to ***Advanced timer (TIMERx, x=0, 7)Hall sensor function***.

Master-slave management

The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode and so on, which is selected by the TSCFGy[4:0] (y=3..6) in SYSCFG_TIMERxCFG (x=1..4).

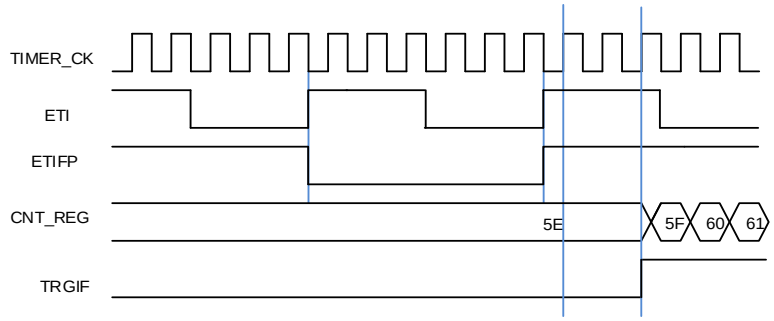
Table 12-11. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode y=4: pause mode y=5: event mode y=6: external clock mode 0	TSCFGy[4:0] 00000:Mode disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP(1) 01001: CI2FE2 01010: CI3FE3	If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion. If ETIFP (the filtered output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	For the ITIx, no filter and prescaler can be used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
	Restart mode	TSCFG3[4:0]	For ITI0, no polarity	For the ITI0, no filter

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
Exam1	The counter will be cleared and restart when a rising edge of trigger input comes.	5'b00001, ITI0 is selected.	selector can be used.	and prescaler can be used.
	Figure 12-66. 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.	TSCFG4[4:0] =5'b00110, 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 12-67. Pause mode  			
Exam3	**Event mode** The counter will start to count when a rising edge of trigger input comes.	TSCFG5[4:0] =5'b01000, 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.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler

Figure 12-68. Event mode



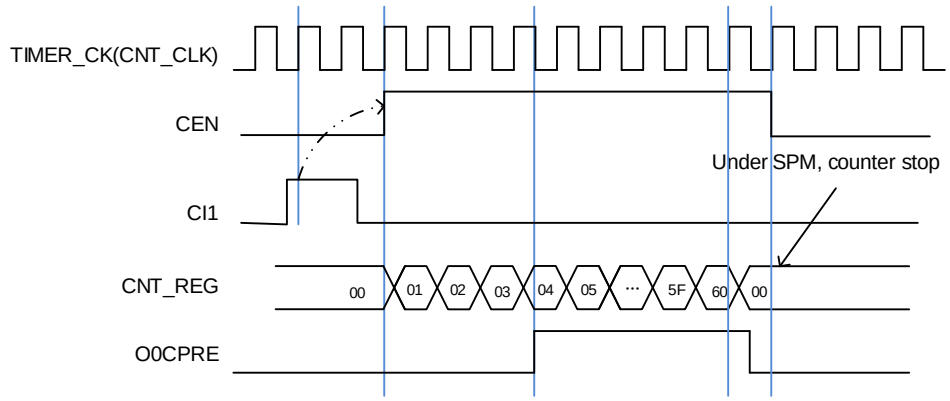
Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL bit.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the CEN bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signal will change to, as the compare match event occurs without taking the comparison result into account.

Figure 12-69. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60



Timers interconnection

Please refer to ***Advanced timer (TIMERx, x=0, 7)Timers interconnection***.

Timer DMA mode

Timer DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

12.2.5. Registers definition (TIMERx, x=1,2,3,4)

TIMER1 base address: 0x4000 0000

TIMER2 base address: 0x4000 0400

TIMER3 base address: 0x4000 0800

TIMER4 base address: 0x4000 0C00

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 ratio between CK_TIMER (the timer clock) and DTS (the dead time and sampling clock) which is used for the dead time generator and the digital filter. 00: f_{DTS}= f_{CK_TIMER} 01: f_{DTS}= f_{CK_TIMER} /2 10: f_{DTS}= f_{CK_TIMER} /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when the counter is counting down, compare interrupt flag of channels can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in

TIMERx_CHCTL0 register). Only when the counter is counting up, compare interrupt flag of channels can be set.

11: Center-aligned and counting up/down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Both when the counter is counting up and counting down, compare interrupt flag of channels can be set.

After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.

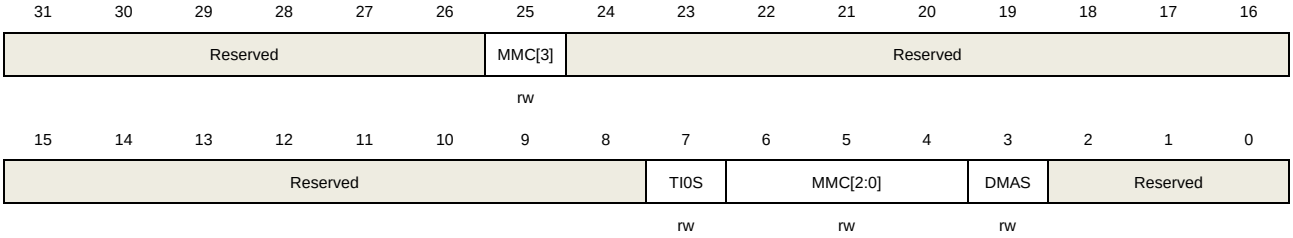
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode or decoder mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generates an update interrupt or a DMA request: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Only counter overflow/underflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	MMC[3]	Master mode control Refer to MMC[2:0] description.
24:8	Reserved	Must be kept at reset value.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 0000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 0001: Enable. This mode is used to start several timers at the same time or control a slave timer to be enabled in a period. In this mode, the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 0010: Update. In this mode, the master mode controller selects the update event as TRGO. 0011: Capture/compare pulse. In this mode, the master mode controller generates a TRGO pulse when a capture or a compare match occurs in channel 0. 0100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO. 0101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO. 0110: Compare. In this mode, the master mode controller selects the O2CPRE

		signal as TRGO. 0111: Compare. In this mode, the master mode controller selects the O3CPRE signal as TRGO. 1000: Decoder clock output. In this mode, the master mode controller selects the decoder clock signal as TRGO. This value just used in quadrature decoder mode 0~4 and non-quadrature decoder mode 0~3. 1001~1111: Reserved.
3	DMAS	DMA request source selection 0: DMA request of CHx is sent when capture/compare event occurs. 1: DMA request of channel CHx 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	Reserved						
rw	rw	rw		rw				rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal. 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of slave mode controller is used to enable external clock mode 1 In external clock mode 1, the counter is clocked by any active edge of the ETIFP signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled It is possible to simultaneously use external clock mode 1 with the restart mode, pause mode or event mode. But the TSCFGy[4:0](y=3,4,5) bits must not be 5'b01000 in this case. The external clock input will be ETIFP if external clock mode 0 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in TSCFG6[4:0] bit-field in

		SYSCFG_TIMERxCFG1 register.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETIFP must not be higher than 1/4 of TIMER_CK frequency. When the frequency of external trigger signal is high, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disabled 01: ETIFP frequency divided by 2 10: ETIFP frequency divided by 4 11: ETIFP frequency divided by 8
11:8	ETFC[3:0]	External trigger filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample ETIFP signal and the length of the digital filter applied to ETIFP. 0000: Filter disabled. f_{SAMP}= f_{DTS}, N=1. 0001: f_{SAMP}= f_{CK_TIMER}, N=2. 0010: f_{SAMP}= f_{CK_TIMER}, N=4. 0011: f_{SAMP}= f_{CK_TIMER}, N=8. 0100: f_{SAMP}=f_{DTS}/2, N=6. 0101: f_{SAMP}=f_{DTS}/2, N=8. 0110: f_{SAMP}=f_{DTS}/4, N=6. 0111: f_{SAMP}=f_{DTS}/4, N=8. 1000: f_{SAMP}=f_{DTS}/8, N=6. 1001: f_{SAMP}=f_{DTS}/8, N=8. 1010: f_{SAMP}=f_{DTS}/16, N=5. 1011: f_{SAMP}=f_{DTS}/16, N=6. 1100: f_{SAMP}=f_{DTS}/16, N=8. 1101: f_{SAMP}=f_{DTS}/32, N=5. 1110: f_{SAMP}=f_{DTS}/32, N=6. 1111: f_{SAMP}=f_{DTS}/32, N=8.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

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														DECDISIE	DECJIE
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	Reserved	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	Reserved	TRGIE	Reserved	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	DECDISIE	Quadrature decoder signal disconnection interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for quadrature decoder signal disconnection detection is enabled (when DECDISDEN =1).
16	DECJIE	Quadrature decoder signal jump (the two signals jump at the same time) interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for quadrature decoder signal jump detection is enabled (when DECJDEN =1).
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														DECDISIF	DECJIF
														rc_w0	rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CH3OF	CH2OF	CH1OF	CH0OF	Reserved		TRGIF	Reserved	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	DECDISIF	Quadrature decoder signal disconnection interrupt flag 0: No quadrature decoder signal disconnection interrupt occurred 1: Quadrature decoder signal disconnection interrupt occurred

		Note: This bit just used for quadrature decoder signal disconnection detection is enabled (when DECDISDEN =1).
16	DECJIF	Quadrature decoder signal jump (the two signals jump at the same time) interrupt flag 0: No quadrature decoder signal jump interrupt occurred 1: Quadrature decoder signal jump interrupt occurred **Note:** This bit just used for quadrature decoder signal jump detection is enabled (when DECJDEN =1).
15:13	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 by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	Reserved	Must be kept at reset value.
4	CH3IF	Channel 3 capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software.

		If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	Reserved	Must be kept at reset value.
4	CH3G	Channel 3 capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2 capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set,

		and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, while in down counting mode it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS[2]	CH0MS[2]	Reserved					CH1COMCTL[3]	Reserved							CH0COMCTL[3]
							Reserved								Reserved
rw	rw						rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	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	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29:25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control

		Refer to CH0COMCTL[2:0] description
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[2:0] 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. The CH1MS[2:0] bit-field is writable only when the channel is not active (the CH1EN bit in TIMERx_CHCTL2 register is reset). 000: Channel 1 is configured as output. 001: Channel 1 is configured as input, IS1 is connected to CI1FE1. 010: Channel 1 is configured as input, IS1 is connected to CI0FE1. 011: Channel 1 is configured as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2 (x=1...4) register). 100~111: Reserved.
7	CH0COMCEN	Channel 0 output compare clear enable When this bit is set, the O0CPRE signal is cleared when high level is detected on ETIFP input. 0: Channel 0 output compare clear disabled 1: Channel 0 output compare clear enabled
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[3] and CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O. The active level of O0CPRE is high, while the active level of CH0_O depends on CH0P bit. 0000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 0010: Clear the channel output on match. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 0011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 0100: Force low. O0CPRE is forced low level. 0101: Force high. O0CPRE is forced high level. 0110: PWM mode 0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV, otherwise it is inactive. When counting down,

		O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV, otherwise it is active. 0111: PWM mode 1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV, otherwise it is active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV, otherwise it is inactive. 1000~1111: Reserved. If configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from “Timing” mode to “PWM” mode or the result of the comparison changes.
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).
2	CH0COMFEN	Channel 0 output compare fast enable 0: Channel 0 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and CH0_O is set to the compare level.The minimum delay from an edge on the result of the comparison to activate CH0_O output is 5 clock cycles. 1: Channel 0 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and CH0_O is set to the compare level.The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (the CH0EN bit in TIMERx_CHCTL2 register is reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=1...4) register). 100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Same as output compare mode.

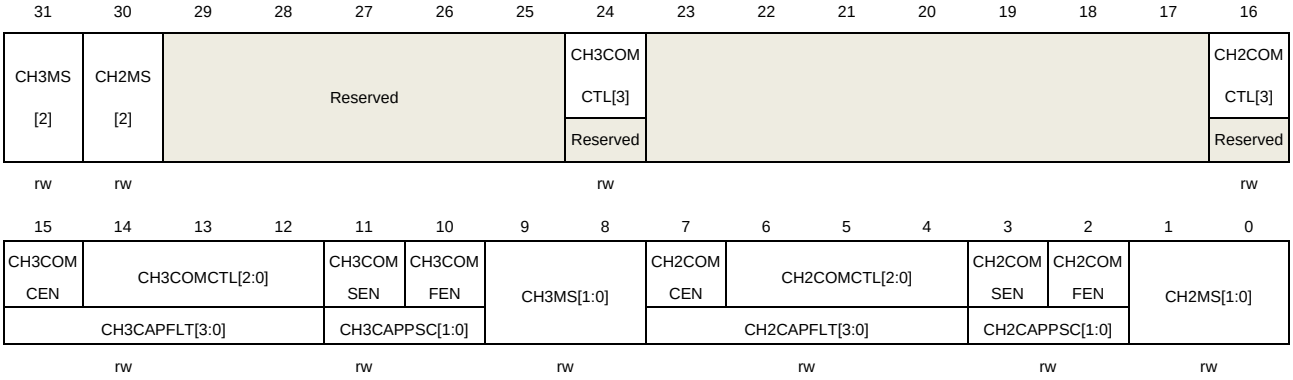
30	CH0MS[2]	Channel 0 I/O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8. 1010: $f_{SAMP}=f_{DTS}/16$, N=5. 1011: $f_{SAMP}=f_{DTS}/16$, N=6. 1100: $f_{SAMP}=f_{DTS}/16$, N=8. 1101: $f_{SAMP}=f_{DTS}/32$, N=5. 1110: $f_{SAMP}=f_{DTS}/32$, N=6. 1111: $f_{SAMP}=f_{DTS}/32$, N=8.
3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection Same as output compare mode.

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Refer to CH3MS[1:0]description.
30	CH2MS[2]	Channel 2 I/O mode selection Refer to CH2MS[1:0] description.
29:25	Reserved	Must be kept at reset value.
24	CH3COMCTL[3]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
23:17	Reserved	Must be kept at reset value.
16	CH2COMCTL[3]	Channel 2 compare output control Refer to CH2COMCTL[2:0] description
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH2COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH2COMCTL[2:0] description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH2COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH2COMFEN description.
9:8	CH3MS[1:0]	Channel 3 I/O mode selection This bit-field specifies the direction of the channel and the input signal selection. The CH3MS[2:0] bit-field is writable only when the channel is not active (the CH3EN bit in TIMERx_CHCTL2 register is reset).

		00: Channel 3 is configured as output. 01: Channel 3 is configured as input, IS3 is connected to CI3FE3. 10: Channel 3 is configured as input, IS3 is connected to CI2FE3. 11: Channel 3 is configured as input, IS3 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=1...4) register). 100~111: Reserved.
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, the O2CPRE signal is cleared when high level is detected on ETIFP input. 0: Channel 2 output compare clear disabled 1: Channel 2 output compare clear enabled
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field controls the behavior of O2CPRE which drives CH2_O. The active level of O2CPRE is high, while the active level of CH2_O depends on CH2P bit. 0000: Timing mode. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 0001: Set the channel output on match. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV. 0010: Clear the channel output on match. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV. 0011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV. 0100: Force low. O2CPRE is forced low level. 0101: Force high. O2CPRE is forced high level. 0110: PWM mode 0. When counting up, O2CPRE is active as long as the counter is smaller than TIMERx_CH2CV, otherwise it is inactive. When counting down, O2CPRE is inactive as long as the counter is larger than TIMERx_CH2CV, otherwise it is active. 0111: PWM mode 1. When counting up, O2CPRE is inactive as long as the counter is smaller than TIMERx_CH2CV, otherwise it is active. When counting down, O2CPRE is active as long as the counter is larger than TIMERx_CH2CV, otherwise it is inactive. 1000~1111: Reserved. If configured in PWM mode, the O2CPRE level changes only when the output compare mode switches from “Timing” mode to “PWM” mode or the result of the comparison changes.
3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disabled 1: Channel 2 output compare shadow enabled

		The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).
2	CH2COMFEN	Channel 2 output compare fast enable 0: Channel 2 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and CH2_O is set to the compare level.The minimum delay from an edge on the result of the comparison to activate CH2_O output is 5 clock cycles. 1: Channel 2 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and CH2_O is set to the compare level.The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH2MS[2:0] bit-field is writable only when the channel is not active (the CH2EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 2 is configured as output. 01: Channel 2 is configured as input, IS2 is connected to CI2FE2. 10: Channel 2 is configured as input, IS2 is connected to CI3FE2. 11: Channel 2 is configured as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=1...4) register). 100~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH3MS[2]	Channel 3 I/O mode selection Same as output compare mode.
30	CH2MS[2]	Channel 2 I/O mode selection Same as output compare mode.
31:16	Reserved	Must be kept at reset value.
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH2CAPFLT description.
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH2CAPPSC description.
9:8	CH3MS[1:0]	Channel 3 mode selection Same as output compare mode.
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2.

0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1.
0001: $f_{SAMP}=f_{CK_TIMER}$, N=2.
0010: $f_{SAMP}= f_{CK_TIMER}$, N=4.
0011: $f_{SAMP}= f_{CK_TIMER}$, N=8.
0100: $f_{SAMP}=f_{DTS}/2$, N=6.
0101: $f_{SAMP}=f_{DTS}/2$, N=8.
0110: $f_{SAMP}=f_{DTS}/4$, N=6.
0111: $f_{SAMP}=f_{DTS}/4$, N=8.
1000: $f_{SAMP}=f_{DTS}/8$, N=6.
1001: $f_{SAMP}=f_{DTS}/8$, N=8.
1010: $f_{SAMP}=f_{DTS}/16$, N=5.
1011: $f_{SAMP}=f_{DTS}/16$, N=6.
1100: $f_{SAMP}=f_{DTS}/16$, N=8.
1101: $f_{SAMP}=f_{DTS}/32$, N=5.
1110: $f_{SAMP}=f_{DTS}/32$, N=6.
1111: $f_{SAMP}=f_{DTS}/32$, N=8.

3:2 CH2CAPPSC[1:0] Channel 2 input capture prescaler
This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is cleared.
00: Prescaler disabled, capture is done on each channel input edge.
01: Capture is done every 2 channel input edges.
10: Capture is done every 4 channel input edges.
11: Capture is done every 8 channel input edges.

1:0 CH2MS[1:0] Channel 2 mode selection
Same as output compare mode.

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	Reserved	CH3P	CH3EN	CH2NP	Reserved	CH2P	CH2EN	CH1NP	Reserved	CH1P	CH1EN	CH0NP	Reserved	CH0P	CH0EN
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CH3NP	Channel 3 complementary capture/compare polarity Refer to CH0NP description.

14	Reserved	Must be kept at reset value.
13	CH3P	Channel 3 capture/compare function polarity Refer to CH0P description
12	CH3EN	Channel 3 capture/compare function enable Refer to CH0EN description
11	CH2NP	Channel 2 complementary capture/compare polarity Refer to CH0NP description.
10	Reserved	Must be kept at reset value.
9	CH2P	Channel 2 capture/compare function polarity Refer to CH0P description
8	CH2EN	Channel 2 capture/compare function enable Refer to CH0EN description
7	CH1NP	Channel 1 complementary capture/compare polarity Refer to CH0NP description.
6	Reserved	Must be kept at reset value.
5	CH1P	Channel 1 capture/compare function polarity Refer to CH0P description
4	CH1EN	Channel 1 capture/compare function enable Refer to CH0EN description
3	CH0NP	Channel 0 complementary output polarity When channel 0 complementary is configured in output mode, this bit must be kept at reset value. When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0.
2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. 00: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. 01: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. 10: Reserved.

		11: Noninverted/both CIxFE0's edges.
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
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
31:16	CNT[31:16]	This bit-field indicates the current counter value (bit 16 to 31). Writing to this bit-field can change the value of the counter. This bit-field only for TIMER1.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

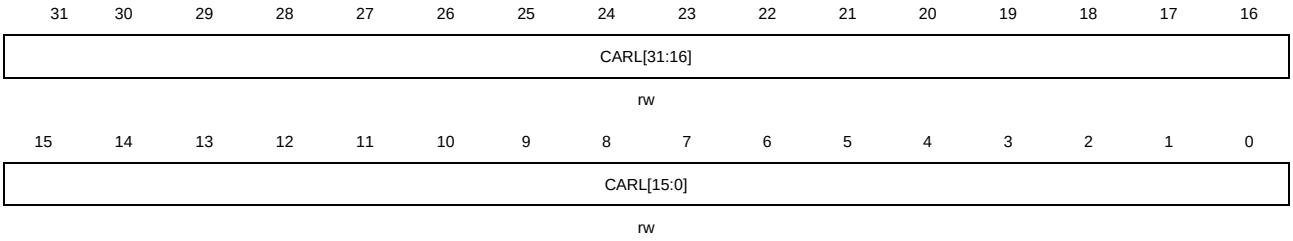
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock

The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 0xFFFF FFFF

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

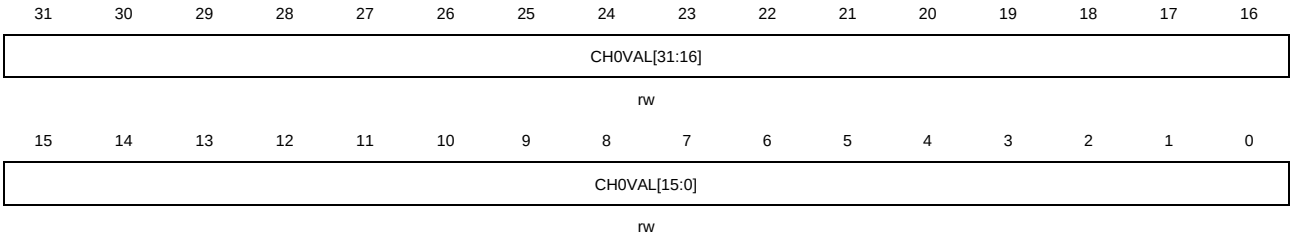


Bits	Fields	Descriptions
31:16	CARL[31:16]	Counter auto reload value (bit 16 to 31) This bit-field only for TIMER1.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

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	CH0VAL[31:16]	Capture/compare value of channel 0 (bit 16 to 31) This bit-field only for TIMER1.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the

shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
31:16	CH1VAL[31:16]	Capture/compare value of channel 1(bit 16 to 31) This bit-field only for TIMER1.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH2VAL[31:16]															
rw															
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	CH2VAL[31:16]	Capture/compare value of channel 2 (bit 16 to 31) This bit-field only for TIMER1.
15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only.

When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3VAL[31:16]															
rw															
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	CH3VAL[31:16]	Capture/compare value of channel 3 (bit 16 to 31) This bit-field only for TIMER1.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Control register 2 (TIMERx_CTL2)

Address offset: 0x74
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												DECDISDEN	DECJDEN	Reserved	
												rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	DECDISDEN	Quadrature decoder signal disconnection detection enable

		0: Quadrature decoder signal disconnection detection is disabled 1: Quadrature decoder signal disconnection detection is enabled
18	DECJDEN	Quadrature decoder signal jump (the two signals jump at the same time) detection enable 0: Quadrature decoder signal jump detection is disabled 1: Quadrature decoder signal jump detection is enabled
17:0	Reserved	Must be kept at reset value.

Watchdog counter period register(TIMERx_WDGPER)

Address offset: 0x94

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
WDGPER[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	WDGPER[31:0]	Watchdog counter period value This register contains the period of the two watchdog counter. When the counters continus to count to this value, the counter will timeout and the interrupt flag DECDISIF is set. If DECDISIE=1, the corresponding interrupt is generated. **Note**: This register is just used in quadrature decoder signal disconnection detection function (with DECDISDEN =1).

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.

13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b111111: transfer 63 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4). 6'b000000: TIMERx_CTL0 6'b000001: TIMERx_CTL1 … In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

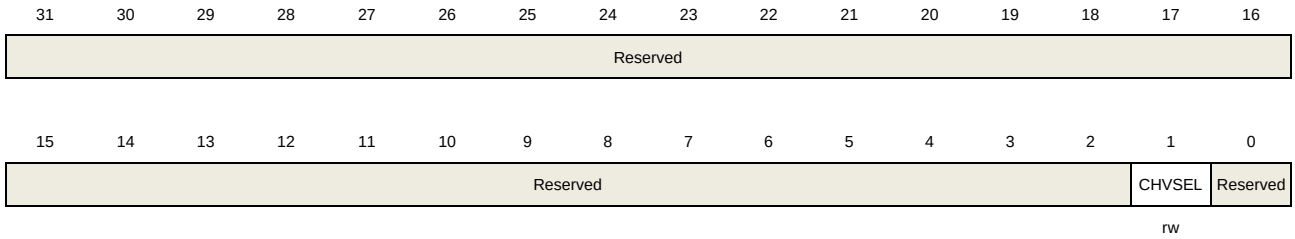
Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	Reserved	Must be kept at reset value.

12.3. General level3 timer (TIMERx, x=15,16)

12.3.1. Overview

The general level3 timer module (TIMER15/16) is a three-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 level3 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time Insertion module which is suitable for motor control applications.

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

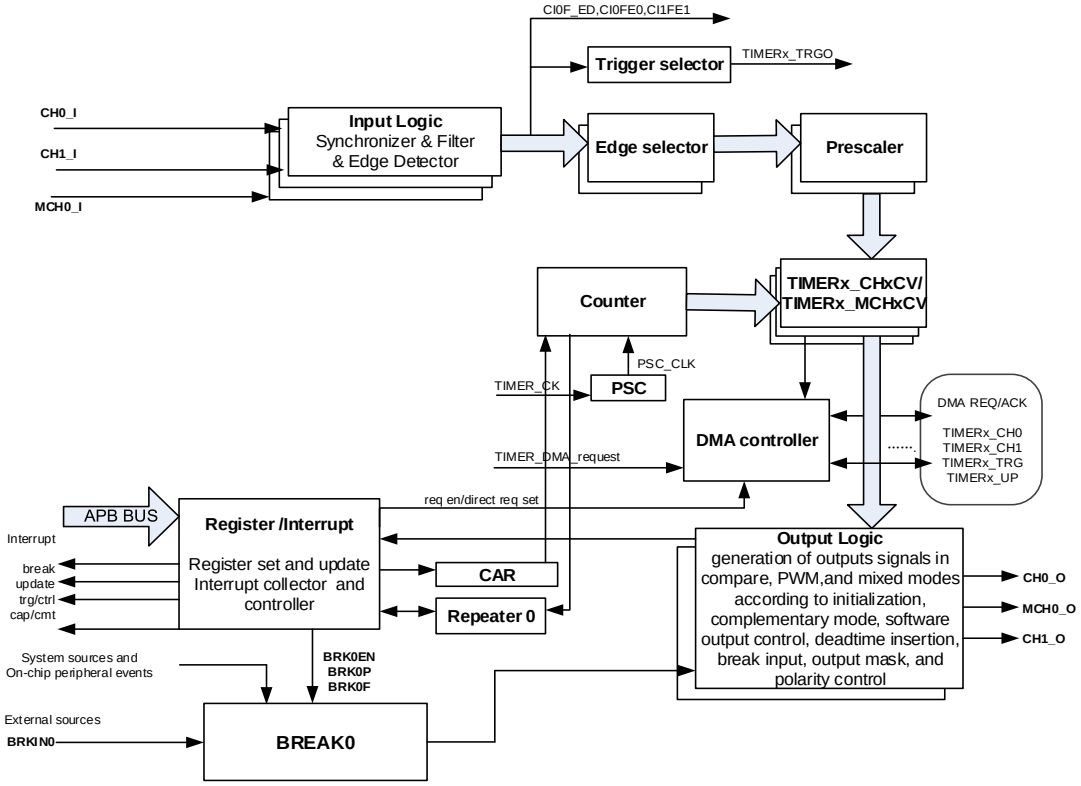
12.3.2. Characteristics

- Total channel num: 3.
- Counter width: 16-bit.
- Source of counter clock is selectable: internal clock, internal trigger, external input.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- 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 function: BREAK.
- Interrupt output or DMA request: 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.

12.3.3. Block diagram

Figure 12-70. General level3 timer block diagram provides details of the internal configuration of the general level3 timer.

Figure 12-70. General level3 timer block diagram



12.3.4. Function overview

Clock selection

The general level3 timer has the capability of being clocked by either the TIMER_CK or an alternate clock source controlled by TSCFGy[4:0] (y=3..6,15) in SYSCFG_TIMERxCFG (x=15,16) registers.

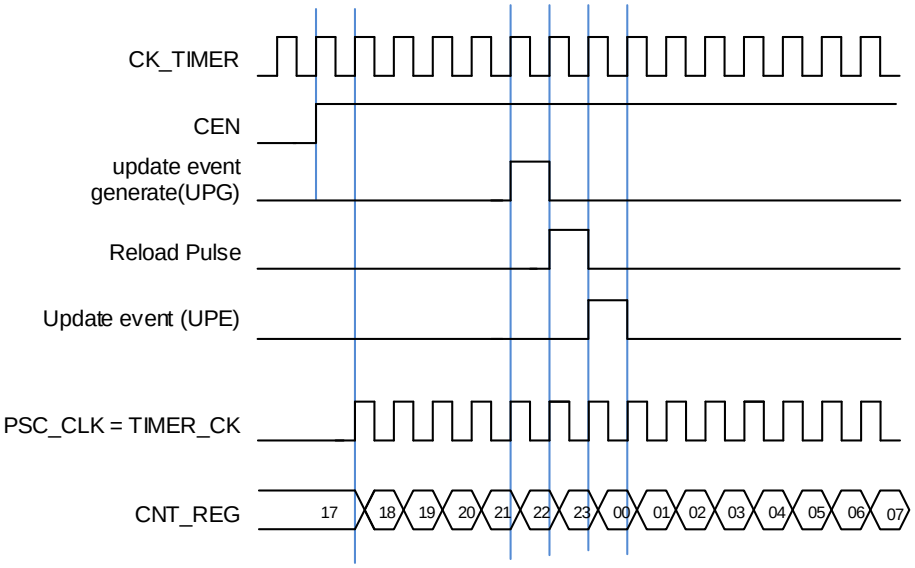
- TSCFGy[4:0] (y=3..6,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=15,16) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=3..6,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=15,16) registers. 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 TSCFG6[4:0] bit-filed in SYSCFG_TIMERxCFG(x=0,7) registers are setting to a nonzero value, the prescaler is clocked by other clock sources selected in the TSCFG6[4:0] bit-filed, more details will be introduced later. When the TSCFGy[4:0] (y=3,4,5) are setting to an available value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 12-71. Normal mode, internal clock divided by 1



- TSCFG6[4:0] are setting to a nonzero value (external clock mode 0). External input pin is selected as timer clock source

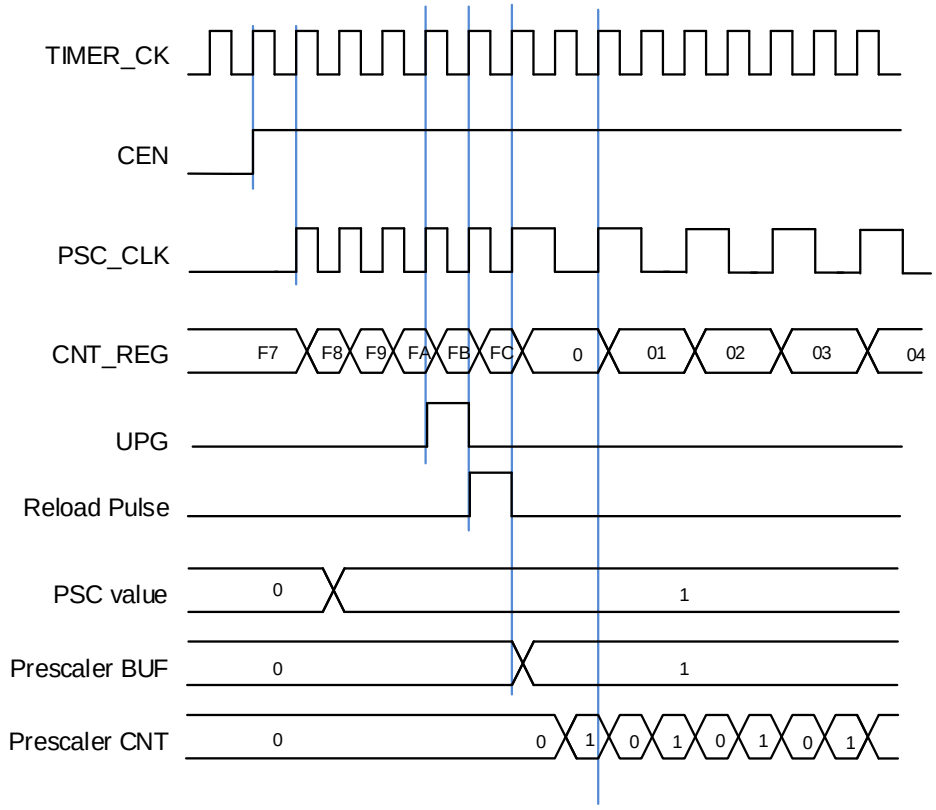
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7 and 0xB.

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 TSCFG6[4:0] to 0x1~0x4.

Clock prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed on the go but is taken into account at the next update event.

Figure 12-72. Counter timing diagram with prescaler division change from 1 to 2



Up counting mode

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 restarts from 0. If the repetition counter is set, the update events will be generated after (TIMERx_CREP0+1) times of overflow. Otherwise the update event is generated each time when overflows. 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 registers (repetition counter, auto reload register, prescaler register) are updated.

Figure 12-73. Timing diagram of up counting mode, PSC=0/2 and *Figure 12-74. Timing diagram 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 12-73. Timing diagram of up counting mode, PSC=0/2

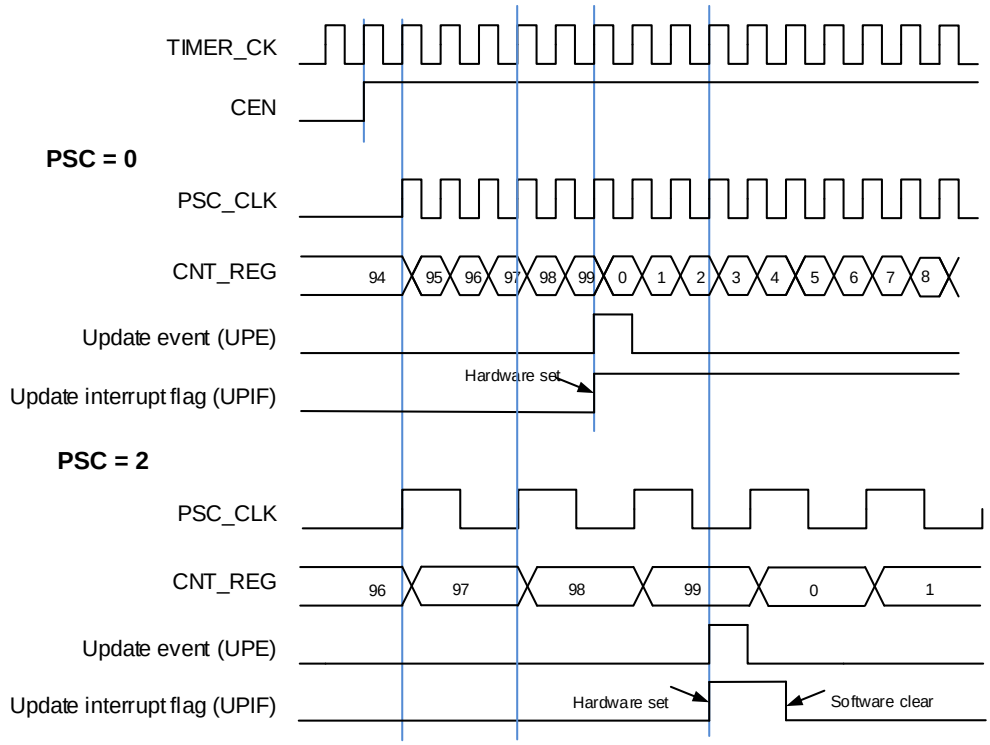
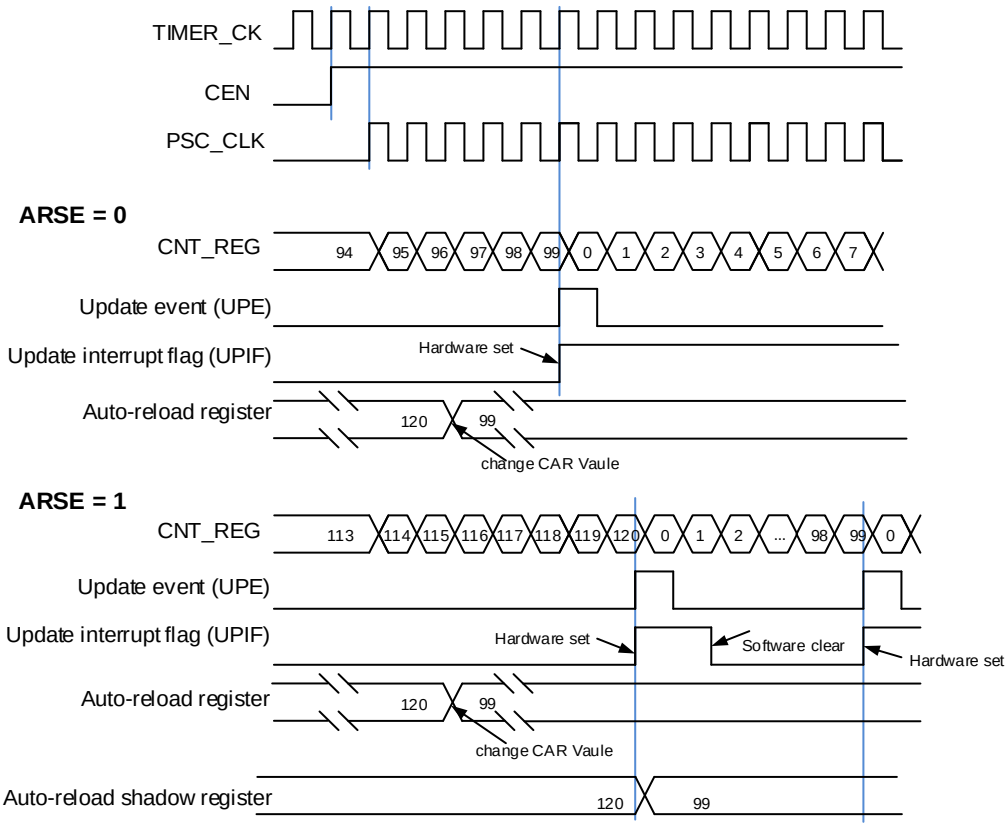


Figure 12-74. Timing diagram of up counting mode, change TIMERx_CAR on the go



Down counting mode

In this mode, the counter counts down continuously from the counter reload value, which is defined in the TIMERx_CAR register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. If the repetition counter is set, the update event will be generated after (TIMERx_CREP0 + 1) times of underflow. Otherwise, the update event is generated each time when counter underflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

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

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

Figure 12-75. Timing diagram of down counting mode, PSC=0/2 and *Figure 12-76. Timing diagram of down counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior in different clock frequencies when TIMERx_CAR = 0x99.

Figure 12-75. Timing diagram of down counting mode, PSC=0/2

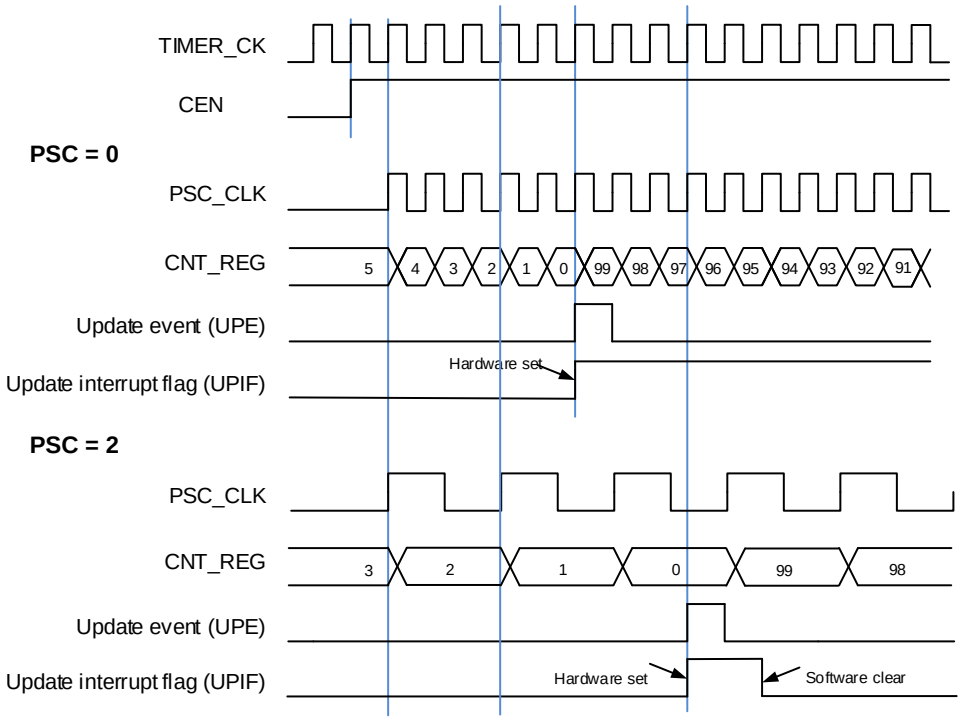
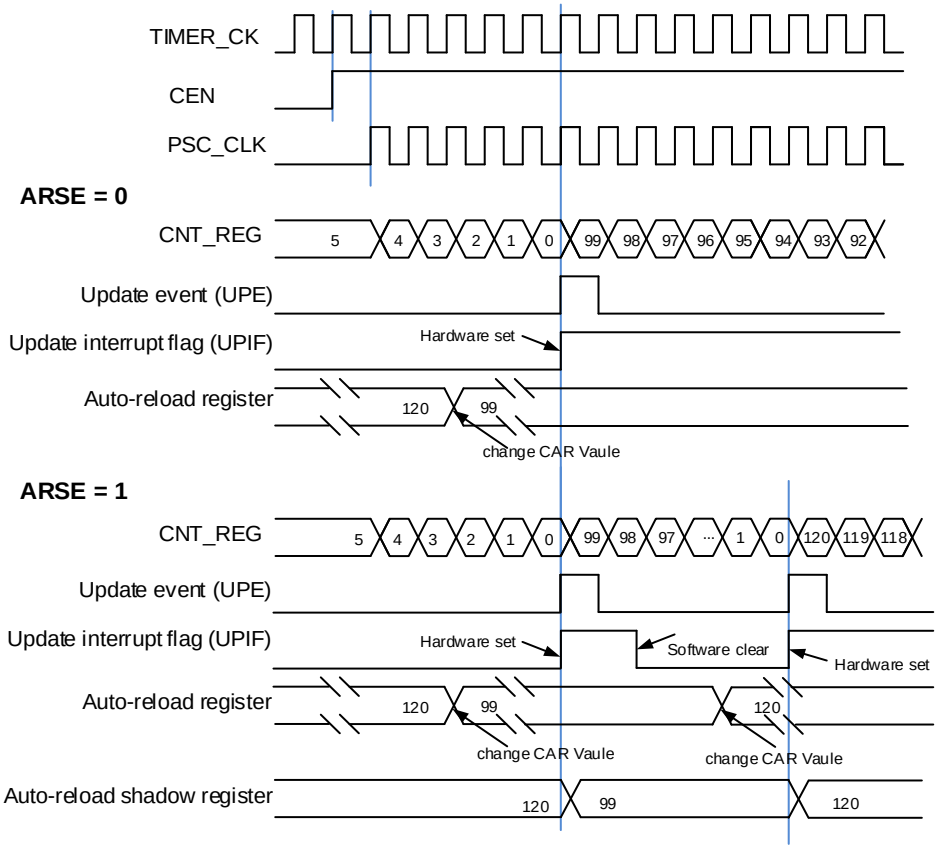


Figure 12-76. Timing diagram of down counting mode, change TIMERx_CAR ongoing



Center-aligned counting mode

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 (TIMERx_CAR-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned counting mode. The counting direction is updated by hardware automatically.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

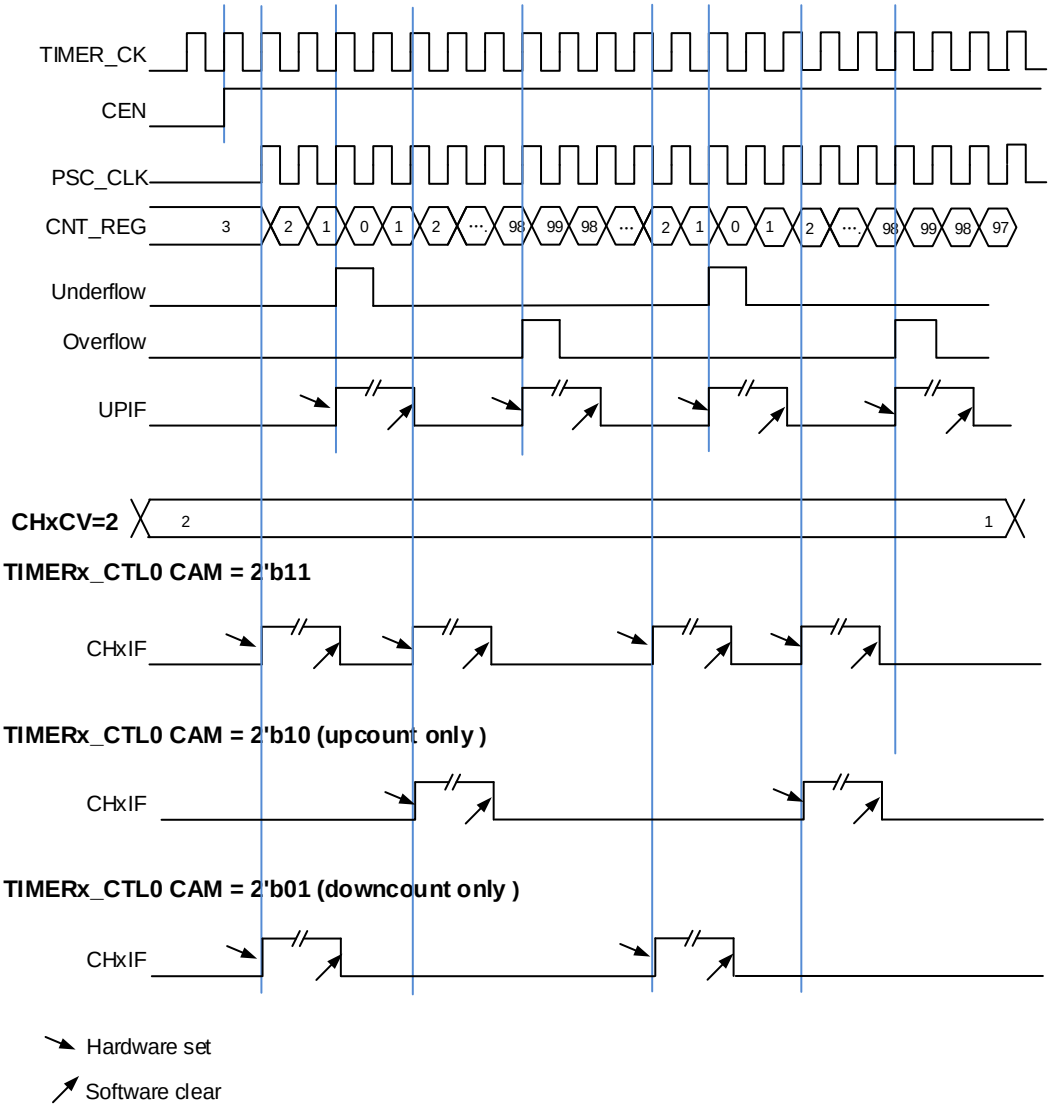
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM[1:0] in TIMERx_CTL0. The details refer to ***Figure 12-77. Timing diagram of center-aligned counting mode***.

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

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

Figure 12-77. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 12-77. Timing diagram of center-aligned counting mode



Counter repetition

Repetition counter is used to generate the update event or update the timer registers only after a given number (N+1) cycles of the counter, where N is the value of CREP0 bit in TIMERx_CREP0 register. The repetition counter is decremented at each counter overflow in up counting mode, at each counter underflow in down counting mode or at each counter overflow and at each counter underflow in center-aligned counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP0 in TIMERx_CREP0 register and generate an update event.

The new written CREP0 value will not take effect until the next update event. When the value of CREP0 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 CREP0 value takes effect. If an update event is generated by software after writing an odd number to CREP0, the update events will be generated on the underflow. If the next update event occurs on overflow after writing an odd number to CREP0, then the subsequent update events will be generated on the overflow.

Figure 12-78. Repetition counter timing diagram of center-aligned counting mode

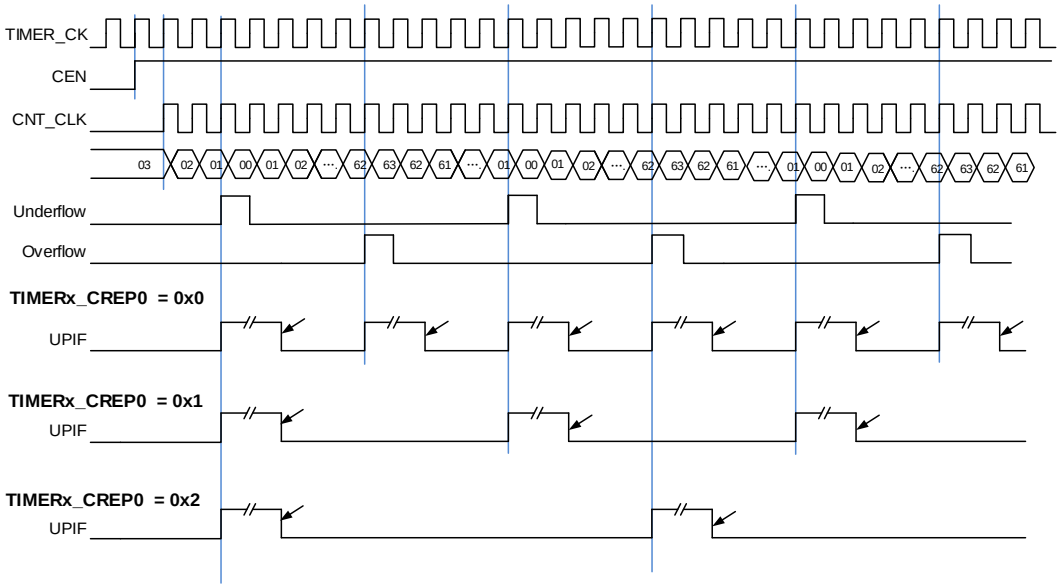


Figure 12-79. Repetition counter timing diagram of up counting mode

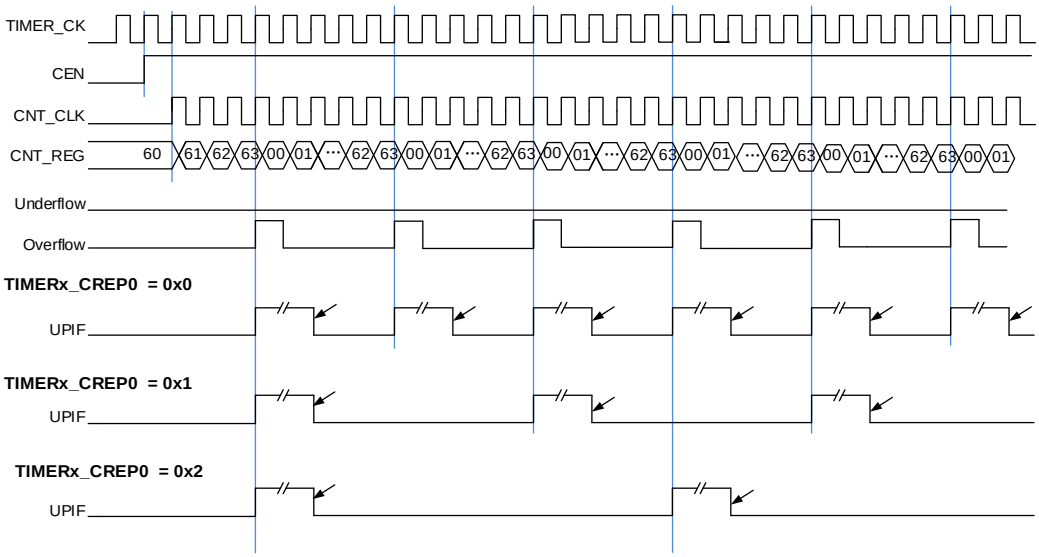
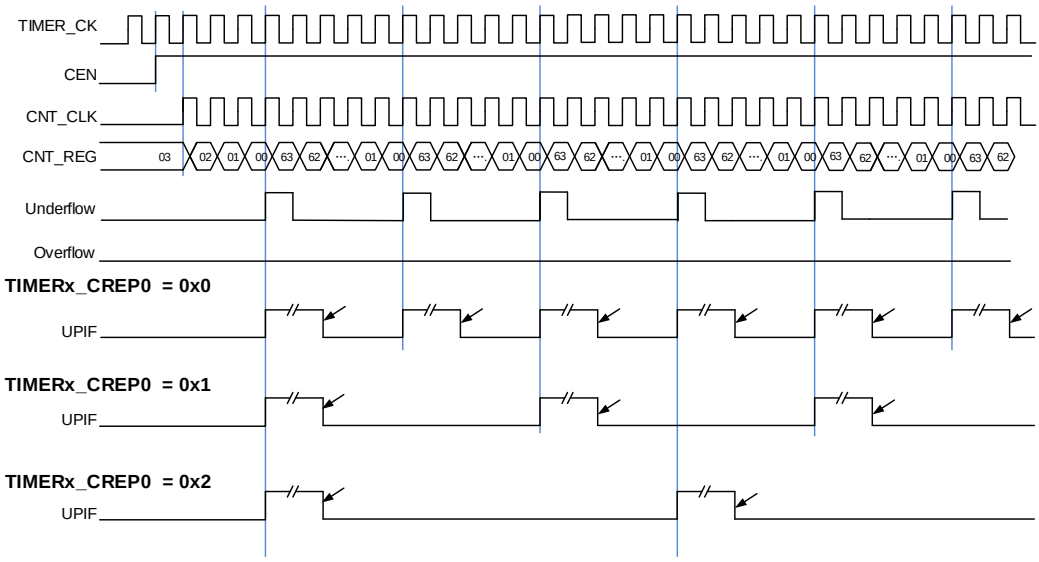


Figure 12-80. Repetition counter timing diagram of down counting mode



Capture/compare channels

The general level3 timer has three 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.

When the channels are used for input, channel 0 and multi mode channel 0 can perform input capture independently; when the channels are used for comparison output, the channel 0 and multi mode channel 0 can output independent and complementary outputs.

■ Input capture mode

When MCHxMSEL=2'b00 (independent mode), channel x and multi mode channel x can perform input capture independently.

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV/ TIMERx_MCH0CV(x=0, 1) registers, at the same time the CHxIF/ MCH0IF(x=0, 1) bits are set and the channel interrupt is generated if it is enabled when CHxIE/ MCH0IE =1(x=0, 1).

Figure 12-81. Input capture logic for channel 0

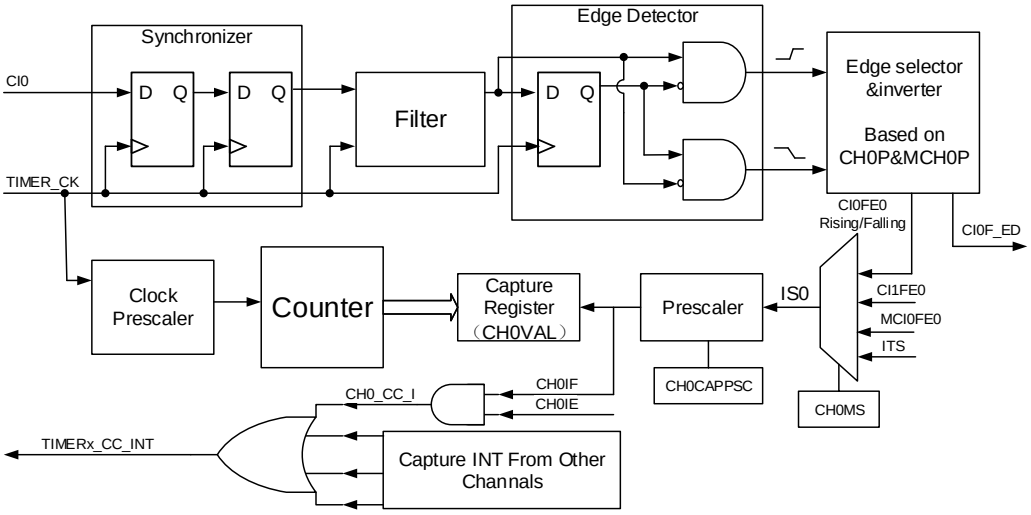
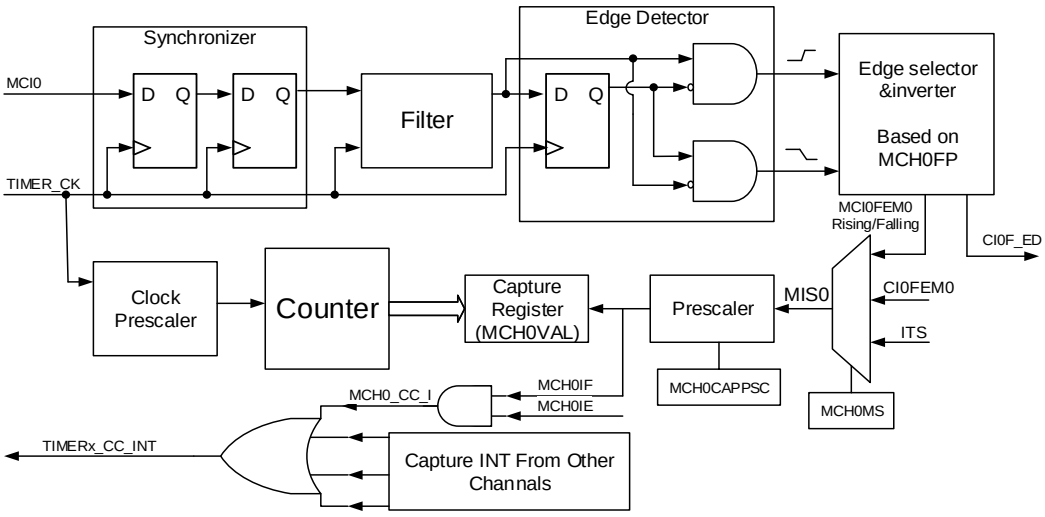


Figure 12-82. Input capture logic for multi mode channel 0



The input signals of channelx (CIx/ MCIx) are the TIMERx_CHx/ TIMERx_MCHxCV signal.

First, the input signal of channel (CIx/ MCIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP/ MCHxP or MCHxFP bits. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS/ MCHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV/ TIMERx_MCHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT bit in TIMERx_CHCTL0 register and MCHxCAPFLT bit in TIMERx_MCHCTL0 register).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT or MCHxCAPFLT bit.

Step2: Edge selection (CHxP and MCHxP bits in TIMERx_CHCTL2 register, MCHxFP[1:0] bits in TIMERx_MCHCTL2 register).

Rising edge or falling edge, choose one by configuring CHxP and MCHxP bits or MCHxFP[1:0] bits.

Step3: Capture source selection (CHxMS bit in TIMERx_CHCTL0 register, MCHxMS bit in TIMERx_MCHCTL0 register).

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x000 or MCHxMS != 0x000) and TIMERx_CHxCV/ TIMERx_MCHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN and MCHxEN bits in TIMERx_CHCTL2).

Result: When the wanted input signal is captured, TIMERx_CHxCV/ TIMERx_MCHxCV will be set by counter's value and CHxIF/ MCHxIF bit is asserted. If the CHxIF/ MCHxIF bit is 1, the CHxOF/ MCHxOF bit will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN bits, MCHxIE and MCHxDEN bits in TIMERx_DMAINTEN.

Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx and TIMERx_MCHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 3'b001 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 3'b010 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

- **Output compare mode**

Figure 12-83. Output compare logic (when MCHxMSEL = 2'00, x=0), *Figure 12-84. Output compare logic (when MCHxMSEL = 2'11, x=0)* and *Figure 12-85. Output compare logic (x=1)* show the logic circuit of output compare mode.

Figure 12-83. Output compare logic (when MCHxMSEL = 2'00, x=0)

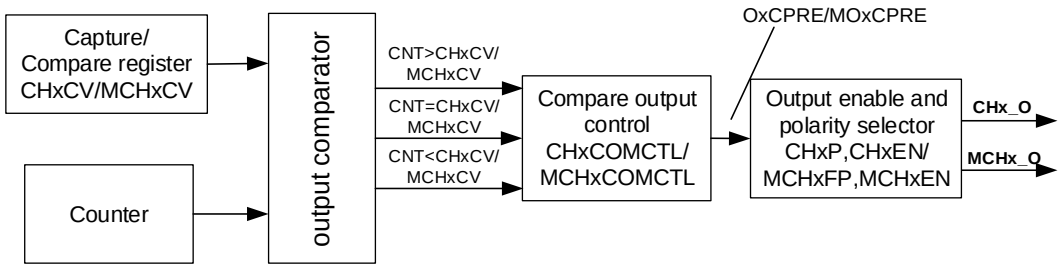


Figure 12-84. Output compare logic (when MCHxMSEL = 2'11, x=0)

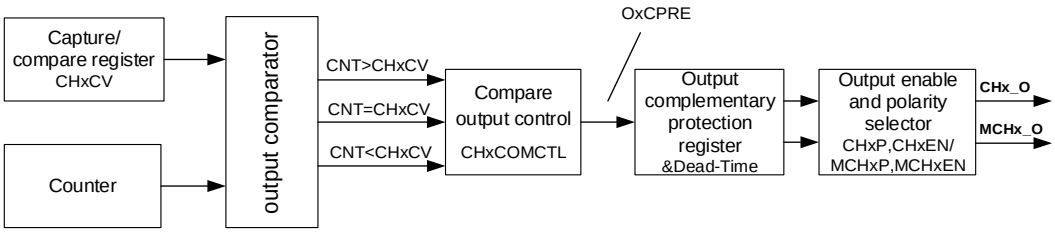
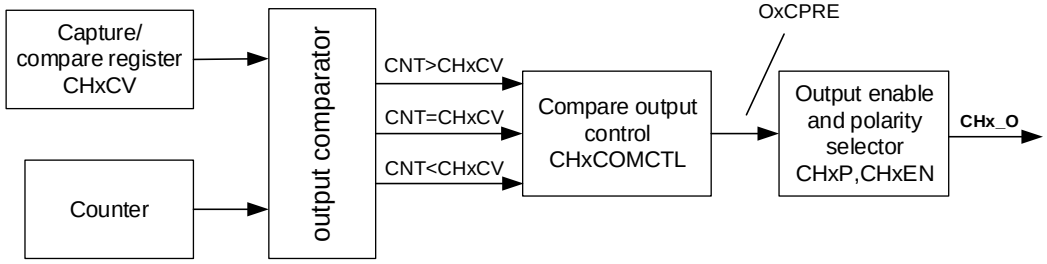


Figure 12-85. Output compare logic (x=1)



The relationship between the channel output signal CHx_O/MCHx_O and the OxCPRE/ MOxCPRE signal (more details refer to ***Channel output prepare signal***) is described as blew(the active level of OxCPRE is high and the active level of MOxCPRE is high).

- When MCHxMSEL=2'b00 (in TIMERx_CTL2 register), the MCHx_O output is independent from the CHx_O output. The output level of CHx_O depends on OxCPRE signal, CHxP bit and CHxEN bit (please refer to the TIMERx_CHCTL2 register for more details). The output level of MCHx_O depends on MOxCPRE signal, MCHxFP[1:0] bits and MCHxEN bit (please refer to the TIMERx_MCHCTL2 and TIMERx_CHCTL2 registers for more details). Please refer to ***Figure 12-83. Output compare logic (when MCHxMSEL = 2'00, x=0)***.
- When MCHxMSEL=2'b11, the MCHx_O output is the inverse of the CHx_O output. The output level of CHx_O/MCHx_O depends on OxCPRE signal, CHxP/ MCHxP bits and CHxEN/MCHxEN bits. Please refer to ***Figure 12-84. Output compare logic (when***

MCHxMSEL = 2'11, x=0).

For examples (the MCHx_O output is independent from the CHx_O output):

1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxEN=1 (the output of CHx_O is enabled):

If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

2) Configure MCHxP=1 (the active level of MCHx_O is low, contrary to MOxCPRE), MCHxEN=1 (the output of MCHx_O is enabled):

If the output of MOxCPRE is active(high) level, the output of MCHx_O is active(low) level;

If the output of MOxCPRE is inactive(low) level, the output of MCHx_O is active(high) level.

When MCHxMSEL=2'b11 and CHx_O and MCHx_O are output at the same time, the specific outputs of CHx_O and MCHx_O are related to the relevant bits (ROS, IOS, POE and DTCFG bits) in the TIMERx_CCHP register. Please refer to ***Outputs complementary*** for more details.

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 TIMERx_CHxCV/ TIMERx_MCHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL/ MCHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV/ TIMERx_MCHxCV register, the CHxIF/ MCHxIF bit will be set and the channel (n) interrupt is generated if CHxIE/ MCHxIE = 1. And the DMA request will be asserted, if CHxDEN/ MCHxDEN =1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN/ MCHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL/ MCHxCOMCTL.
- Select the active polarity by CHxP/MCHxP/ MCHxFP.
- Enable the output by CHxEN/ MCHxEN.

Step3: Interrupt/DMA request enable configuration by CHxIE/ MCHxIE /CHxDEN/ MCHxDEN.

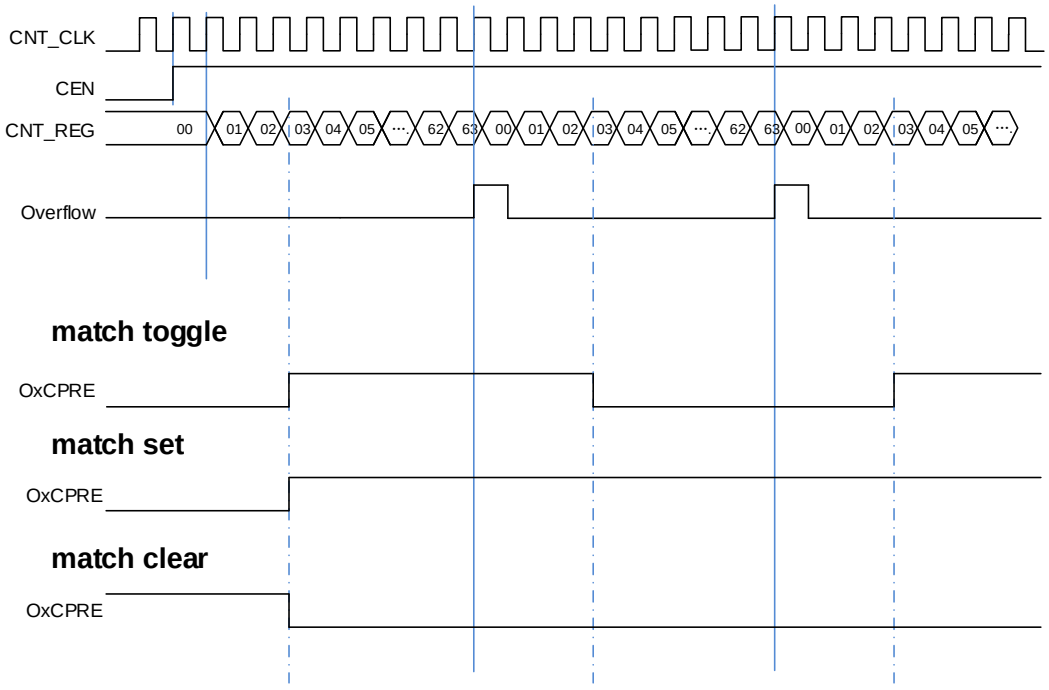
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV/ TIMERx_MCHxCV.

The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 12-86. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 12-86. Output-compare in three modes



PWM mode

In the PWM output mode (by setting the CHxCOMCTL/ MCHxCOMCTL bit to 4'b0110 (PWM mode 0) or to 4'b0111(PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV/ TIMERx_MCHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV/ TIMERx_MCHxCV. ***Figure 12-87. Timing diagram of EAPWM*** shows the EAPWM output and interrupts waveform.

The CAPWM's period is determined by 2*TIMERx_CAR, and the duty cycle is determined by 2*TIMERx_CHxCV/ TIMERx_MCHxCV. ***Figure 12-88. Timing diagram 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 counter, the output will be always active in PWM mode 0 (CHxCOMCTL=4'b0110). And if the value of TIMERx_CHxCV is greater than the value of counter, the output will be always inactive in PWM mode 1 (CHxCOMCTL=4'b0111).

Figure 12-87. Timing diagram of EAPWM

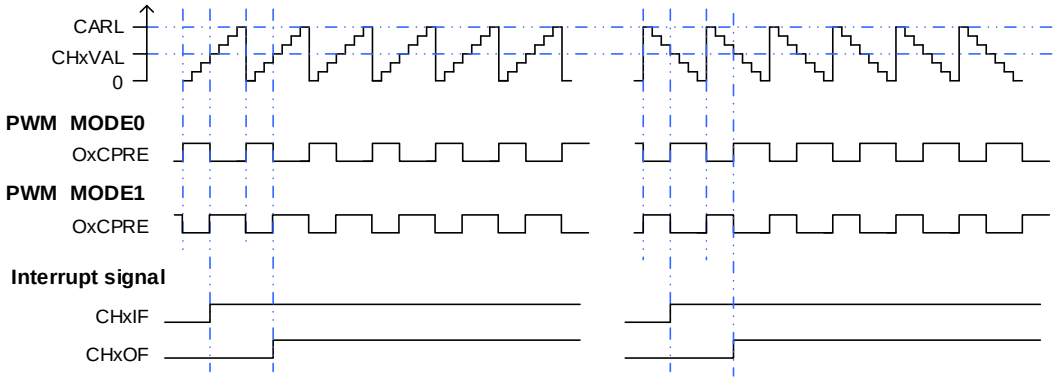
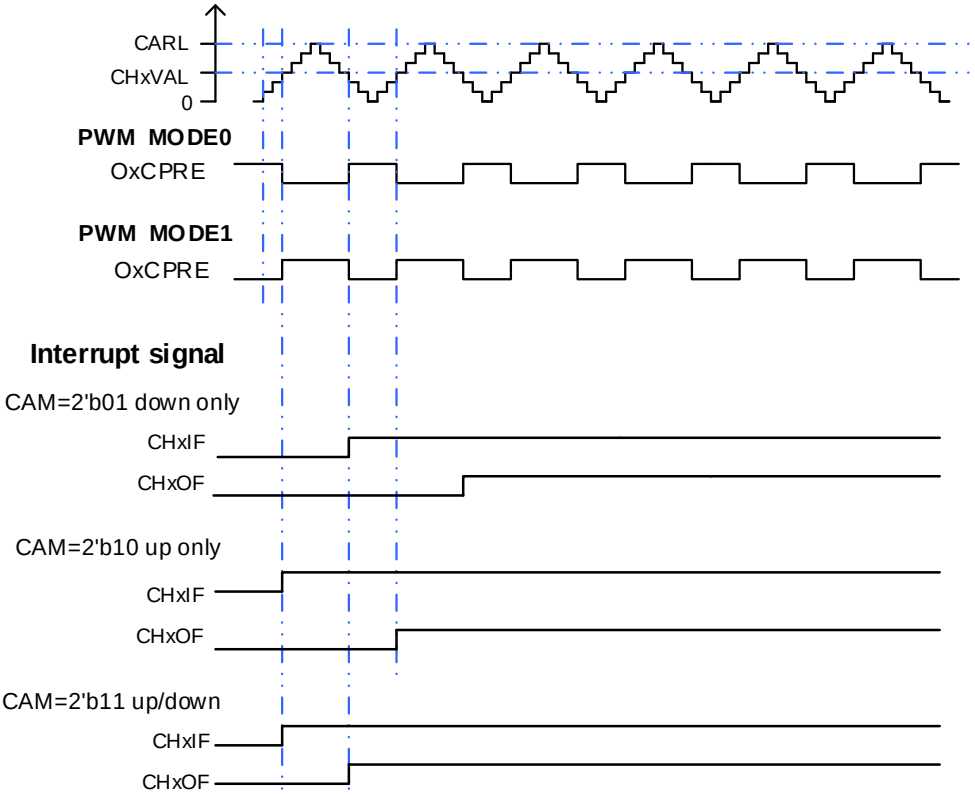


Figure 12-88. Timing diagram of CAPWM



Composite PWM mode

In the Composite PWM mode (CHxCPWMEN = 1'b1, CHxMS[2:0] = 3'b000 and CHxCOMCTL = 4'b0110 or 4'b0111), the PWM signal output in channel x (x=0, 1) is composited by CHxVAL and CHxCOMVAL_ADD bits.

If CHxCOMCTL = 4'b0110 (PWM mode 0) and DIR = 1'b0 (up counting mode), the channel x output is forced low when the counter matches the value of CHxVAL. It is forced high when the counter matches the value of CHxCOMVAL_ADD.

If CHxCOMCTL = 4'b0111 (PWM mode 1) and DIR = 1'b0 (up counting mode), the channel x output is forced high when the counter matches the value of CHxVAL. It is forced low when

the counter matches the value of CHxCOMVAL_ADD.

When CHxVAL or CHxCOMVAL_ADD = 0 / CARL, the channel x output is specially processed, and by setting the CHxPERFOREN bit in the TIMERx_CHCTL2 register, the OxCPRE output is forced to either high or low level (determined according to the selected composite PWM mode).

The PWM period is determined by (CARL + 0x0001) and the PWM pulse width is determined by the ***Table 12-12.The Composite PWM pulse width***.

Table 12-12.The Composite PWM pulse width

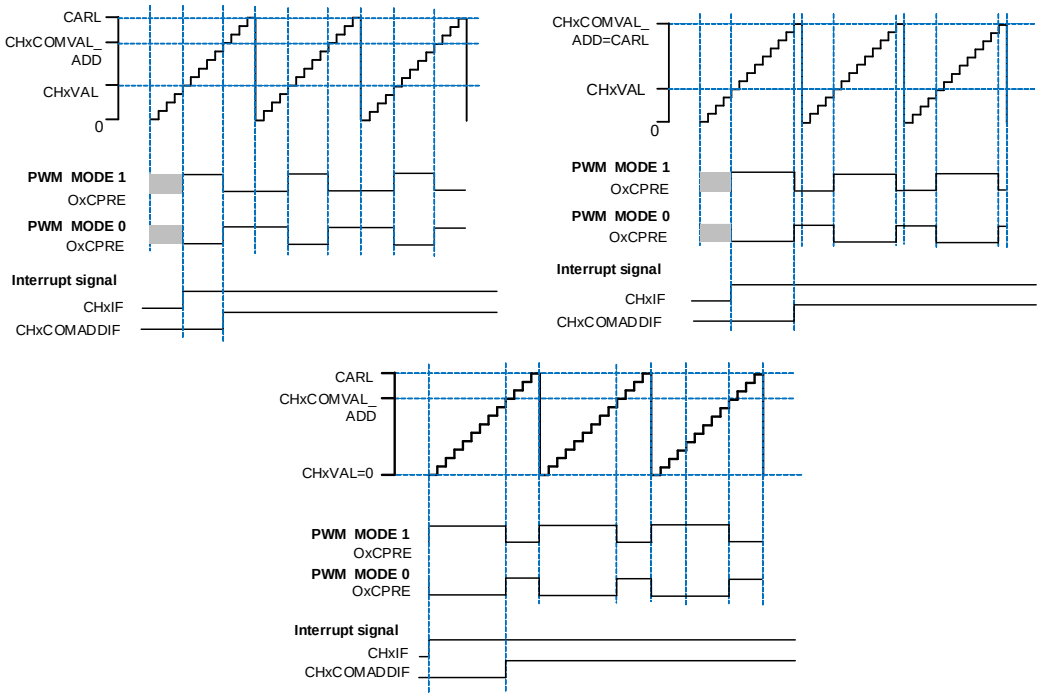
Condition	Mode	PWM pulse width
CHxVAL < CHxCOMVAL_ADD ≤ CARL	PWM mode 0	(CARL + 0x0001) + (CHxVAL – CHxCOMVAL_ADD)
	PWM mode 1	(CHxCOMVAL_ADD – CHxVAL)
CHxCOMVAL_ADD < CHxVAL ≤ CARL	PWM mode 0	(CHxVAL - CHxCOMVAL_ADD)
	PWM mode 1	(CARL + 0x0001) + (CHxCOMVAL_ADD – CHxVAL)
(CHxVAL = CHxCOMVAL_ADD ≤ CARL) or (CHxVAL > CARL > CHxCOMVAL_ADD)	PWM mode 0 (up counting) or PWM mode 1 (down counting)	100%
	PWM mode 0 (down counting) or PWM mode 1 (up counting)	0%
CHxCOMVAL_ADD > CARL > CHxVAL	PWM mode 0(up counting) or PWM mode 1(down counting)	0%
	PWM mode 0(down counting) or PWM mode 1(up counting)	100%
(CHxVAL>CARL) and (CHxCOMVAL_ADD > CARL)	-	The output of CHx_O is keeping

When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when CHxCPWMEN=1) and the channel x additional compare interrupt is generated if CHxCOMADDIE = 1 (Only interrupt is generated, no DMA request is generated).

According to the relationship among CHxVAL, CHxCOMVAL_ADD and CARL, it can be divided into four situations:

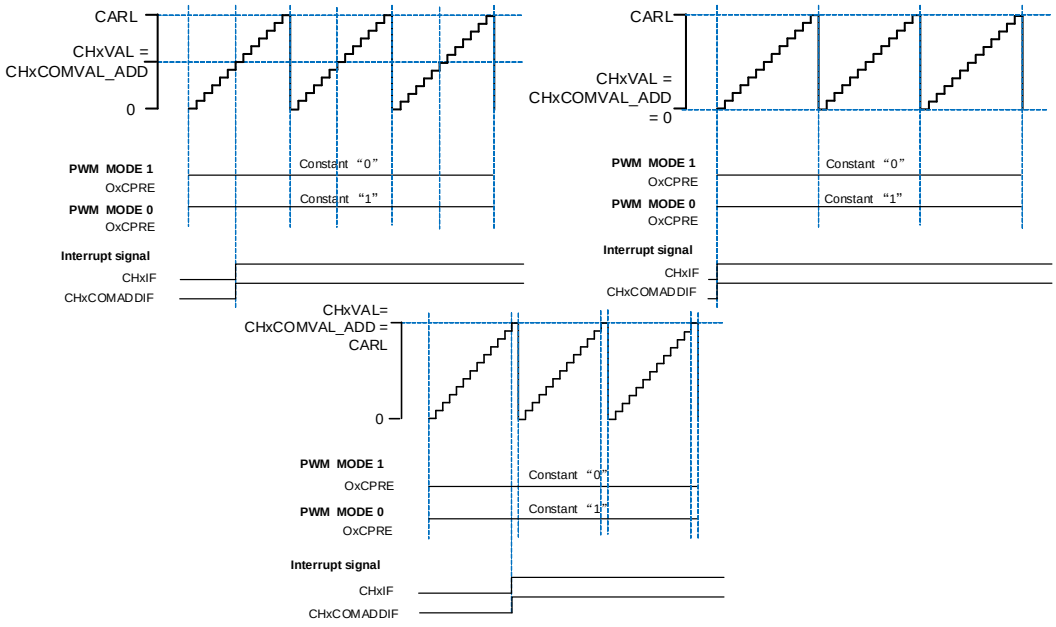
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-89. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



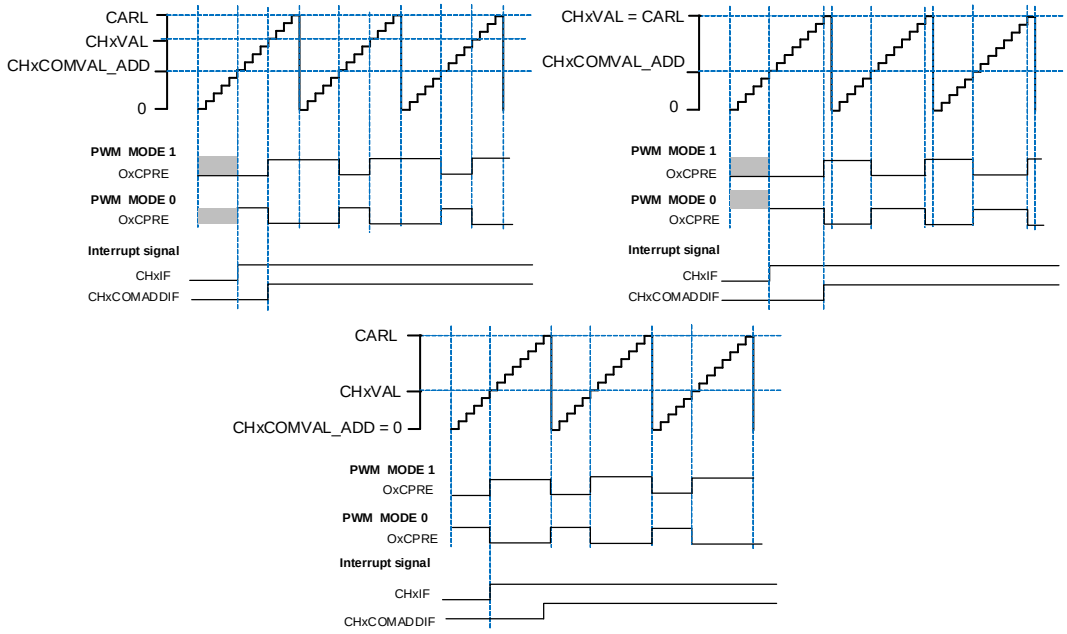
2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-90. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



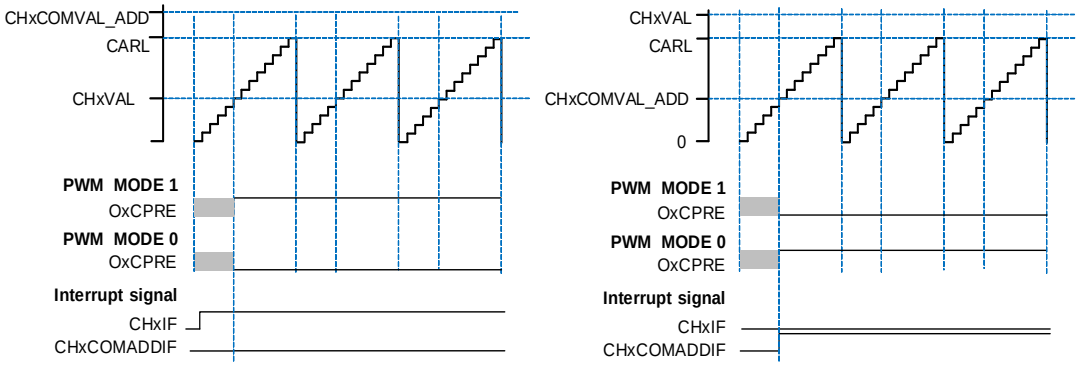
3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 12-91. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

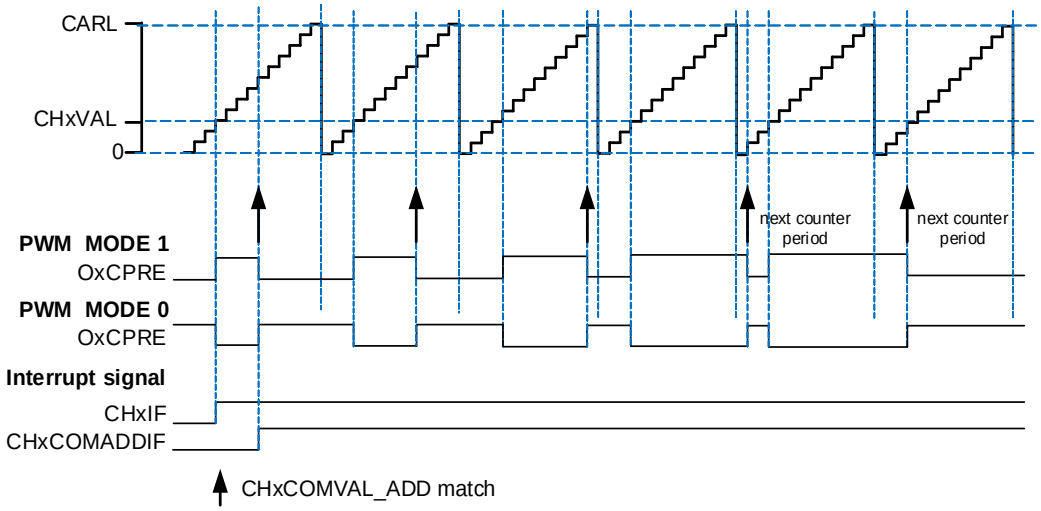
Figure 12-92. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



The composite PWM mode is intended to support the generation of PWM signals where the period is not modified while the signal is being generated, but the duty cycle will be varied. ***Figure 12-93. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD*** shows the PWM output and interrupts waveform.

In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 12-93. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD



If more than one channels are configured in composite PWM mode, it is possible to fix an offset for the channel x match edge of each pair with respect to other channels. This behavior is useful in the generation of lighting PWM control signals where it is desirable that edges are not coincident with each other pair to help eliminate noise generation. The CHxVAL register value is the shift of the PWM pulse with respect to the beginning of counter period.

Output match pulse select

Basing on that CHx_O (x=0, 1) outputs are configured by CHxCOMCTL[3:0] (x=0, 1) bits when the match events occur, the output signal is configured by CHxOMPSEL[1:0] (x=0, 1) bit to be normal or a pulse.

When the match events occur, the CHxOMPSEL[1:0] (x=0, 1) bits are used to select the output of OxCPRE which drives CHx_O:

- CHxOMPSEL = 2'b00, the OxCPRE signal is output normally with the configuration of CHxCOMCTL[3:0] bits;
- CHxOMPSEL = 2'b01, only the counter is counting up, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b10, only the counter is counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.
- CHxOMPSEL = 2'b11, both the counter is counting up and counting down, the OxCPRE signal is output a pulse when the match events occur, and the pulse width is one CK_TIMER clock cycle.

Figure 12-94. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL≠2'b00)

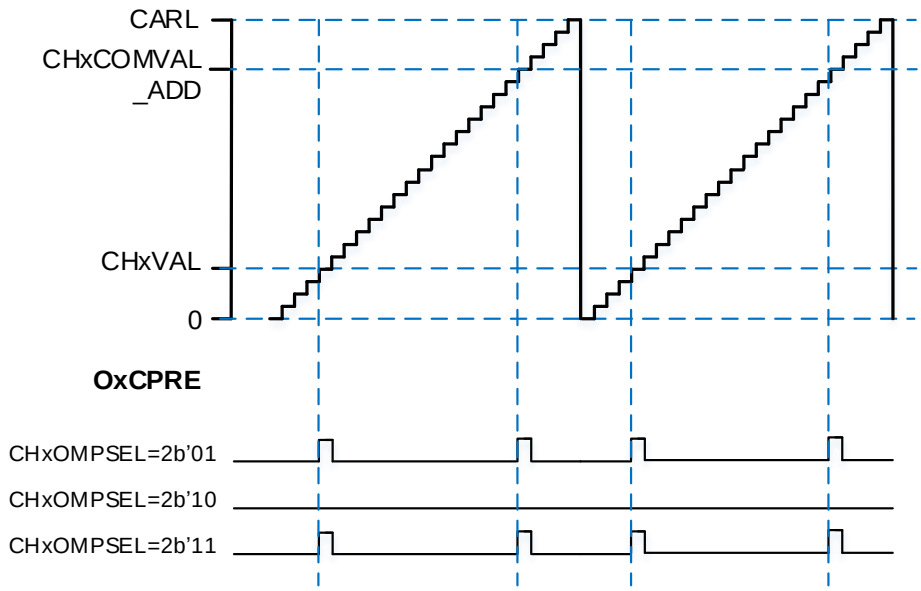
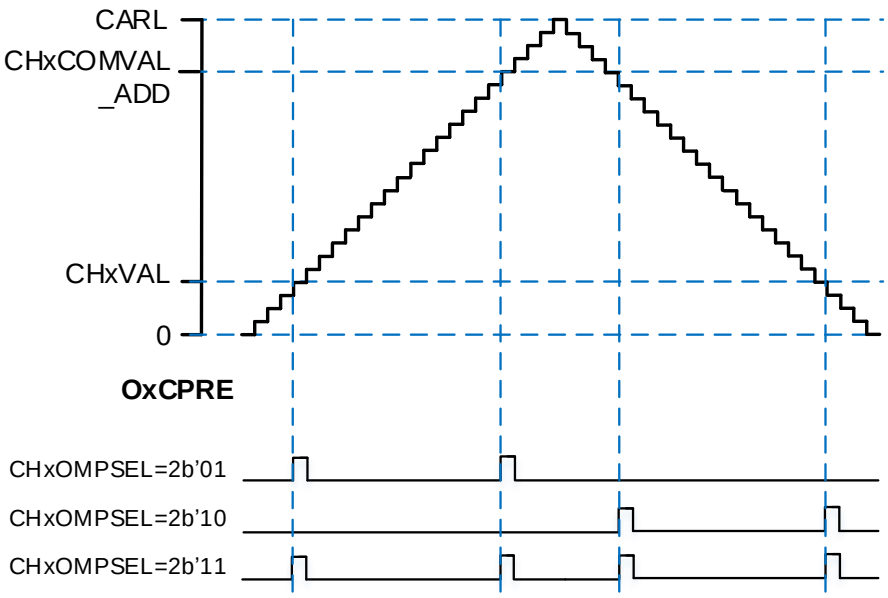


Figure 12-95. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL≠2'b00)



Channel output prepare signal

As is shown in ***Figure 12-83. Output compare logic (when MCHxMSEL = 2'00, x=0)***, ***Figure 12-84. Output compare logic (when MCHxMSEL = 2'11, x=0)*** and ***Figure 12-85. Output compare logic (x=1)***, when TIMERx is configured in compare match output mode, a middle signal named OxCPRE or MOxCPRE (channel x output or multi mode channel x output prepare signal) will be generated before the channel outputs signal.

The OxCPRE and MOxCPRE signal have several types of output function. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit and the MOxCPRE signal type is defined by configuring the MCHxCOMCTL bit.

Take OxCPRE as an example for description below, these include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/ PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

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

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

Outputs complementary

The outputs of CHx_O and MCHx_O have two situations:

- MCHxMSEL=2'b00: The MCHx_O output is independent from the CHx_O output;
- MCHxMSEL=2'b11: The outputs of MCHx_O and CHx_O are complementary and the MCHxOMCTL bits are not used in the generation of the MCHx_O output.

Function of complementary is for a pair of channels, CHx_O and MCHx_O, the two output signals cannot be active at the same time. TIMERx's one pair of channel has this function. The complementary signals CHx_O and MCHx_O are controlled by a group of parameters: the CHxEN and MCHxEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and MCHxP bits in the TIMERx_CHCTL2 register.

When the the outputs of CHx_O and MCHx_O are complementary, there are three situations: output enable、output off-state and output disabled. The details are shown in ***Table 12-13. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)***.

Table 12-13. Complementary outputs controlled by parameters (MCHxMSEL =2'b11)

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / MCHx_O = LOW CHx_O / MCHx_O output disable(1).	
				1	CHx_O/ MCHx_O output "off-state" (2): the CHx_O/ MCHx_O output inactive level firstly: CHx_O = CHxP, MCHx_O = MCHxP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, MCHx_O = ISOxN. (3)	
			1	0		
				1		
		1	x	x	CHx_O/ MCHx_O output "off-state": the CHx_O/ MCHx_O output inactive level firstly: CHx_O = CHxP, MCHx_O = MCHxP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, MCHx_O = ISOxN.	
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O output disable.	
				1	CHx_O = LOW CHx_O output disable.	MCHx_O=OxCPRE⊕(4)MCHxP MCHx_O output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = LOW MCHx_O output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	MCHx_O = (! OxCPRE)(5)⊕MCHxP. MCHx_O output enable.
	1		0	0	CHx_O = CHxP CHx_O output "off-state".	MCHx_O = MCHxP MCHx_O output "off-state".
				1	CHx_O = CHxP CHx_O output "off-state"	MCHx_O=OxCPRE⊕MCHxP MCHx_O output enable
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable	MCHx_O = MCHxP MCHx_O output "off-state".
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable	MCHx_O = (! OxCPRE)⊕MCHxP MCHx_O output enable.

Note:

(1) output disable: the CHx_O / MCHx_O 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 / MCHx_O 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

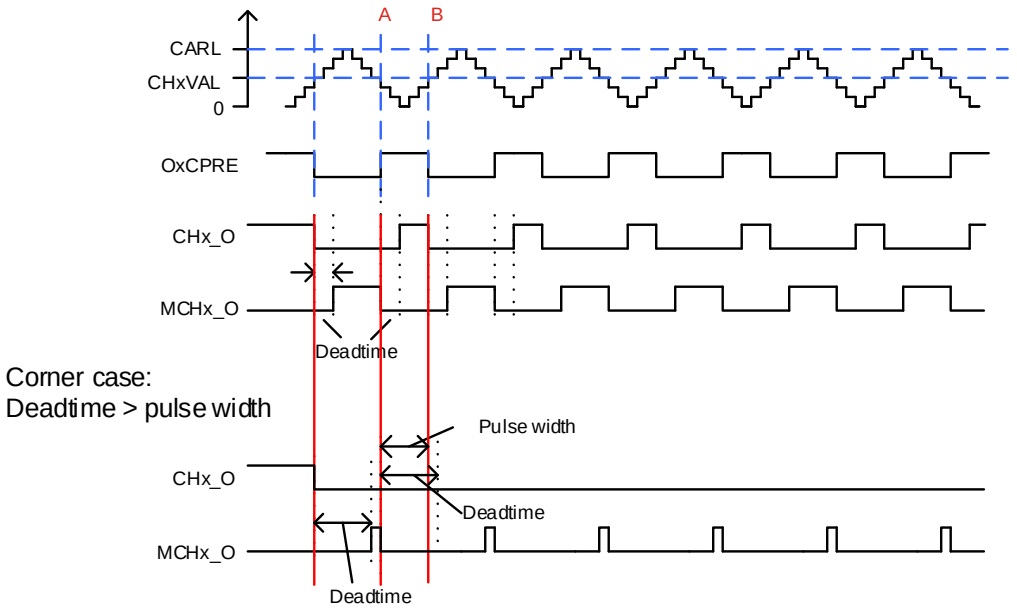
The dead time insertion is enabled when MCHxMSEL=2'b11 and both CHxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the ***Complementary channel protection register 0 (TIMERx_CCHP0)*** for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channel x match event (TIMERx_CNT = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 12-96. Complementary output with dead-time insertion***, CHx_O signal remains at the low level until the end of the dead time delay, while MCHx_O signal will be cleared at once. Similarly, at point B when the channelx match event (TIMERx_CNT = CHxVAL) occurs again, OxCPRE is cleared, and CHx_O signal will be cleared at once, while MCHx_O signal remains at the low level until the end of the dead time delay.

Sometimes, we can see corner cases about the dead time insertion. For example: the dead time delay is greater than or equal to the duty cycle of the CHx_O signal, then the CHx_O signal is always inactive (As shown in ***Figure 12-96. Complementary output with dead-time insertion***).

Figure 12-96. Complementary output with dead-time insertion



Break function

The MCHx_O output is the inverse of the CHx_O output when the MCHxMSEL=2'b11 (and the MCHxOMCTL bits are not used in the generation of the MCHx_O output). In this case, CHx_O and MCHx_O signals cannot be set to active level at the same time.

The general level3 timers have the BREAK function. The BREAK function can be enabled by

setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is configured by the BRKP bit in TIMERx_CCHP register, the input is active on level.

In BREAK function, CHx_O and MCHx_O are controlled by the POEN, OAEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register.

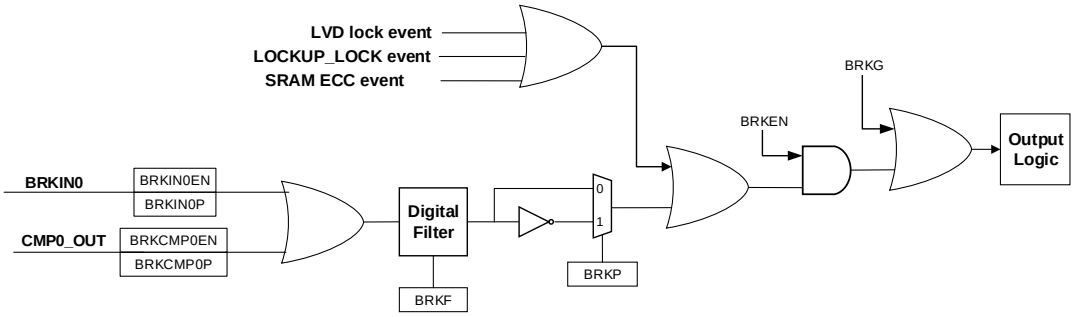
The break event is the result of logic ORed of all sources. The BREAK function can handle three types of event sources:

- External sources: coming from BRKIN0 input;
- System sources: LVD lock event, Cortex®-M33 LOCKUP_LOCK event or SRAM ECC error event;
- On-chip peripheral events: input by comparator output.

BREAK events can also be generated by software using BRKG bit in the TIMERx_SWEVG register.

Refer to ***Figure 12-97. BREAK function logic diagram***, BRKIN0 can select GPIO pins from the TRIGSEL module, which can select by ***Trigger selection for TIMER15_BRKIN register (TRIGSEL_TIMER15BRKIN)*** and ***Trigger selection for TIMER16_BRKIN register (TRIGSEL_TIMER16BRKIN)***.

Figure 12-97. BREAK function logic diagram

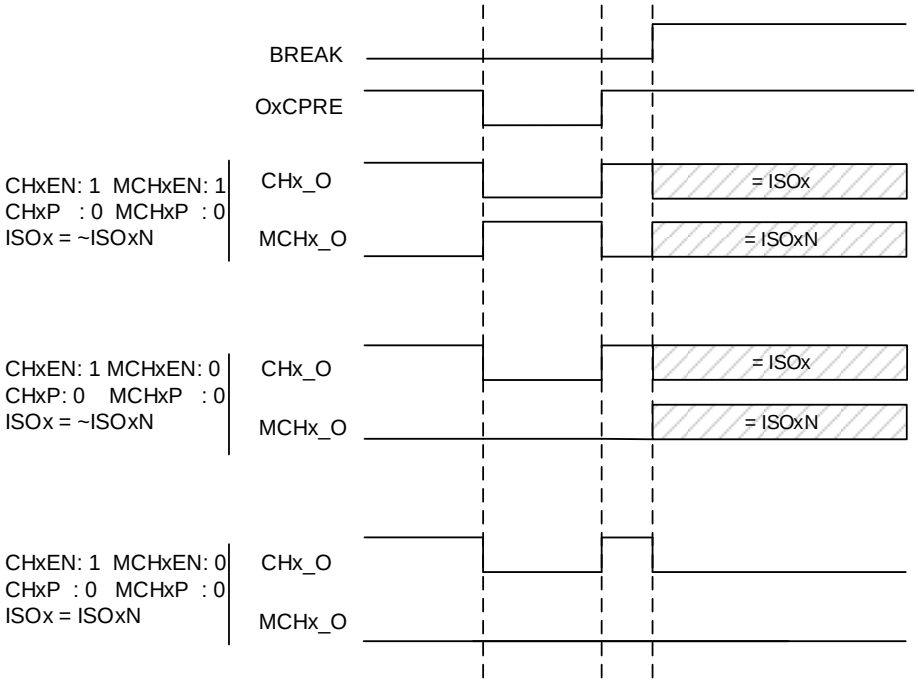


When a BREAK event occurs, the outputs are force at an inactive level, or at a predefined level (either active or inactive) after a deadtime duration.

When the MCHxMSEL = 2'b11 and a break occurs, the POEN bit is cleared asynchronously. As soon as POEN is 0, the level of the CHx_O and MCHx_O outputs are determined by the ISOx and ISOxN bits in the TIMERx_CTL1 register. If IOS = 0, the timer releases the enable output, otherwise, the enable output remains high. When IOS=1, the output behavior of the channel is shown in ***Figure 12-98. Output behavior of the channel in response to BREAK (the break input high active and IOS=1)***. The complementary outputs are first in the reset state, and then the dead time generator is reactivated to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead time.

Figure 12-98. Output behavior of the channel in response to BREAK (the break input

high active and IOS=1)



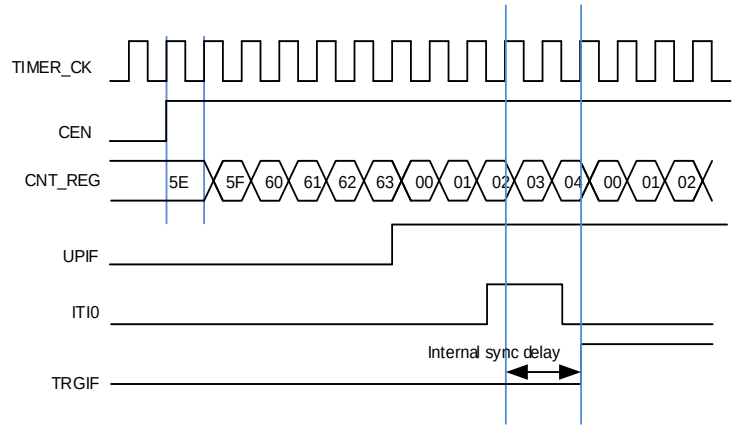
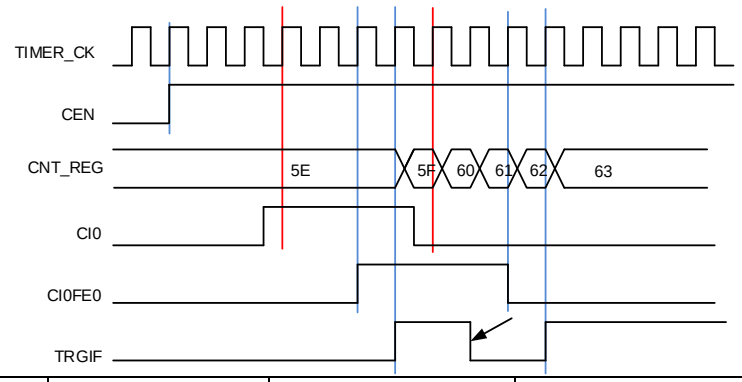
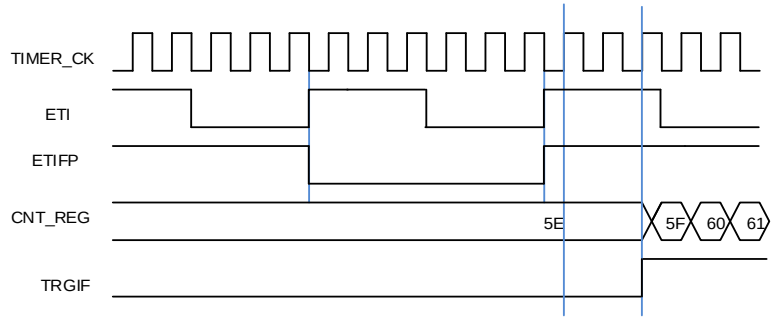
When a break occurs, the BRKIF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

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 and so on, which is selected by the TSCFGy[4:0] (y=3..6) in SYSCFG_TIMERxCFG(x=15,16).

Table 12-14. Slave mode example table

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[4:0] y=3: restart mode y=4: pause mode y=5: event mode y=6: external clock mode 0	TSCFGy[4:0] 00000:Mode disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 10010: MCI0FEM0	If you choose the CIxFEx(x=0, 1) or MCIxFEMx(x=0), configure the CHxP, MCHxP and MCHxFP for the polarity selection and inversion.	For the ITIx no filter and prescaler can be used. For the CIx/ MCIx, configure Filter by CHxCAPFLT/ MCHxCAPFLT, no prescaler can be used.
Exam1	**Restart mode** The counter can be clear and restart when a rising trigger input.	TSCFG3[4:0] 5'b00001, ITI0 is the selection.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 12-99. Restart mode  			
Exam2	**Pause mode** The counter can be paused when the trigger input is low.	TSCFG4[4:0] =5'b00110, CI0FE0 is the selection.	TI0S=0.(Non-xor) [MCHxP=0, CH0P=0] no inverted. Capture will be sensitive to the rising edge only.	Filter is bypass in this example.
	Figure 12-100. Pause mode  			
Exam3	**Event mode** The counter will start to count when a rising edge of trigger input comes.	TSCFG5[4:0] =5'b01000, 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 12-101. Event mode  			

Single pulse mode

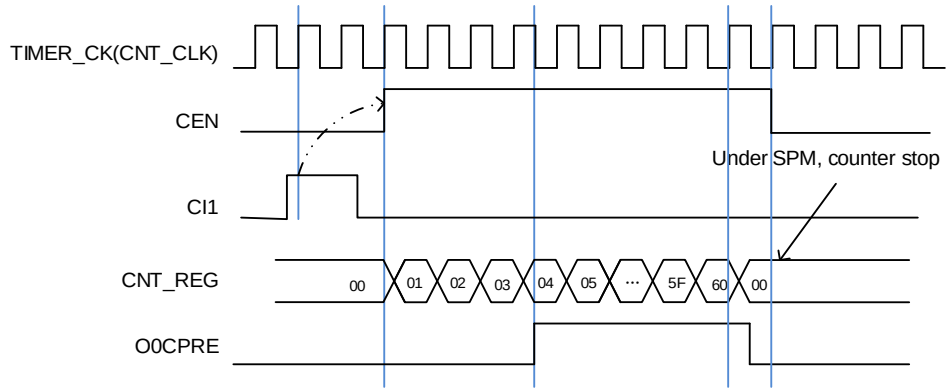
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped automatically when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL or MCHxCOMCTL bits.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held. If the CEN bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE/ MOxCPRE signals will change to, as the compare match event occurs without taking the comparison result into account.

Single pulse mode is also applicable to composite PWM mode (CHxCPWMEN = 1'b1 and CHxMS[2:0] = 3'b000).

Figure 12-102. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timers interconnection

Please refer to ***Advanced timer (TIMERx, x=0, 7)Timers interconnection***.

Counter synchronization and counter initial direction and value refresh

Please refer to ***Advanced timer (TIMERx, x=0, 7)Counter synchronization and counter initial direction and value refresh.***

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 a 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®-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

12.3.5. Register definition (TIMERx, x=15,16)

TIMER15 base address: 0x4001 5000

TIMER16 base address: 0x4000 1800

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 ratio between CK_TIMER (the timer clock) and DTS (the dead time and sampling clock) which is

		used for the dead time generator and the digital filter. 00: f_{DTS}= f_{CK_TIMER} 01: f_{DTS}= f_{CK_TIMER} /2 10: f_{DTS}= f_{CK_TIMER} /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter align mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when the counter is counting down, compare interrupt flag of channels can be set. 10: Center-aligned and counting up assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Only when the counter is counting up, compare interrupt flag of channels can be set. 11: Center-aligned and counting up/down assert mode. The counter counts in center-aligned mode and channel is configured in output mode (CHxMS = 3'b000 in TIMERx_CHCTL0 register). Both when the counter is counting up and counting down, compare interrupt flag of channels can be set. After the counter is enabled, these bits cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generates an update interrupt or a DMA request: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Only counter overflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable This bit is used to enable or disable the update event generation.

		0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: – The UPG bit is set. – The counter generates an overflow or underflow event. – The slave mode controller generates an update event. 1: Update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UG bit is set or the slave mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode. While in event mode, the hardware can set the CEN bit automatically.

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
CCUC[2:1]		Reserved				MMC[3]	Reserved								
rw						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC[0]	Reserved	CCSE
					rw	rw	rw	rw	rw			rw	rw		rw

Bits	Fields	Descriptions
31:30	CCUC[2:1]	Commutation control shadow register update control Refer to CCUC [0] description.
29:26	Reserved	Must be kept at reset value.
25	MMC[3]	Master mode control Refer to MMC[2:0] description.
24:11	Reserved	Must be kept at reset value.
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of multi mode 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_CCHP0 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_CCHP0 register is 00.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 0000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 0001: Enable. This mode is used to start several timers at the same time or control a slave timer to be enabled in a period. In this mode, the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 0010: Update. In this mode, the master mode controller selects the update event as TRGO. 0011: Capture/compare pulse. In this mode, the master mode controller generates a TRGO pulse when a capture or a compare match occurs in channel 0. 0100: Compare. In this mode, the master mode controller selects the O0CPRE signal as TRGO. 0101: Compare. In this mode, the master mode controller selects the O1CPRE signal as TRGO. 1001: In this mode, the master mode controller selects the soft synchronization event signal (generated by setting the SWSYNCG to 1) as TRGO. Others: Reserved.
3	DMAS	DMA request source selection 0: DMA request of CHx/MCHx is sent when capture/compare event occurs. 1: DMA request of channel CHx/MCHx is sent when update event occurs.
2	CCUC[0]	Commutation control shadow register update control The CCUC[2:1] and CCUC[0] field are used to control the commutation control shadow register update. When the commutation control shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are enabled (CCSE=1), the update control of the shadow registers with the CCUC[2:0] bit-field are shown as below: 000: The shadow registers update when CMTG bit is set. 001: The shadow registers update when CMTG bit is set or a rising edge of TRGI

		occurs. 100: The shadow registers update when the counter generates an overflow event. 101: The shadow registers update when the counter generates an underflow event. 110: The shadow registers update when the counter generates an overflow or underflow event. When a channel does not have a complementary output, this bit has no effect. **Note**: When CCUC[2:0] bit-field are set to 100, the update of the shadow registers also considers the value the CCUSEL bit in the TIMERx_CFG register.
1	Reserved	Must be kept at reset value.
0	CCSE	Commutation control shadow enable 0: The shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are disabled. 1: The shadow registers (for CHxEN, MCHxEN and CHxCOMCTL bits) are enabled. After these bits have been written, they are updated when commutation event comes. When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	MSM	Master-slave mode This bit can be used to synchronize the selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected. 0: Master-slave mode disabled 1: Master-slave mode enabled
6:0	Reserved	Must be kept at reset value.

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		CH1COMADDIE	CH0COMADDIE	Reserved			MCH0DEN	Reserved			MCH0IE	Reserved			
		rw	rw				rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	CMTDEN	Reserved		CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	Reserved		CH1IE	CH0IE	UPIE
	rw	rw			rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CH1COMADDIE	Channel 1 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode.
28	CH0COMADDIE	Channel 0 additional compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used in composite PWM mode.
27:25	Reserved	Must be kept at reset value.
24	MCH0DEN	Multi mode channel 0 capture/compare DMA request enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH0SEL[1:0] = 2'b00).
23:21	Reserved	Must be kept at reset value.
20	MCH0IE	Multi mode channel 0 capture/compare interrupt enable 0: Disabled 1: Enabled **Note:** This bit just used for channel input and output independent mode (when MMCH0SEL[1:0] = 2'b00).
19:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: Disabled 1: Enabled
13	CMTDEN	Commutation DMA request enable 0: Disabled 1: Enabled

12:11	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		CH1COMADDIF	CH0COMADDIF	Reserved			MCH0OF	Reserved			MCH0IF	Reserved			
		rc_w0	rc_w0				rc_w0				rc_w0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					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
31:30	Reserved	Must be kept at reset value.
29	CH1COMADDIF	Channel 1 additional compare interrupt flag. Refer to CH0COMADDIF description.
28	CH0COMADDIF	Channel 0 additional compare interrupt flag. This flag is set by hardware and cleared by software. If channel 0 is in output mode, this flag is set when a compare event occurs. 0: No channel 0 output compare interrupt occurred 1: Channel 0 output compare interrupt occurred **Note:** This flag just used in composite PWM mode.
27:25	Reserved	Must be kept at reset value.
24	MCH0OF	Multi mode channel 0 over capture flag When multi mode channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while MCH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
23:21	Reserved	Must be kept at reset value.
20	MCH0IF	Multi mode channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If multi mode channel 0 is in input mode, this flag is set when a capture event occurs. If multi mode channel 0 is in output mode, this flag is set when a compare event occurs. If multi mode channel 0 is set to input mode, this bit will be reset by reading TIMERx_MCH0CV. 0: No multi mode channel 0 capture/compare interrupt occurred 1: Multi mode channel 0 capture/compare interrupt occurred
19:11	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 This flag is set by hardware when the break input is active, and cleared by software if the BREAK input is not at active level. 0: No active level on BREAK input has been detected. 1: An active level on BREAK input has been detected.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller is enabled in pause mode, either edge of the trigger input can generate a trigger event. 0: No trigger event occurred 1: Trigger interrupt occurred
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when the commutation event of channel occurs, and cleared by software. 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4:3	Reserved	Must be kept at reset value.
2	CH1IF	Channel 1 capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 capture/compare interrupt flag This flag is set by hardware and cleared by software. If channel 0 is in input mode, this flag is set when a capture event occurs. If channel 0 is in output mode, this flag is set when a compare event occurs. If channel 0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14
Reset value: 0x0000 0000

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

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

Reserved	CH1COMADDG	CH0COMADDG	Reserved	MCH0G	Reserved
	w	w		w	

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
31:30	Reserved	Must be kept at reset value.
29	CH1COMADDG	Channel 1 additional compare event generation. Refer to CH0COMADDG description.
28	CH0COMADDG	Channel 0 additional compare event generation. This bit is set by software to generate a compare event in channel 0 additional, it is automatically cleared by hardware. When this bit is set, the CH0COMADDIF flag will be set, and the corresponding interrupt will be sent if enabled. 0: No generate a channel 0 additional compare event 1: Generate a channel 0 additional compare event **Note:** This bit just used in composite PWM mode.
27:21	Reserved	Must be kept at reset value.
20	MCH0G	Multi mode channel 0 capture or compare event generation. This bit is set by software to generate a capture or compare event in multi mode channel 0, it is automatically cleared by hardware. When this bit is set, the MCH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if multi mode channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_MCH0CV register, and the MCH0OF flag is set if the MCH0IF flag has been set. 0: No generate a multi mode channel 0 capture or compare event 1: Generate a multi mode channel 0 capture or compare event
19:8	Reserved	Must be kept at reset value.
7	BRKG	BREAK event generation This bit is set by software to generate an event and cleared by hardware automatically. When this bit is set, the POEN bit will be cleared and BRKIF flag will be set, related interrupt can occur if enabled. 0: No generate a BREAK event 1: Generate a BREAK event
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register will be set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event

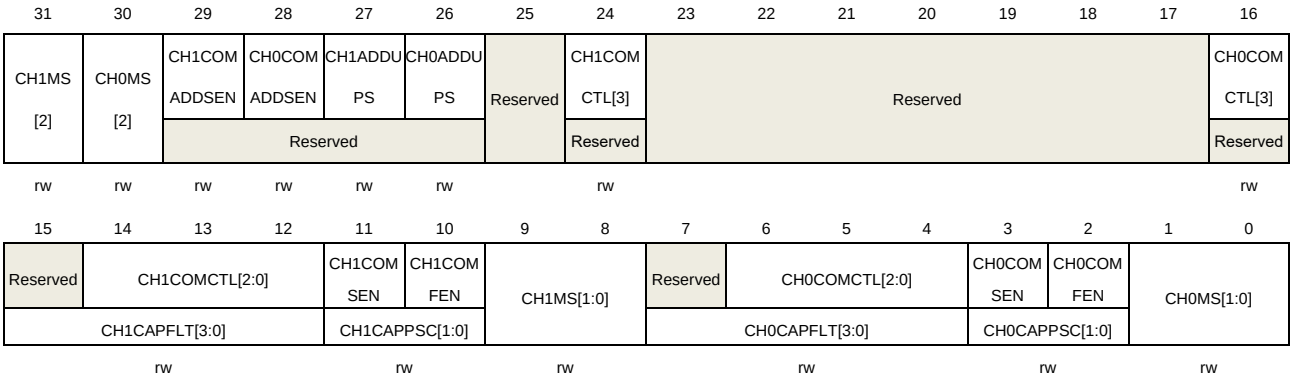
		1: Generate a trigger event
5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, MCHxEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel commutation update event
4:3	Reserved	Must be kept at reset value.
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, and the corresponding interrupt or DMA request will be sent if enabled. In addition, if channel 0 is configured in input mode, the current value of the counter is captured to TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag has been set. 0: No generate a channel 0 capture or compare event 1: Generate a channel 0 capture or compare event
0	UPG	Update event generation This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared if the up counting mode is selected. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Refer to CH1MS[1:0]description
30	CH0MS[2]	Channel 0 I/O mode selection Refer to CH0MS[1:0] description
29	CH1COMADDSEN	Channel 1 additional compare output shadow enable Refer to CH0COMADDSEN description.
28	CH0COMADDSEN	Channel 0 additional compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0COMV_ADD register which updates at each update event will be enabled. 0: Channel 0 additional compare output shadow disabled 1: Channel 0 additional compare output shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000.
27	CH1ADDUPS	Channel 1 additional update source 0: TIMERx_CH1COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH1COMV_ADD register is updated when the counter matches the value of CH1VAL.
26	CH0ADDUPS	Channel 0 additional update source 0: TIMERx_CH0COMV_ADD register is updated when an update event occurs. 1: TIMERx_CH0COMV_ADD register is updated when the counter matches the value of CH0VAL.
25	Reserved	Must be kept at reset value.
24	CH1COMCTL[3]	Channel 1 compare output control Refer to CH0COMCTL[2:0] description.
23:17	Reserved	Must be kept at reset value.
16	CH0COMCTL[3]	Channel 0 compare output control Refer to CH0COMCTL[2:0] description.
15	Reserved	Must be kept at reset value.
14:12	CH1COMCTL[2:0]	Channel 1 compare output control Refer to CH0COMCTL[2:0] 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. The CH1MS[2:0] bit-field is writable only when the channel is not active (When MCH1MSEL[1:0] = 2'b00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH1MSEL[1:0] = 2'b11, the CH1EN and MCH1EN bits in TIMERx_CHCTL2 register are reset). 000: Channel 1 is configured as output. 001: Channel 1 is configured as input, IS1 is connected to CI1FE1. 010: Channel 1 is configured as input, IS1 is connected to CI0FE1. 011: Channel 1 is configured as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=15,16) register). 100~111: Reserved.
7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control The CH0COMCTL[3] and CH0COMCTL[2:0] bit-field control the behavior of O0CPRE which drives CH0_O and MCH0_O. The active level of O0CPRE is high, while the active level of CH0_O and MCH0_O depend on CH0P and MCH0P bits. **Note**: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, This bit-field controls the behavior of O0CPRE which drives CH0_O and MCH0_O. The active level of O0CPRE is high, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 0000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 0010: Clear the channel output on match. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 0011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 0100: Force low. O0CPRE is forced low level. 0101: Force high. O0CPRE is forced high level. 0110: PWM mode 0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV, otherwise it is inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV, otherwise it is active. 0111: PWM mode 1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV, otherwise it is active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV, otherwise it is inactive. 1000~1111: Reserved. **Note**: In the composite PWM mode (CH0CPWMEN = 1'b1 and CH0MS = 3'b000),

		the PWM signal output in channel 0 is composited by TIMERx_CH0CV and TIMERx_CH0COMV_ADD. Please refer to ***Composite PWM mode*** for more details. If configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from “Timing” mode to “PWM” mode or the result of the comparison changes. When the outputs of CH0 and MCH0 are complementary, this bit-field is preloaded. If CCSE =1, this bit-field will only be updated when a channel commutation event is generated. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000 (compare mode).
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register which updates at each update event will be enabled. 0: Channel 0 output compare shadow disabled 1: Channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and CH0MS bit-field is 000.
2	CH0COMFEN	Channel 0 output compare fast enable 0: Channel 0 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and CH0_O is set to the compare level.The minimum delay from an edge on the result of the comparison to activate CH0_O output is 5 clock cycles. 1: Channel 0 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and CH0_O is set to the compare level.The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. The CH0MS[2:0] bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2’b00, the CH1EN bit in TIMERx_CHCTL2 register is reset; when MCH0MSEL[1:0] = 2’b11, the CH0EN and MCH0EN bits in TIMERx_CHCTL2 register are reset). 000: Channel 0 is configured as output. 001: Channel 0 is configured as input, IS0 is connected to CI0FE0. 010: Channel 0 is configured as input, IS0 is connected to CI1FE0. 011: Channel 0 is configured as input, IS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=15,16) register). 100: Channel 0 is configured as input, IS0 is connected to MCI0FE0.

101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	CH1MS[2]	Channel 1 I/O mode selection Same as output compare mode.
30	CH0MS[2]	Channel 0 I/O mode selection Same as output compare mode.
29:16	Reserved	Must be kept at reset value.
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description.
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description.
9:8	CH1MS[1:0]	Channel 1 I/O mode selection Same as output compare mode.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8. 1010: $f_{SAMP}=f_{DTS}/16$, N=5. 1011: $f_{SAMP}=f_{DTS}/16$, N=6. 1100: $f_{SAMP}=f_{DTS}/16$, N=8. 1101: $f_{SAMP}=f_{DTS}/32$, N=5. 1110: $f_{SAMP}=f_{DTS}/32$, N=6. 1111: $f_{SAMP}=f_{DTS}/32$, N=8.
3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges.

		11: Capture is done every 8 channel input edges.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection Same as output compare mode.

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CH1PERFOREN	CH0PERFOREN	Reserved											
		rw	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								MCH1P	Reserved	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
								rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CH1PERFOREN	Channel 1 in the composite PWM mode, the period point match moment O1CPRE level setting Refer to CH0PERFOREN description.
28	CH0PERFOREN	Channel 0 in the composite PWM mode, the period point match moment O0CPRE level setting 0: disable 1: In the composite PWM 0 mode, the level will be forced to high at the time of period point matching; In the composite PWM 1 mode, the level at the matching time of the period point is forced to low. **Note:** In the central alignment mode, the period point is the counter underflow time; In the edge-aligned mode, the period point is the counter flow moment.
27:8	Reserved	Must be kept at reset value.
7	MCH1P	Multi mode channel 1 output polarity Refer to MCH0P description.
6	Reserved	Must be kept at reset value.
5	CH1P	Channel 1 capture/compare polarity Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare enable Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity

		When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, this bit specifies the CH0_ON output signal polarity. 0: Multi mode channel 0 output active high 1: Multi mode channel 0 output active low When CH0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CH0. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
2	MCH0EN	Multi mode channel 0 capture/compare enable When multi mode channel 0 is configured in output mode, setting this bit enables CH0_ON signal in active state. When multi mode channel 0 is configured in input mode, setting this bit enables the capture event in multi mode channel 0. 0: Multi mode channel 0 disabled 1: Multi mode channel 0 enabled
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, these bits specifie the channel 0 input signal's polarity. [MCH0P, CH0P] will select the active trigger or capture polarity for channel 0 input signals. 00: Channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will not be inverted. 01: Channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And channel 0 input signal will be inverted. 10: Reserved. 11: Noninverted/both channel 0 input signal's edges. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.
0	CH0EN	Channel 0 capture/compare enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel 0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 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
Reserved															

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

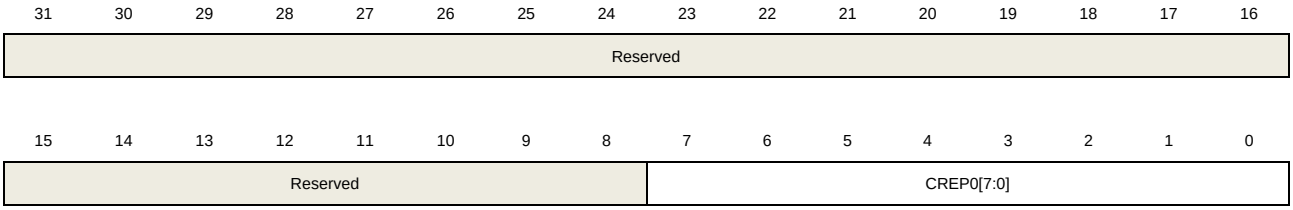
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30

Reset value: 0x0000 0000

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



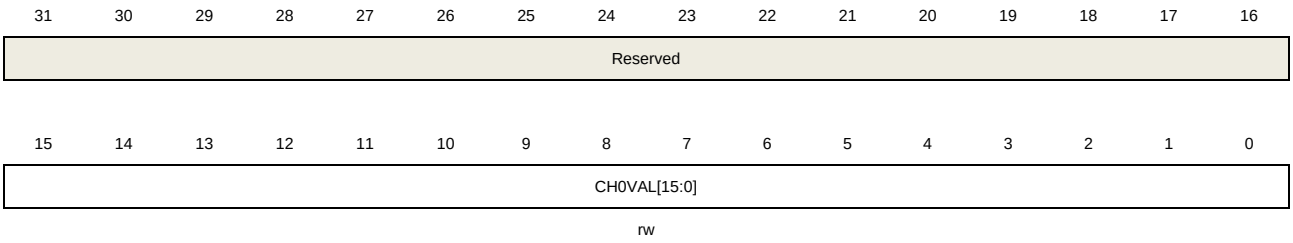
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is also affected by this bit-field when these shadow registers are enabled.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture/compare value of channel 0 When channel 0 is configured in input mode, this bit-field indicates the counter value

at the last capture event. And this bit-field is read-only.
When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1VAL[15:0]	Capture/compare value of channel 1 When channel 1 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register 0 (TIMERx_CCHP0)

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												BRKF[3:0]			
												rw			

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
31:20	Reserved	Must be kept at reset value.
19:16	BRKF[3:0]	BREAK input signal filter An event counter is used in the digital filter, in which a transition on the output occurs

after N input events. This bit-field specifies the frequency used to sample BREAK input signal and the length of the digital filter applied to BREAK.
0000: Filter disabled. BREAK act asynchronously, N=1
0001: $f_{SAMP} = f_{CK_TIMER}$, N=2
0010: $f_{SAMP} = f_{CK_TIMER}$, N=4
0011: $f_{SAMP} = f_{CK_TIMER}$, N=8
0100: $f_{SAMP} = f_{DTS}/2$, N=6
0101: $f_{SAMP} = f_{DTS}/2$, N=8
0110: $f_{SAMP} = f_{DTS}/4$, N=6
0111: $f_{SAMP} = f_{DTS}/4$, N=8
1000: $f_{SAMP} = f_{DTS}/8$, N=6
1001: $f_{SAMP} = f_{DTS}/8$, N=8
1010: $f_{SAMP} = f_{DTS}/16$, N=5
1011: $f_{SAMP} = f_{DTS}/16$, N=6
1100: $f_{SAMP} = f_{DTS}/16$, N=8
1101: $f_{SAMP} = f_{DTS}/32$, N=5
1110: $f_{SAMP} = f_{DTS}/32$, N=6
1111: $f_{SAMP} = f_{DTS}/32$, N=8
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

15 POEN

Primary output enable
This bit is set by software or automatically set by hardware depending on the OAEN bit. It is cleared asynchronously by hardware as soon as the break input is active. When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and MCHx_O) if the corresponding enable bits (CHxEN, MCHxEN in TIMERx_CHCTL2 register) have been set.
0: Channel outputs are disabled or forced to idle state.
1: Channel outputs are enabled.

14 OAEN

Output automatic enable
This bit specifies whether the POEN bit can be set automatically by hardware.
0: POEN cannot be set by hardware.
1: POEN can be set by hardware automatically at the next update event, if the break input is not active.
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

13 BRKP

BREAK input signal polarity
This bit specifies the polarity of the BREAK input signal.
0: BREAK input active low
1: BREAK input active high
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

12 BRKEN

BREAK input signal enable

		This bit can be set to enable the BREAK input signal 0: BREAK input disabled 1: BREAK input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 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 12-13. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***. 0: “off-state” disabled. If the CHxEN or MCHxEN bit is reset, the corresponding channel is output disabled. 1: “off-state” enabled. If the CHxEN or MCHxEN bit is reset, the corresponding channel is “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 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 12-13. Complementary outputs controlled by parameters (MCHxMSEL =2’b11)***. 0: “off-state” disabled. If the CHxEN/ MCHxEN bits are both reset, the channels are output disabled. 1: “off-state” enabled. No matter the CHxEN/ MCHxEN bits, the channels are “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP0 register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-field specifies the write protection property of registers. 00: Protect disabled. No write protection. 01: PROT mode 0. The ISOx / ISOxN bits in TIMERx_CTL1 register, the BRKEN/ BRKP/OAEN/DTCFG bits in TIMERx_CCHP0 register are writing protected. 10: PROT mode 1. In addition to the registers in PROT mode 0, the CHxP/MCHxP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) the ROS/IOS bits in TIMERx_CCHP0 register are writing protected. 11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN bits in TIMERx_CHCTL0 and TIMERx_MCHCTL0 registers (if the related channel is configured in output) are writing protected. This bit-field can be written only once after the system reset. Once the TIMERx_CCHP0 register has been written, this bit-field will be writing protected.
7:0	DTCFG[7:0]	Dead time configuration

This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between the value of DTCFG and the duration of dead-time is as follow:

DTCFG[7:5] = 3'b0xx: DT value = DTCFG[7:0] * t_{DT}, t_{DT} = t_{DTS}.

DTCFG[7:5] = 3'b10x: DT value = (64+DTCFG[5:0]) * t_{DT}, t_{DT} = t_{DTS}*2.

DTCFG[7:5] = 3'b110: DT value = (32+DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS}*8.

DTCFG[7:5] = 3'b111: DT value = (32+DTCFG[4:0]) * t_{DT}, t_{DT} = t_{DTS}*16.

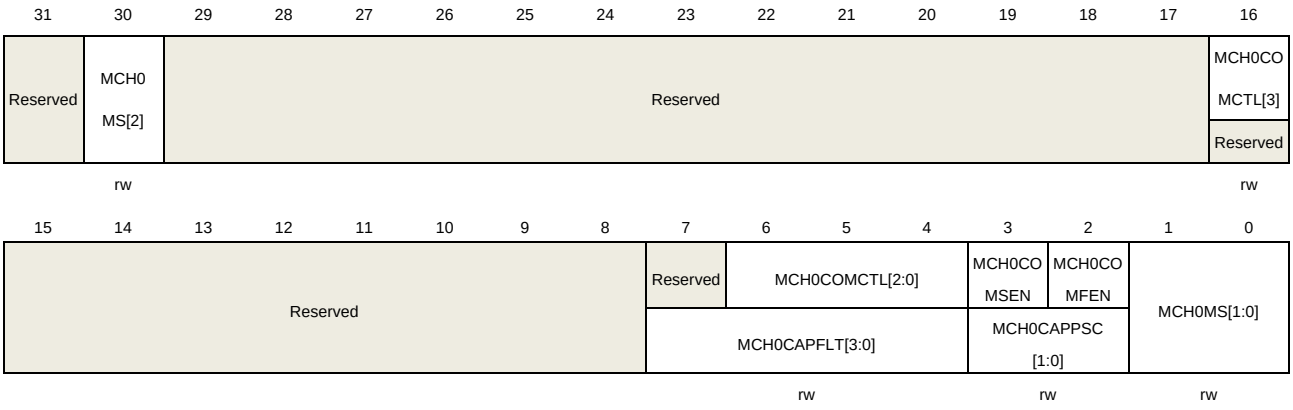
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Multi mode channel control register 0 (TIMERx_MCHCTL0)

Address offset: 0x48

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:17	Reserved	Must be kept at reset value.
16	MCH0COMCTL[3]	Multi mode channel 0 compare output control. Refer to MCH0COMCTL[2:0] description.
15:7	Reserved	Must be kept at reset value.
6:4	MCH0COMCTL[2:0]	Multi mode channel 0 output compare control When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b00, the MCH0COMCTL[3] and MCH0COMCTL[2:0] bit-field control the behavior of MO0CPRE which drives MCH0_O. The active level of MO0CPRE is high, while the active level of MCH0_O depends on MCH0FP[1:0] bits. **Note**: When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, the CH0COMCTL[2:0] bit-field controls the behavior of

		O0CPRE which drives CH0_O and MCH0_O, while the active level of CH0_O and MCH0_O depends on CH0P and MCH0P bits. 0000: Timing mode. The MO0CPRE signal keeps stable, independent of the comparison between the register TIMERx_MCH0CV and the counter TIMERx_CNT. 0001: Set the channel output on match. MO0CPRE signal is forced high when the counter matches the output compare register TIMERx_MCH0CV. 0010: Clear the channel output on match. MO0CPRE signal is forced low when the counter matches the output compare register TIMERx_MCH0CV. 0011: Toggle on match. MO0CPRE toggles when the counter matches the output compare register TIMERx_MCH0CV. 0100: Force low. MO0CPRE is forced low level. 0101: Force high. MO0CPRE is forced high level. 0110: PWM mode 0. When counting up, MO0CPRE is active as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is inactive. When counting down, MO0CPRE is inactive as long as the counter is larger than TIMERx_MCH0CV, otherwise it is active. 0111: PWM mode 1. When counting up, MO0CPRE is inactive as long as the counter is smaller than TIMERx_MCH0CV, otherwise it is active. When counting down, MO0CPRE is active as long as the counter is larger than TIMERx_MCH0CV, otherwise it is inactive. 1000~1111: Reserved. If configured in PWM mode, the MO0CPRE level changes only when the output compare mode switches from “Timing” mode to “PWM” mode or the result of the comparison changes. When the outputs of CH0 and MCH0 are complementary, this bit-field is preloaded. If CCSE =1, this bit-field will only be updated when a channel commutation event is generated. This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and MCH0MS bit-field is 00 (compare mode).
3	MCH0COMSEN	Multi mode channel 0 output compare shadow enable When this bit is set, the shadow register of TIMERx_MCH0CV register which updates at each update event will be enabled. 0: Multi mode channel 0 output compare shadow disabled 1: Multi mode channel 0 output compare shadow enabled The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 and MCH0MS bit-field is 00.
2	MCH0COMFEN	Multi mode channel 0 output compare fast enable 0: Multi mode channel 0 output quickly compare disable.Only the valid edge of the comparison result will generate a comparison match, and MCH0_O is set to the

		compare level.The minimum delay from an edge on the result of the comparison to activate MCH0_O output is 5 clock cycles. 1: Multi mode channel 0 output quickly compare enable.Both the valid edge of the trigger input signal and the comparison result will generate a comparison match, and MCH0_O is set to the compare level.The minimum delay from an edge on the trigger input to activate MCH0_O output is 3 clock cycles.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (When MCH0MSEL[1:0] = 2'b00, MCH0EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 0 is configured as output. 001: Multi mode channel 0 is configured as input, MIS0 is connected to MCI0FEM0. 010: Reserved. 011: Multi mode channel 0 is configured as input, MIS0 is connected to ITS, this mode is working only if an internal trigger input is selected (through TSCFG15[4:0] bit-field in SYSCFG_TIMERxCFG2(x=15,16) register). 100: Multi mode channel 0 is configured as input, MIS0 is connected to CI0FEM0. 101~111: Reserved.

Input capture mode:

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	MCH0MS[2]	Multi mode channel 0 I/O mode selection Refer to MCH0MS[1:0] description.
29:8	Reserved	Must be kept at reset value.
7:4	MCH0CAPFLT[3:0]	Multi mode channel 0 input capture filter control. An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample MCI0 input signal and the length of the digital filter applied to MCI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1. 0001: $f_{SAMP}=f_{CK_TIMER}$, N=2. 0010: $f_{SAMP}= f_{CK_TIMER}$, N=4. 0011: $f_{SAMP}= f_{CK_TIMER}$, N=8. 0100: $f_{SAMP}=f_{DTS}/2$, N=6. 0101: $f_{SAMP}=f_{DTS}/2$, N=8. 0110: $f_{SAMP}=f_{DTS}/4$, N=6. 0111: $f_{SAMP}=f_{DTS}/4$, N=8. 1000: $f_{SAMP}=f_{DTS}/8$, N=6. 1001: $f_{SAMP}=f_{DTS}/8$, N=8. 1010: $f_{SAMP}=f_{DTS}/16$, N=5. 1011: $f_{SAMP}=f_{DTS}/16$, N=6. 1100: $f_{SAMP}=f_{DTS}/16$, N=8.

		1101: $f_{SAMP}=f_{DTS}/32$, N=5. 1110: $f_{SAMP}=f_{DTS}/32$, N=6. 1111: $f_{SAMP}=f_{DTS}/32$, N=8.
3:2	MCH0CAPPSC[1:0]	Multi mode channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when MCH0EN bit in TIMERx_CHCTL2 register is cleared. 00: Prescaler disabled, capture is done on each channel input edge. 01: Capture is done every 2 channel input edges. 10: Capture is done every 4 channel input edges. 11: Capture is done every 8 channel input edges.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection Same as output compare mode

Multi mode channel control register 2 (TIMERx_MCHCTL2)

Address offset: 0x50

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														MCH0FP[1:0]	
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	MCH0FP[1:0]	Multi mode channel 0 capture/compare free polarity When multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b00, these bits specifie the multi mode channel 0 output signal polarity. 00: Multi mode channel 0 active high 01: Multi mode channel 0 active low 10: Reserved. 11: Reserved. When multi mode channel 0 is configured in input mode, these bits specifie the multi mode channel 0 input signal's polarity. MCH0FP[1:0] will select the active trigger or capture polarity for multi mode channel 0 input signals. 00: Multi mode channel 0 input signal's rising edge is the active signal for capture or trigger operation in slave mode. And multi mode channel 0 input signal will not be inverted. 01: Multi mode channel 0 input signal's falling edge is the active signal for capture or trigger operation in slave mode. And multi mode channel 0 input signal will be

inverted.
10: Reserved.
11: Noninverted/both multi mode channel 0 input signal's edges.
This bit cannot be modified when PROT[1:0] bit-field in TIMERx_CCHP0 register is 11 or 10.

Multi mode channel 0 capture/compare value register (TIMERx_MCH0CV)

Address offset: 0x54
Reset value: 0x0000 0000

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

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
MCH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	MCH0VAL[15:0]	Capture/compare value of multi mode channel 0. When multi mode channel 0 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When multi mode channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 0 additional compare value register (TIMERx_CH0COMV_ADD)

Address offset: 0x64
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0COMVAL_ADD	Additional compare value of channel 0

	[15:0]	When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1).

Channel 1 additional compare value register (TIMERx_CH1COMV_ADD)

Address offset: 0x68

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0030 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		CH1C PWMEN	CH0C PWMEN	Reserved						MCH0MSEL[1:0]		Reserved			
		rw	rw							rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		Reserved							
				rw		rw									

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CH1CPWMEN	Channel 1 composite PWM mode enable 0: Disabled

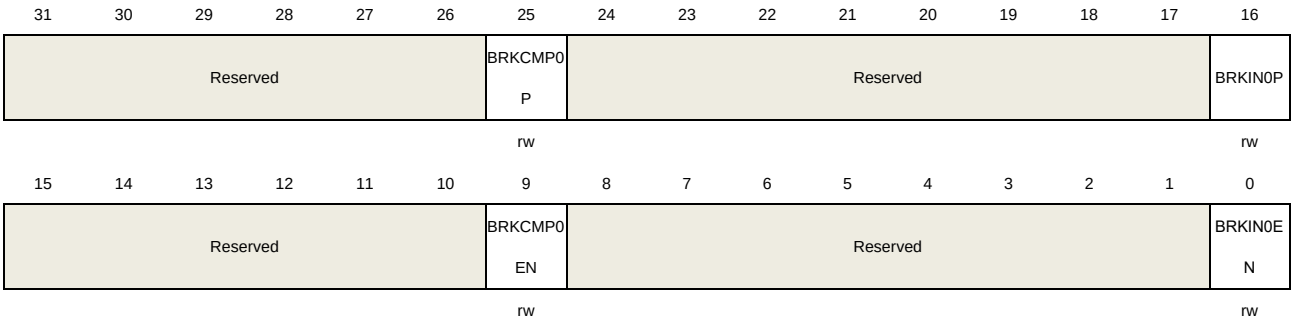
		1: Enabled
28	CH0CPWMEN	Channel 0 composite PWM mode enable 0: Disabled 1: Enabled
27:22	Reserved	Must be kept at reset value.
21:20	MCH0MSEL[1:0]	Multi mode channel 0 mode select 00: Independent mode, MCH0 is independent of CH0 01: Reserved 10: Reserved 11: Complementary mode, only the CH0 is valid for input, and the outputs of MCH0 and CH0 are complementary
19:12	Reserved	Must be kept at reset value.
11:10	CH1OMPSEL[1:0]	Channel 1 output match pulse select When the match events occur, this bit is used to select the output of O1CPRE which drives CH1_O. 00: The O1CPRE signal is output normal with the configuration of CH1COMCTL[2:0] bits. 01: Only when the counter is counting up, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O1CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
9:8	CH0OMPSEL[1:0]	Channel 0 output match pulse select When the match events occur, this bit is used to select the output of O0CPRE which drives CH0_O. 00: The O0CPRE signal is output normal with the configuration of CH0COMCTL[2:0] bits. 01: Only when the counter is counting up, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 10: Only when the counter is counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle. 11: Both when the counter is counting up and counting down, the O0CPRE signal is output a pulse when the match events occurs, and the pulse width is one CK_TIMER clock cycle.
7:0	Reserved	Must be kept at reset value.

TIMERx alternate function control register 0 (TIMERx_AFCTL0)

Address offset: 0x8C

Reset value: 0x0000 0001

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



Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	BRKCMP0P	BREAK CMP0 input polarity This bit is used to configure the CMP0 input polarity, and the specific polarity is determined by this bit and the BRKP bit. 0: CMP0 input signal will not be inverted (BRKP =0, the input signal is active low; BRKP =1, the input signal is active high) 1: CMP0 input signal will be inverted (BRKP=0, the input signal is active high; BRKP =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
24:17	Reserved	Must be kept at reset value.
16	BRKIN0P	BREAK BRKIN0 alternate function input polarity This bit is used to configure the BRKIN0 input polarity, and the specific polarity is determined by this bit and the BRKP bit. 0: BRKIN0 input signal will not be inverted (BRKP =0, the input signal is active low; BRKP =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRKP=0, the input signal is active high; BRKP =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
15:10	Reserved	Must be kept at reset value.
9	BRKCMP0EN	BREAK CMP0 enable 0: CMP0 input disabled 1: CMP0 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
8:1	Reserved	Must be kept at reset value.
0	BRKIN0EN	BREAK BRKIN0 alternate function input enable

0: BRKIN0 alternate function input disabled

1: BRKIN0 alternate function input enabled

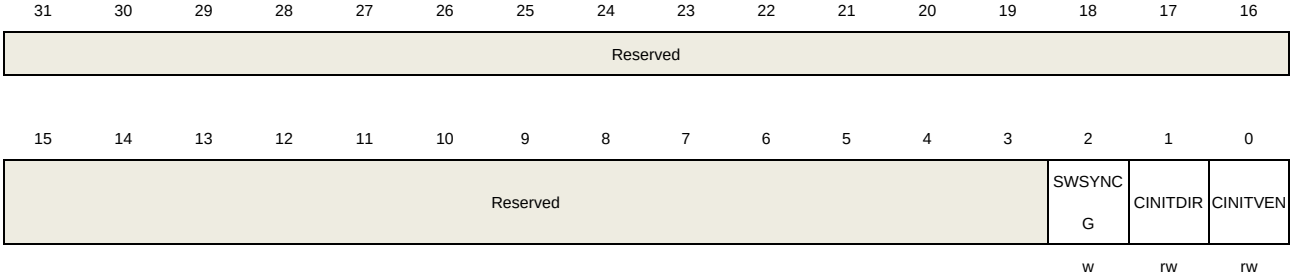
This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.

Counter initial control register (TIMERx_CINITCTL)

Address offset: 0xA4

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	SWSYNCG	Soft synchronization event generation This bit is set by software and cleared by hardware automatically. Reads this bit always return a 0. 0: No affect 1: Generate soft synchronization event
1	CINITDIR	Counter initial count direction 0: When the synchronization event occurs, the counter count down. 1: When the synchronization event occurs, the counter count up. This bit indicates the initial direction of the counter after a synchronization event occurs and the counter initial value is loaded from the TIMERx_CINITV register. **Note**: This bit is only used when the CAM[1:0] ≠ 00.
0	CINITVEN	Counter initial value register enable 0: Counter initial value register disable. The counter register can't load from the counter initial value register. 1: Counter initial value register enable. When a synchronization event (or soft synchronization event generated by setting the SWSYNCG bit) occurs, the counter register can load from the counter initial value register.

Counter initial value register (TIMERx_CINITV)

Address offset: 0xA8

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
CINITVAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CINITVAL [15:0]	Counter initial value This bits-field indicates the counter initial value. When the CINITVEN bit is 0, the counter register can't load the initial value which are set in CINITVAL bits-field. When the CINITVEN bit is 1, when the synchronization event occurs, the counter register can load the initial value which are set in CINITVAL bits-field.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing (R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b111111: transfer 63 times
7:6	Reserved	Must be kept at reset value.
5:0	DMATA[5:0]	DMA transfer access start address This field defines the start address of accessing the TIMERx_DMATB register by DMA. When the first access to the TIMERx_DMATB register is done, this bit-field

specifies the address just accessed. And then the address of the second access to the TIMERx_DMATB register will be (start address + 0x4).

6'b000000: TIMERx_CTL0

6'b000001: TIMERx_CTL1

…

In a word: start address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0xE4

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	DMATB[31:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address ranges from (start address) to (start address + transfer count * 4) will be accessed. The transfer count is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	CCUSEL	Commutation control shadow register update select This bit is valid only when the CCUC[2:0] bit-field are set to 100.

		0: The shadow registers update when the counter generates an overflow/ underflow event. 1: The shadow registers update when the counter generates an overflow/ underflow event and the repetition counter value is zero.
2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	OUTSEL	The output value selection bit This bit-field is set and reset by software. 1: If POEN bit and IOS bit are 0, the output is disabled. 0: No effect.

12.4. Basic timer (TIMERx, x=5,6)

12.4.1. Overview

The basic timer module(TIMER5/6) has a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate a DMA request and a TRGO to connect to DAC.

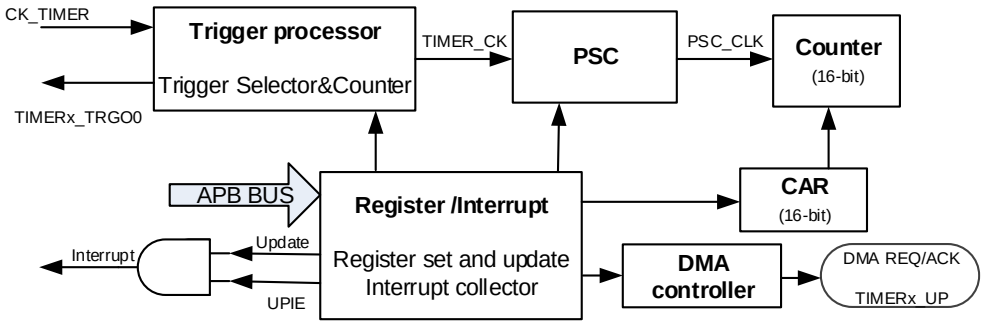
12.4.2. Characteristics

- Counter width: 16 bits (TIMER5/6).
- Source of count clock is internal clock only.
- Multiple counter modes: count up.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Auto reload function.
- Interrupt output or DMA request: update event.

12.4.3. Block diagram

Figure 12-103. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 12-103. Basic timer block diagram



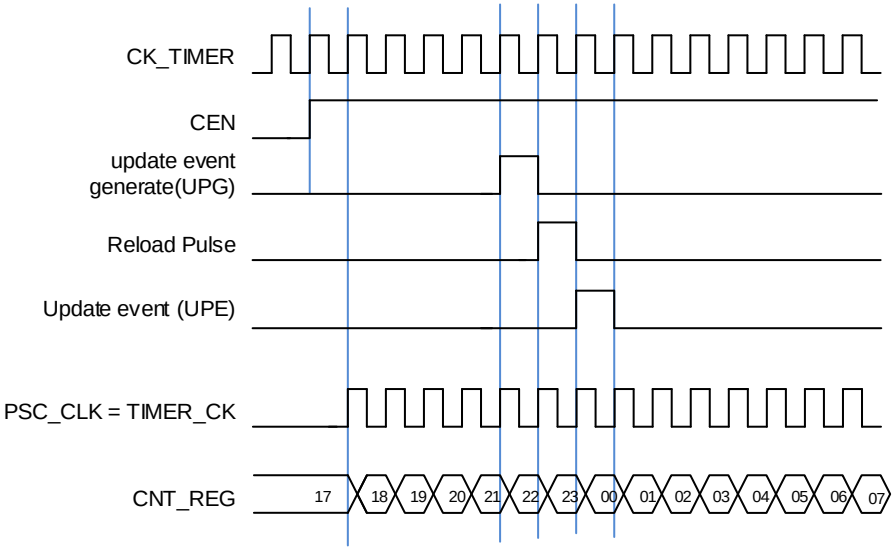
12.4.4. Function overview

Clock selection

The basic TIMER can only be clocked by the internal timer clock CK_TIMER, which is from the source named CK_TIMER in RCU

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

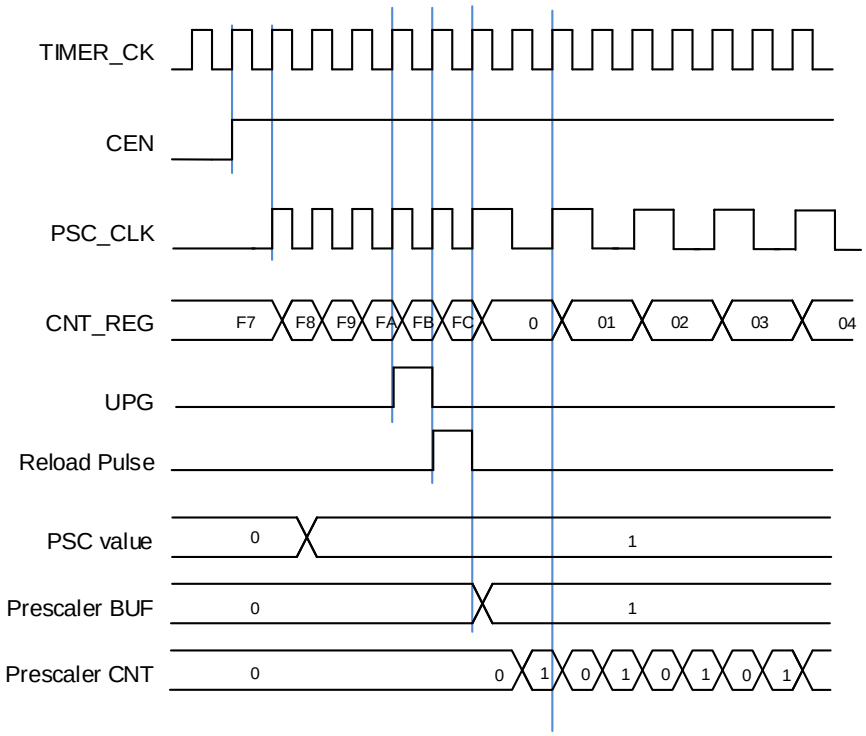
Figure 12-104. Normal mode, internal clock divided by 1



Prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed ongoing, but it is adopted at the next update event.

Figure 12-105. Counter timing diagram with prescaler division change from 1 to 2



Up counting mode

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 restarts from 0. The update event is generated each time when counter overflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

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

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

The following figures show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 12-106. Timing chart of up counting mode, PSC=0/2

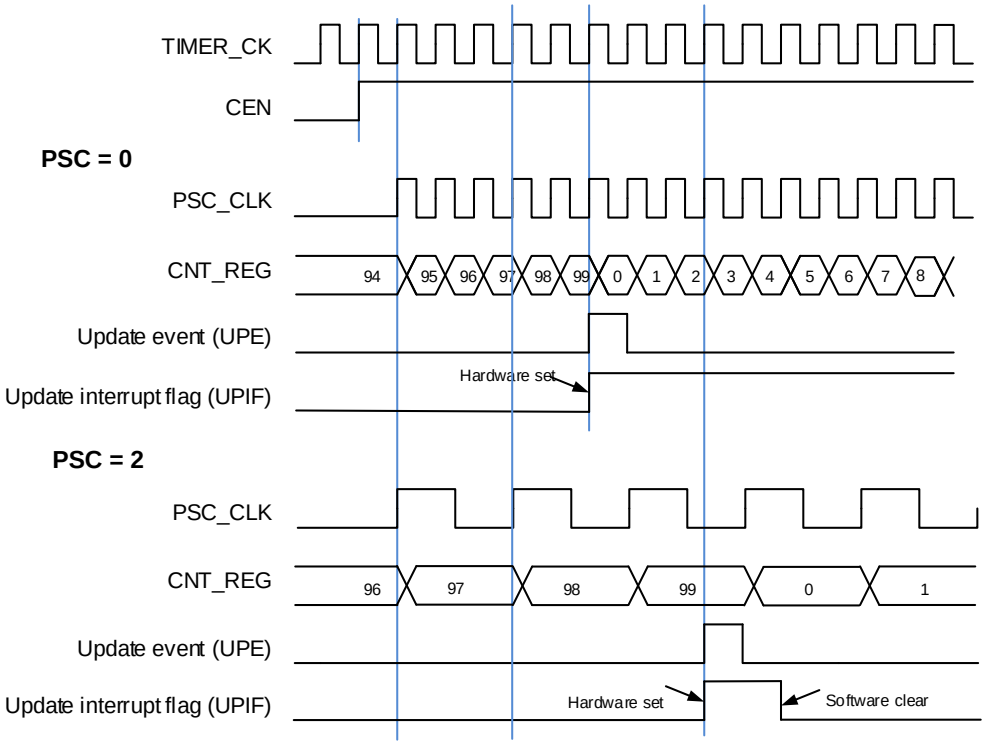
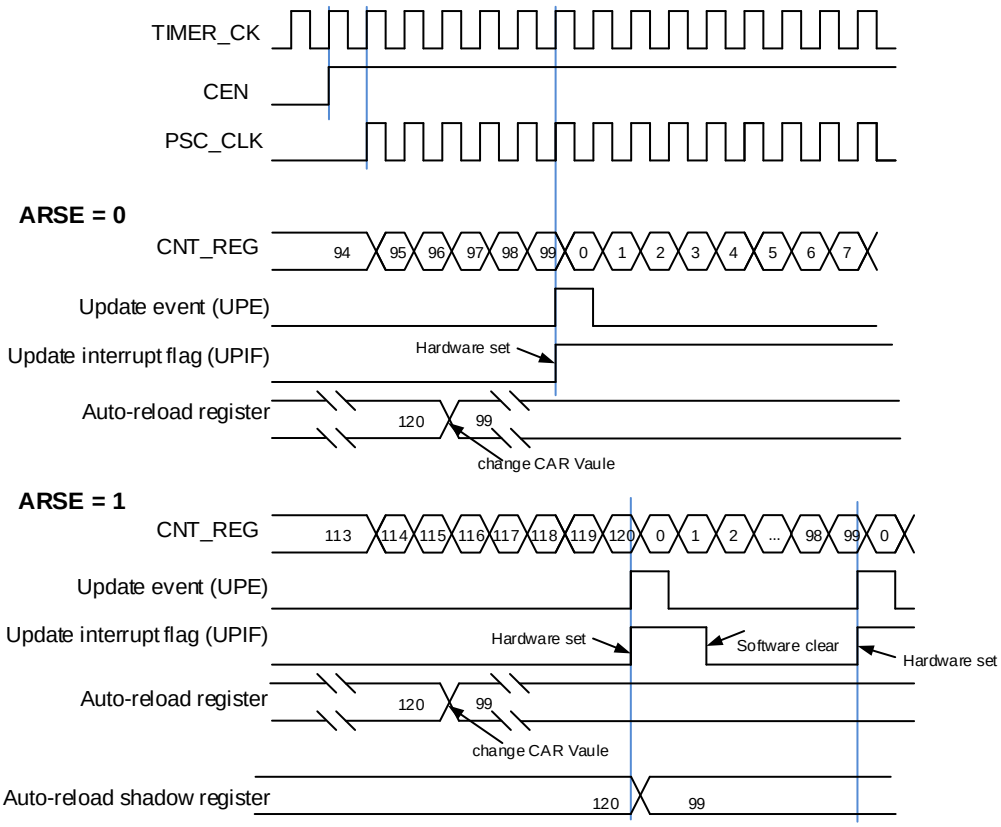


Figure 12-107. Timing chart of up counting mode, change TIMERx_CAR ongoing



Timer debug mode

When the Cortex®-M33 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

12.4.5. Registers definition (TIMERx, x=5,6)

TIMER5 base address: 0x4000 1000

TIMER6 base address: 0x4000 1400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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

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 is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and the counter stops at next update event.
2	UPS	Update source This bit is used to select the update event sources by software. 0: When enabled, any of the following events generates an update interrupt or a DMA request: – The UPG bit is set – The counter generates an overflow event – The slave mode controller generates an update event. 1: When enabled, only counter overflow generates an update interrupt or a DMA request.
1	UPDIS	Update disable. This bit is used to enable or disable the update event generation. 0: Update event enable. The update event is generated and the buffered registers are loaded with their preloaded values when one of the following events occurs: – The UPG bit is set

- The counter generates an overflow event
- The slave mode controller generates an update event.

1: Update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UG bit is set or the slave mode controller generates a hardware reset event.

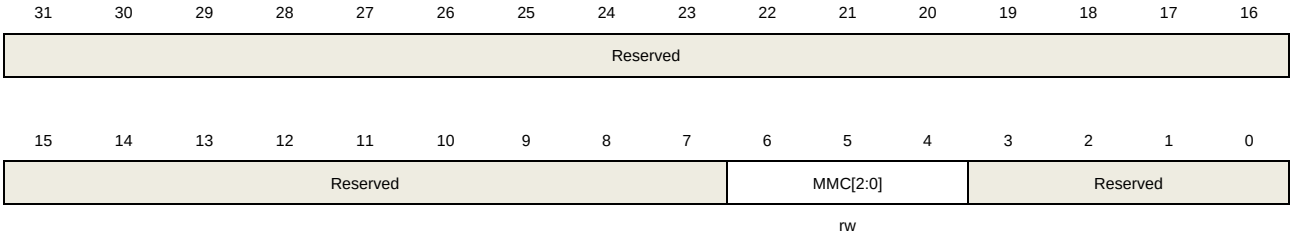
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock mode, pause mode or decoder mode. While in event mode, the hardware can set the CEN bit automatically.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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



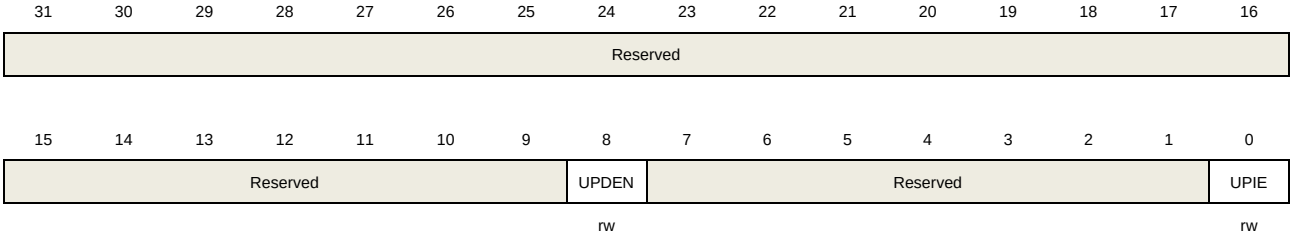
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset. 001: Enable. This mode is used to start several timers at the same time or control a slave timer to be enabled in a period. In this mode, the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 010: Update. In this mode, the master mode controller selects the update event as TRGO. 100~111: Reserved.
3:0	Reserved	Must be kept at reset value.

Interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

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



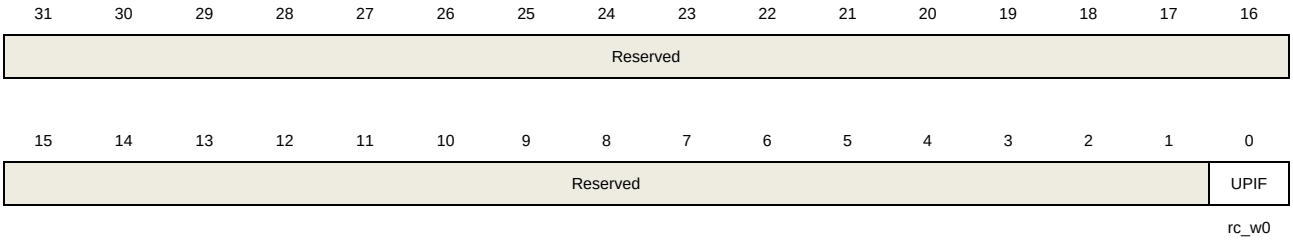
Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7:1	Reserved	Must be kept at reset value.
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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



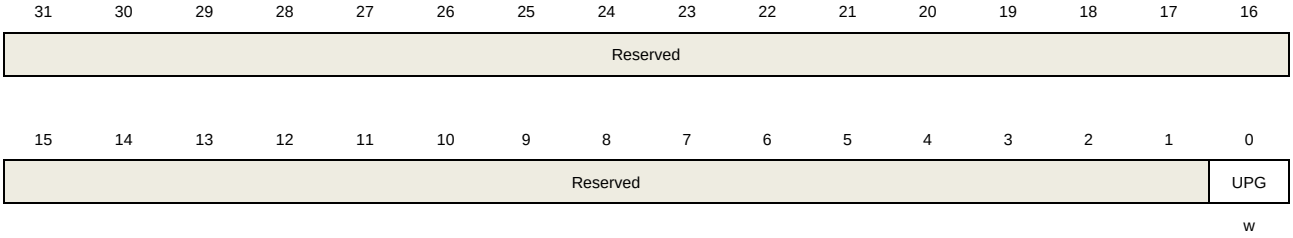
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

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



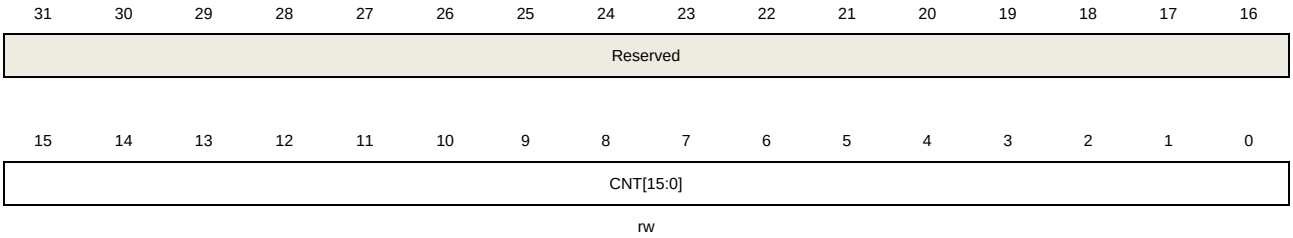
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPG	This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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



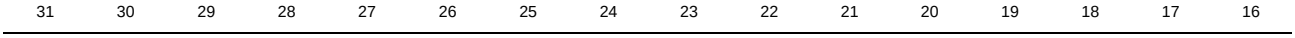
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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



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 PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-field will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 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
Reserved															

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

13. Real-time Clock(RTC)

13.1. Overview

The RTC is usually used as a clock-calendar. The RTC circuits are located in two power supply domains. The ones in the Backup Domain consist of a 32-bit up-counter, an alarm, a prescaler, a divider and the RTC clock configuration register. That means the RTC settings and time are kept when the device resets or wakes up from Standby mode. While the circuits in the VDD domain only include the APB interface and a control register. In the following sections, the details of the RTC function will be described.

13.2. Characteristics

- 32-bit programmable counter for counting elapsed time
  Programmable prescaler: Max division factor is up to 2^{20}
- Separate clock domains:
  - PCLK1 clock domain
  - RTC clock domain (this clock must be at least 4 times slower than the PCLK1 clock)
- RTC clock source:
  - HXTAL clock divided by 128
  - LXTAL oscillator clock
  - IRC40K oscillator clock
  - AHB clock divided by 10
- Maskable interrupt source:
  - Alarm interrupt
  - Second interrupt
  - Overflow interrupt

13.3. Function overview

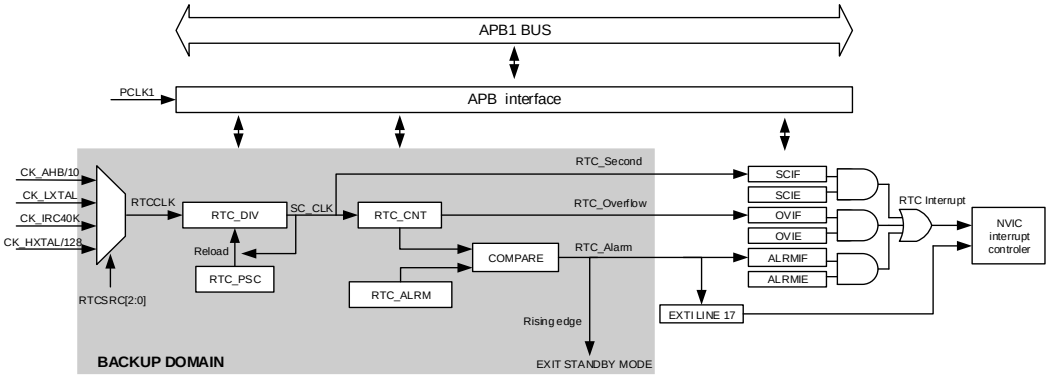
The RTC circuits consist of two major units: APB interface located in PCLK1 clock domain and RTC core located in RTC clock domain.

APB Interface is connected with the APB1 bus. It includes a set of registers, can be accessed by APB1 bus.

RTC core includes two major blocks. One is the RTC prescaler block, which generates the RTC time base clock SC_CLK. RTC prescaler block includes a 20-bit programmable divider (RTC prescaler) which can make SC_CLK is divided from RTC source clock. If second interrupt is enabled in the RTC_INTEN register, the RTC will generate an interrupt at every SC_CLK rising edge. Another block is a 32-bit programmable counter, which can be initialized

with the value of current system time. If alarm interrupt is enabled in the RTC_INTEN register, the RTC will generate an alarm interrupt when the system time equals to the alarm time (stored in the RTC_ALRMH/L register).

Figure 13-1. Block diagram of RTC



13.3.1. RTC reset

The APB interface and the RTC_INTEN register are reset by system reset. The RTC core (prescaler, divider, counter and alarm) is reset only by a backup domain reset.

Steps to enable access to the backup registers and the RTC after reset are as follows:

1. Set the PMUEN and BKPIEN bits in the RCU_APB1EN register to enable the power and backup interface clocks.

2. Enable access to the backup registers and RTC by setting the BKPWEN bit in the (PMU_CTL).

13.3.2. RTC reading

The APB interface and RTC core are located in two different power supply domains.

In the RTC core, only counter and divider registers are readable registers. And the values in the two registers and the RTC flags are internally updated at each rising edge of the RTC clock, which is resynchronized by the APB1 clock.

When the APB interface is immediately enabled from a disable state, the read operation is not recommended because the first internal update of the registers has not finished. That means, when a system reset, power reset, waking up from Standby mode or Deep-sleep mode occurs, the APB interface was in disabled state, but the RTC core has been kept running. In these cases, the correct read operation should first clear the RSYNF bit in the RTC_CTL register and wait for it to be set by hardware. While WFI and WFE have no effects on the RTC APB interface.

13.3.3. RTC configuration

The RTC_PSC, RTC_CNT and RTC_ALRM registers in the RTC core are writable. These registers' value can be set only when the peripheral enter configuration mode. And the CMF bit in the RTC_CTL register is used to indicate the configuration mode status. The write operation executes when the peripheral exit configuration mode, and it takes at least three RTCCLK cycles to complete. The value of the LWOFF bit in the RTC_CTL register sets to '1', if the write operation finished. The new write operation should wait for the previous one finished.

The configuration steps are as follows:

A) Wait until the value of LWOFF bit in the RTC_CTL register sets to '1';

B) Enter Configuration mode by setting the CMF bit in the RTC_CTL register;

C) Write to the RTC registers;

D) Exit Configuration mode by clearing the CMF bit in the RTC_CTL register;

E) Wait until the value of LWOFF bit in the RTC_CTL register sets to '1'.

13.3.4. RTC flag assertion

Before the update of the RTC Counter, the RTC second interrupt flag (SCIF) is asserted on the last RTCCLK cycle.

Before the counter equal to the RTC Alarm value which stored in the Alarm register increases by one, the RTC Alarm interrupt flag (ALRMIF) is asserted on the last RTCCLK cycle.

Before the counter equals to 0x0, the RTC Overflow interrupt flag (OVIF) is asserted on the last RTCCLK cycle.

The RTC Alarm write operation and Second interrupt flag must be synchronized by using either of the following sequences:

- Use the RTC alarm interrupt and update the RTC Alarm and/or RTC Counter registers inside the RTC interrupt routine;
- Update the RTC Alarm and/or the RTC Counter registers after the SCIF bit to be set in the RTC Control register.

Figure 13-2. RTC second and alarm waveform example (RTC_PSC = 3, RTC_ALRM = 2)

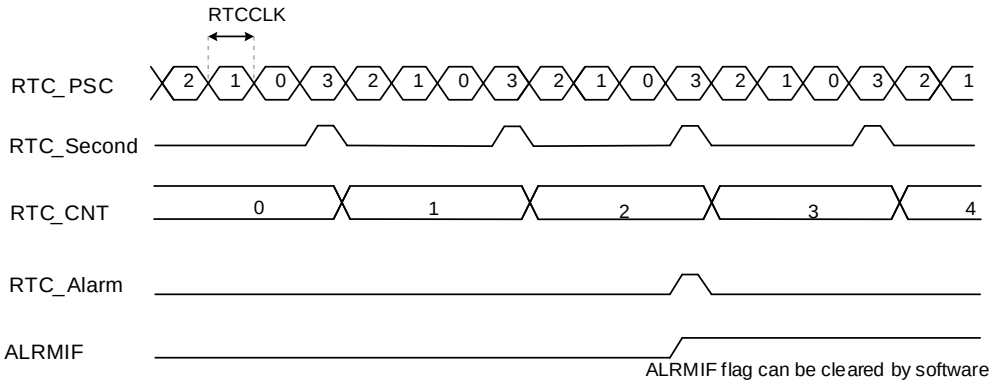
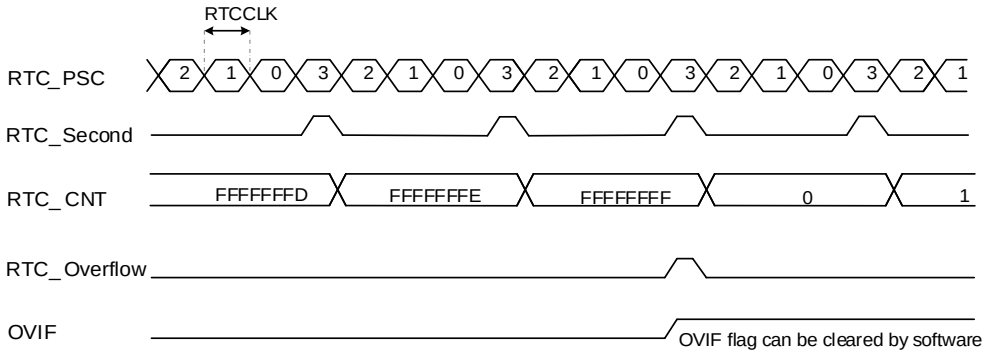


Figure 13-3. RTC second and overflow waveform example (RTC_PSC= 3)



13.4. RTC Register

RTC base address: 0x4000 2800

13.4.1. RTC interrupt enable register(RTC_INTEN)

Address offset : 0x00

Reset value: 0x0000 0000

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

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
													OVIE	ALRMIE	SCIE
													rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	OVIE	Overflow interrupt enable 0: Disable overflow interrupt 1: Enable overflow interrupt
1	ALRMIE	Alarm interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
0	SCIE	Second interrupt enable 0: Disable second interrupt. 1: Enable second interrupt

13.4.2. RTC control register(RTC_CTL)

Address offset: 0x04

Reset value: 0x0000 0020

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

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										LWOFF	CMF	RSYNF	OVIF	ALRMIF	SCIF
										r	rw	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions

31:6	Reserved	Must be kept at reset value.
5	LWOFF	Last write operation finished flag 0: Last write operation on RTC registers did not finished. 1: Last write operation on RTC registers finished.
4	CMF	Configuration mode flag 0: Exit configuration mode. 1: Enter configuration mode.
3	RSYNF	Registers synchronized flag 0: Registers not yet synchronized with the APB1 clock. 1: Registers synchronized with the APB1 clock.
2	OVIF	Overflow interrupt flag 0: Overflow event not detected 1: Overflow event detected. An interrupt will occur if the OVIE bit is set in RTC_INTEN.
1	ALRMIF	Alarm interrupt flag 0: Alarm event not detected 1: Alarm event detected. An interrupt named RTC global interrupt will occur if the ALRMIE bit is set in RTC_INTEN. And another interrupt named the RTC Alarm interrupt will occur if the EXTI 17 is enabled in interrupt mode.
0	SCIF	Second interrupt flag 0: Second event not detected. 1: Second event detected. An interrupt will occur if the SCIE bit is set in RTC_INTEN. Set by hardware when the divider reloads the value in RTC_PSCH/L, thus incrementing the RTC counter.

13.4.3. RTC prescaler high register (RTC_PSCH)

Address offset: 0x08

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PSC[19:16]			
												w			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.

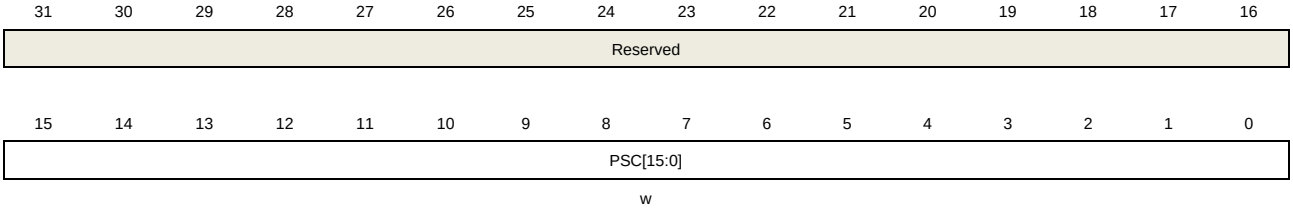
3:0	PSC[19:16]	RTC prescaler value high

13.4.4. RTC prescaler low register(RTC_PSCL)

Address offset: 0x0C

Reset value: 0x0000 8000

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



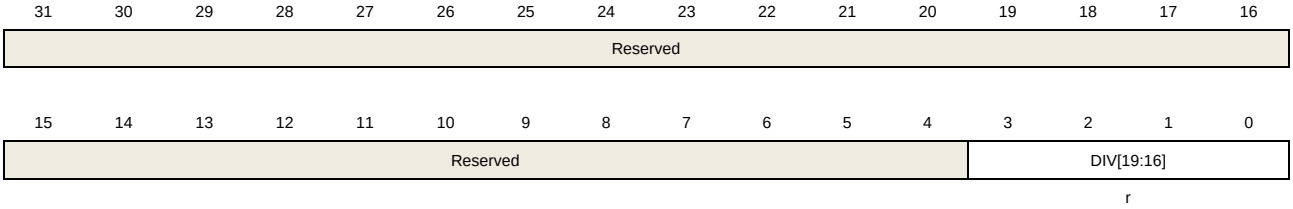
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	RTC prescaler value low The frequency of SC_CLK is the RTCCLK frequency divided by (PSC[19:0]+1).

13.4.5. RTC divider high register (RTC_DIVH)

Address offset: 0x10

Reset value: 0x0000 0000

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



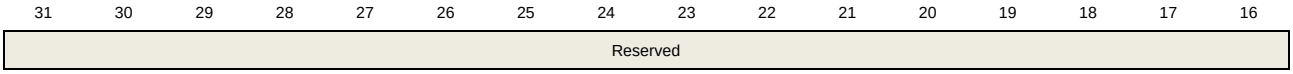
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	DIV[19:16]	RTC divider value high

13.4.6. RTC divider low register (RTC_DIVL)

Address offset: 0x14

Reset value: 0x0000 8000

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



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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DIV[15:0]	RTC divider value low The RTC divider register is reloaded by hardware when the RTC prescaler or RTC counter register updated.

13.4.7. RTC counter high register(RTC_CNTH)

Address offset: 0x18

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[31:16]	RTC counter value high

13.4.8. RTC counter low register (RTC_CNTL)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	RTC counter value low

13.4.9. RTC alarm high register(RTC_ALRMH)

Address offset: 0x20

Reset value: 0x0000 FFFF

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

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
ALRM[31:16]															
w															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	ALRM[31:16]	RTC alarm value high

13.4.10. RTC alarm low register (RTC_ALRML)

Address offset: 0x24

Reset value: 0x0000 FFFF

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

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
ALRM[15:0]															
w															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	ALRM[15:0]	RTC alarm value low

14. Watchdog timer (WDGT)

The watchdog timer (WDGT) is a hardware timing circuitry that can be used to detect system failures due to software malfunctions. There are two watchdog timer peripherals in the chip: free watchdog timer (FWDGT) and window watchdog timer (WWDGT). They offer a combination of a high safety level, flexibility of use and timing accuracy. Both watchdog timers are offered to resolve malfunctions of software.

The watchdog timer will generate a reset when the internal counter reaches a given value. The watchdog timer counter can be stopped while the processor is in the debug mode.

14.1. Free watchdog timer (FWDGT)

14.1.1. 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 timer can be enabled to prevent it from changing the configuration unexpectedly.

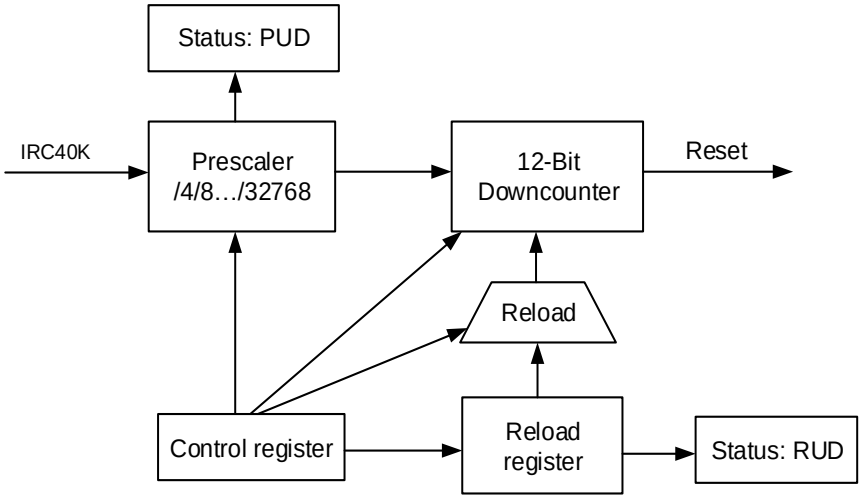
14.1.2. Characteristics

- 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.

14.1.3. Function overview

The free watchdog timer consists of an 8-stage prescaler and a 12-bit down-counter.*Figure 14-1. Free watchdog timer block diagram* shows the functional block of the free watchdog timer module.

Figure 14-1. Free watchdog timer block diagram



The free watchdog timer is enabled by writing the value 0xCCCC to the control register (FWDGT_CTL), then the counter starts counting down. When the counter reaches the value 0x000, there will be a reset.

The counter can be reloaded by writing the value (0xAAAA) to the FWDGT_CTL register at anytime. 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.

The free watchdog timer can automatically start when power on if 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 and the FWDGT_RLD register are write protected. Before writing these registers, the software should write the value (0x5555) to the FWDGT_CTL register. These registers will be protected again by writing any other value to the FWDGT_CTL register. When an update operation of the prescaler register (FWDGT_PSC) or the reload value register (FWDGT_RLD) is ongoing, the status bits in the FWDGT_STAT register are set.

If the FWDGT_HOLD bit in DBG module is cleared, the FWDGT continues to work even the Cortex®-M33 core halted (Debug mode). The FWDGT stops in Debug mode if the FWDGT_HOLD bit is set.

Table 14-1. Min/max FWDGT timeout period at 40 kHz (IRC40K)

Prescaler divider	PSC[3:0] bits	Min timeout (ms) RLD[11:0]=0x000	Max timeout (ms) RLD[11:0]=0xFFF
1 / 4	0000	0.1	409.6
1 / 8	0001	0.2	819.2
1 / 16	0010	0.4	1638.4
1 / 32	0011	0.8	3276.8
1 / 64	0100	1.6	6553.6
1 / 128	0101	3.2	13107.2

Prescaler divider	PSC[3:0] bits	Min timeout (ms) RLD[11:0]=0x000	Max timeout (ms) RLD[11:0]=0xFFF
1 / 256	0110	6.4	26214.4
1 / 512	0111	12.8	52,428.8
1 / 1024	1000	25.6	104,857.6
1 / 2048	1001	51.2	209,715.2
1 / 4096	1010	102.4	419,430.4
1 / 8192	1011	204.8	838,860.8
1 / 16384	1100	409.6	1,677,721.6
1 / 32768	1101~1111	819.2	3,355,443.2

The FWDGT timeout can be more accurate 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 7or more IRC32K clock cycles between two sonsecutive watchdog reload operations.

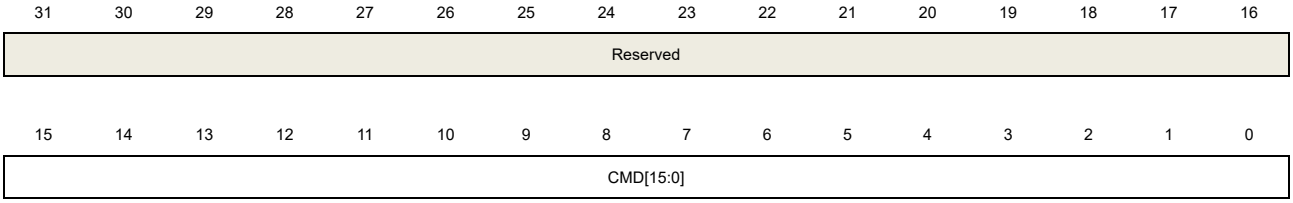
14.1.4. Register definition

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).

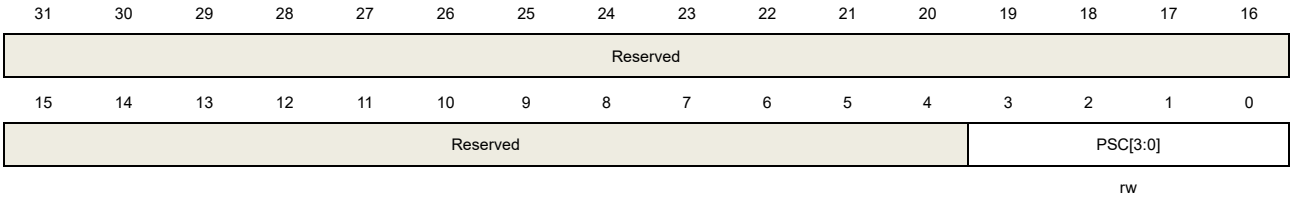


Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different fuctions are realized by writing these bits with different values: 0x5555: Disable the FWDGT_PSC and FWDGT_RLD write protection 0xCCCC: Start the free watchdog timer counter. When the counter 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
31:4	Reserved	Must be kept at reset value.
3:0	PSC[3:0]	Free watchdog timer prescaler selection. Write 0x5555 in the FWDGT_CTL register before writing these bits. During a write operation to this register, the PUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. 0000: 1 / 4 0001: 1 / 8

0010: 1 / 16
0011: 1 / 32
0100: 1 / 64
……
1100: 1 / 16384
1101~1111: 1 / 32768
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 except in case of low-power mode entry.

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).

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				RLD [11:0]											
				rw											

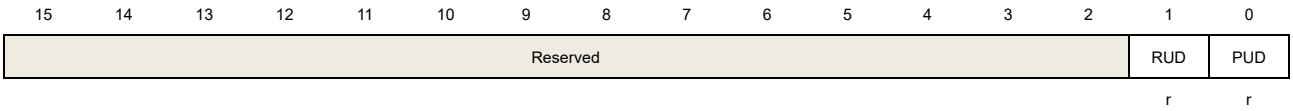
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT counter with the RLD value. These bits are write-protected. Write 0x5555 in the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit 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 except in case of low-power mode entry.

Status register (FWDGT_STAT)

Address offset: 0x0C
Reset value: 0x0000 0000

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

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



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

14.2. Window watchdog timer (WWDGT)

14.2.1. Overview

The window watchdog timer (WWDGT) is used to detect system failures due to software malfunctions. After the window watchdog timer starts, the value of down counter reduces progressively. The watchdog timer causes a reset when the counter reached 0x3F (the CNT[6] bit has been cleared). The watchdog timer also causes a reset when the counter is refreshed before the counter reached the window register value. So the software should refresh the counter in a limited window. The window watchdog timer generates an early wakeup status flag when the counter reaches 0x40. Interrupt occurs if it is enabled.

The window watchdog timer clock is prescaled from the APB1 clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

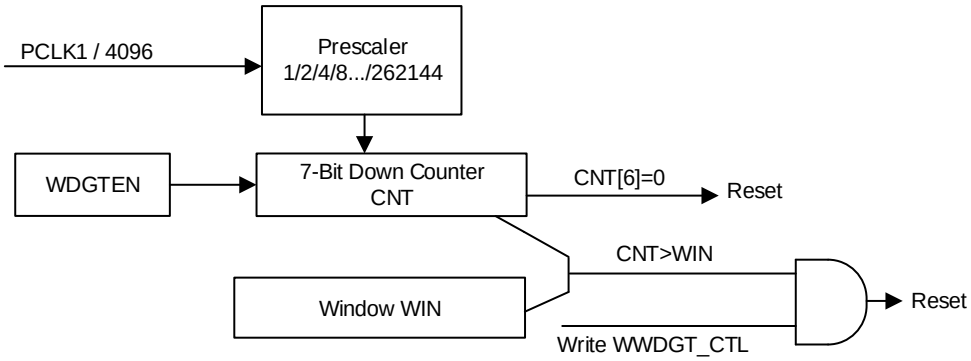
14.2.2. Characteristics

- Programmable free-running 7-bit downcounter.
- Generate a 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.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

14.2.3. Function overview

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

Figure 14-2. Window watchdog timer block diagram



The window watchdog timer is always disabled after power on reset. The software starts the

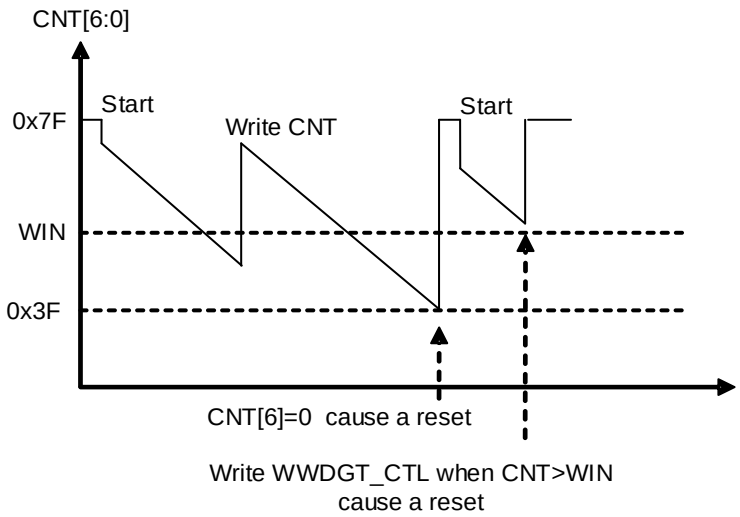
watchdog by setting the WDGTEN bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F, (it implies that the CNT[6] bit should be set). The CNT[5:0] determine the maximum time interval between two reloading. The count down speed depends on the APB1 clock and the prescaler (PSC[1:0] bits in the WWDGT_CFG register).

The WIN[6:0] bits in the configuration register (WWDGT_CFG) specifies the window value. The software can prevent the reset event by reloading the down counter. The counter value is less than the window value and greater than 0x3F, otherwise the watchdog causes a reset.

The early wakeup interrupt (EWI) is enabled by setting the EWIE bit in the WWDGT_CFG register, and the interrupt will be generated when the counter reaches 0x40. The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyse the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things.

The EWI interrupt is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 14-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (14\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB1 clock period measured in ms

The ***Table 14-2. Min / max timeout value at 110 MHz (fPCLK1)*** shows the minimum and maximum values of the t_{WWDGT}.

Table 14-2. Min / max timeout value at 110 MHz (f_{PCLK1})

Prescaler divider	PSC[4:0]	Min timeout value CNT[6:0] =0x40	Max timeout value CNT[6:0]=0x7F
1 / 1	00000	0.02926ms	1.872ms
1 / 2	00001	0.05851ms	3.745ms
1 / 4	00010	0.11703ms	7.49ms
1 / 8	00011	0.23406ms	14.98ms
1 / 16	00100	0.46811ms	29.96ms
1 / 32	00101	0.93623ms	59.92ms
1 / 64	00110	1.87ms	119.84ms
1 / 128	00111	3.75ms	239.68ms
1 / 256	01000	7.49ms	479.36ms
1 / 512	01001	14.98ms	958.72ms
1 / 1024	01010	29.96ms	1917.44ms
1 / 2048	01011	59.92ms	3834.88ms
1 / 4096	01100	119.84ms	7669.76ms
1 / 8192	01101	239.67ms	15339.52ms
1 / 16384	01110	479.35ms	30679.04ms
1 / 32768	01111	958.70ms	61358.08ms
1 / 65536	10000	1917.40ms	122716.16ms
1 / 131072	10001	3834.79ms	245432.32ms
1 / 262144	10010~11111	7669.58ms	490864.64ms

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M33 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

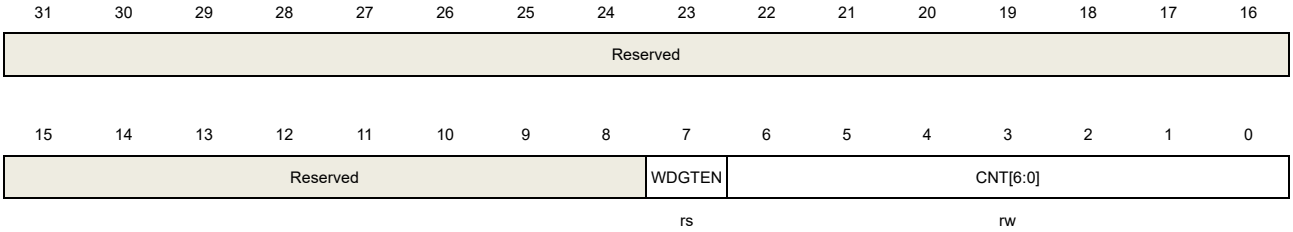
14.2.4. Register definition

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).

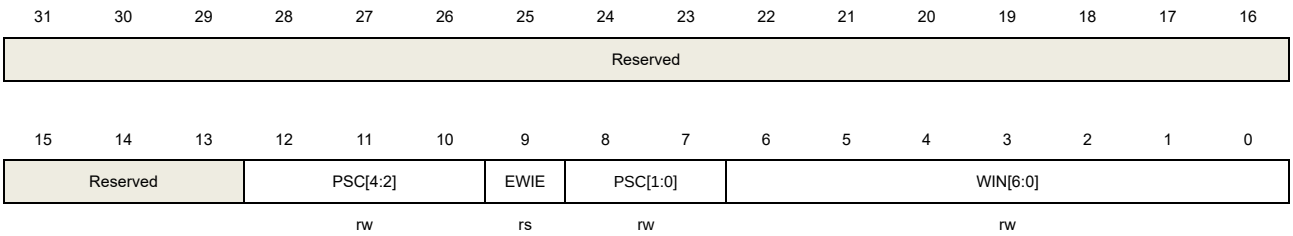


Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	WDGTEN	Start the window watchdog timer. It can be cleared by a hardware reset or software clock reset (refer to APB1 reset register (RCU_APB1RST)). Writing 0 has no effect. 0: Window watchdog timer disabled 1: Window watchdog timer enabled
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset occurs when the value of this counter decreases from 0x40 to 0x3F. When the value of this counter is greater than the window value, writing this counter also causes a reset.

Configuration register (WWDGT_CFG)

Address offset: 0x04
Reset value: 0x0000 007F

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



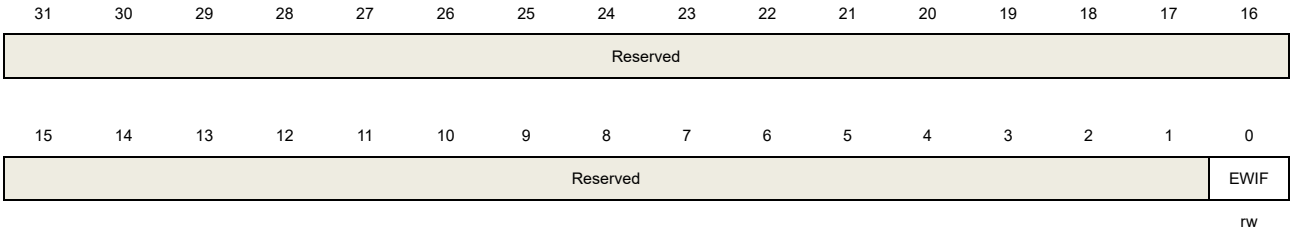
Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:10	PSC[4:2]	High bit value of prescaler. PSC[4:0] consists of high bit value PSC[4:2] and low bit value PSC[1:0]. PSC[4:0] is the time base of the watchdog counter:

		00000: (PCLK1 / 4096) / 1 00001: (PCLK1 / 4096) / 2 00010: (PCLK1 / 4096) / 4 00011: (PCLK1 / 4096) / 8 …… 10001: (PCLK1 / 4096) / 131072 10010~11111: (PCLK1 / 4096) / 262144
9	EWIE	Early wakeup interrupt enable. An interrupt occurs when the counter reaches 0x40 if the bit is set. It can be cleared by a hardware reset or by a RCU WWDGT software reset. A write operation of '0' has no effect.
8:7	PSC[1:0]	Low bit value of prescaler. PSC[4:0] consists of high bit value PSC[4:2] and low bit value PSC[1:0]. PSC[4:0] is the time base of the watchdog counter.
6:0	WIN[6:0]	The Window value. A reset occurs 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 reaches 0x40, this bit is set by hardware even the interrupt is not enabled (EWIE in WWDGT_CFG is cleared). This bit is cleared by writing 0 to it. There is no effect when writing 1 to it.

15. Debug (DBG)

15.1. Introduction

The GD32F50x series provide a large variety of debug, trace and test features. They are implemented with a standard configuration of the Arm® CoreSightTM module together with a daisy chained standard TAP controller. Debug and trace functions are integrated into the Arm® Cortex®-M33. The debug system supports serial wire debug (SWD) and trace functions in addition to standard JTAG debug. The debug and trace functions refer to the following documents:

- Cortex®-M33 Technical Reference Manual
- Arm® Debug Interface v5 Architecture Specification

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

15.2. JTAG/SW function description

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

15.2.1. Switch JTAG or SW interface

By default, the JTAG interface is active. The sequence for switching from JTAG to SWD is:

- Send 50 or more TCK cycles with TMS = 1.
- Send the 16-bit sequence on TMS = 1110011110011110 (0xE79E LSB first).
- Send 50 or more TCK cycles with TMS = 1.

The sequence for switching from SWD to JTAG is:

- Send 50 or more TCK cycles with TMS = 1.
- Send the 16-bit sequence on TMS = 1110011100111100 (0xE73C LSB first).
- Send 50 or more TCK cycles with TMS = 1.

15.2.2. Pin assignment

The JTAG interface provides 5-pin standard JTAG, known as JTAG clock (JTCK), JTAG mode selection (JTMS), JTAG data input (JTDI), JTAG data output (JTDO) and JTAG reset (NJTRST, active low). The serial wire debug (SWD) provide 2-pin SW interface, known as SW data input/output (SWDIO) and SW clock (SWCLK). The two SW pin are multiplexed with two of five JTAG pin, which is SWDIO multiplexed with JTMS, SWCLK multiplexed with JTCK. The JTDO is also used as Trace async data output (TRACESWO) when async trace enabled.

Table 15-1. Pin assignment

Pin	Debug interface
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO
PB4	NJTRST
PB3	JTDO

By default, 5-pin standard JTAG debug mode is chosen after reset. Users can also use JTAG function without NJTRST pin, then the PB4 can be used to other GPIO functions. (NJTRST tied to 1 by hardware). If switch to SW debug mode, the PA15/PB4/PB3 are released to other GPIO functions. If JTAG and SW not used, all 5-pin can be released to other GPIO functions. Please refer to ***General-purpose and alternate-function I/Os (GPIO and AFIO).***

15.2.3. JTAG daisy chained structure

The Cortex®-M33 JTAG TAP is connected to a MCU JTAG TAP. The MCU JTAG IR is 5-bit width, while the Cortex®-M33 JTAG IR is 4-bit width. So when JTAG in IR shift step, it first shift 5-bit BYPASS instruction for MCU JTAG, and then shift normal 4-bit instruction for Cortex®-M33 JTAG. Because of the data shift under MCU JTAG BYPASS mode, adding 1 extra bit to the data chain is needed.

The MCU JTAG IDCODE is 0x790007A3.

15.2.4. Debug reset

The JTAG-DP and SW-DP register are in the power on reset domain. The System reset initializes the majority of the Cortex®-M33, excluding NVIC and debug logic, (FPB, DWT, and ITM). The NJTRST reset can reset JTAG TAP controller only. So, it can perform debug feature under system reset. Such as, halt-after-reset, which is the debugger sets halt under system reset, and the core halts immediately after the system reset is released.

15.2.5. JEDEC-106 ID code

The Cortex®-M33 integrates JEDEC-106 ID code, which is located in ROM table and mapped on the address of 0xE00FF000_0xE00FFFFF.

15.3. Debug hold function description

15.3.1. Debug support for power saving mode

When STB_HOLD bit in DBG control register (DBG_CTL) is set and entering the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, and the debugger can debug in standby mode. When exit the standby mode, a system reset generated.

When DSLP_HOLD bit in DBG control register (DBG_CTL) is set and entering the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M, and the debugger can debug in Deep-sleep mode.

When SLP_HOLD bit in DBG control register (DBG_CTL) is set and entering the sleep mode, the clock of AHB bus for CPU is not closed, and the debugger can debug in sleep mode.

15.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT and CAN

When the core halted and the corresponding bit in DBG control register (DBG_CTL) is set, the following behaved.

For TIMER, the timer counters stopped and hold for debug.

For I2C, SMBUS timeout hold for debug.

For WWDGT or FWDGT, the counter clock stopped for debug.

For CAN, the receive register stopped counting for debug.

15.4. DBG registers

DBG base address: 0xE004 5000

15.4.1. ID code register (DBG_ID)

Address offset: 0x00

Read only

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

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

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

Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register These bits read by software, These bits are unchanged constant

15.4.2. Control register (DBG_CTL)

Address offset: 0x04

Reset value: 0x0000 0000; power reset only

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved.							TIMER16_HOLD	TIMER15_HOLD	Reserved	CAN1_HOLD	TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER7_HOLD	I2C1_HOLD
							rw	rw		rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	CAN0_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	Reserved		TRACE_IOEN	Reserved		STB_HOLD	DSLP_HOLD	SLP_HOLD
rw	rw	rw	rw	rw	rw	rw	rw			rw			rw	rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	TIMER16_HOLD	TIMER 16 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER 16 counter for debug when core halted
23	TIMER15_HOLD	TIMER 15 hold bit

		This bit is set and reset by software 0: no effect 1: hold the TIMER 15 counter for debug when core halted
22	Reserved	Must be kept at reset value
21	CAN1_HOLD	CAN1 hold bit This bit is set and reset by software 0: no effect 1: the receive register of CAN1 stops receiving data when core halted
20	TIMER6_HOLD	TIMER6 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER6 counter for debug when core halted
19	TIMER5_HOLD	TIMER5 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER5 counter for debug when core halted
18	TIMER4_HOLD	TIMER4 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER4 counter for debug when core halted
17	TIMER7_HOLD	TIMER7 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER7 counter for debug when core halted
16	I2C1_HOLD	I2C1 hold bit This bit is set and reset by software 0: no effect 1: hold the I2C1 SMBUS timeout for debug when core halted
15	I2C0_HOLD	I2C0 hold bit This bit is set and reset by software 0: no effect 1: hold the I2C0 SMBUS timeout for debug when core halted
14	CAN0_HOLD	CAN0 hold bit This bit is set and reset by software 0: no effect 1: the receive register of CAN0 stops receiving data when core halted
13	TIMER3_HOLD	TIMER 3 hold bit This bit is set and reset by software 0: no effect

		1: hold the TIMER 3 counter for debug when core halted
12	TIMER2_HOLD	TIMER 2 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER 2 counter for debug when core halted
11	TIMER1_HOLD	TIMER 1 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER 1 counter for debug when core halted
10	TIMER0_HOLD	TIMER 0 hold bit This bit is set and reset by software 0: no effect 1: hold the TIMER 0 counter for debug when core halted
9	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software 0: no effect 1: hold the WWDGT counter clock for debug when core halted
8	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software 0: no effect 1: hold the FWDGT counter clock for debug when core halted
7:6	Reserved	Must be kept at reset value
5	TRACE_IOEN	Trace pin allocation enable This bit is set and reset by software 0: Trace pin allocation disable 1: Trace pin allocation enable
4:3	Reserved	Must be kept at reset value
2	STB_HOLD	Standby mode hold register This bit is set and reset by software 0: no effect 1: At the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, a system reset generated when exit standby mode
1	DSLP_HOLD	Deep-sleep mode hold register This bit is set and reset by software 0: no effect 1: At the Deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M
0	SLP_HOLD	Sleep mode hold register This bit is set and reset by software

0: no effect

1: At the sleep mode, the clock of AHB is on.

16. Cyclic redundancy checks management unit (CRC)

16.1. Overview

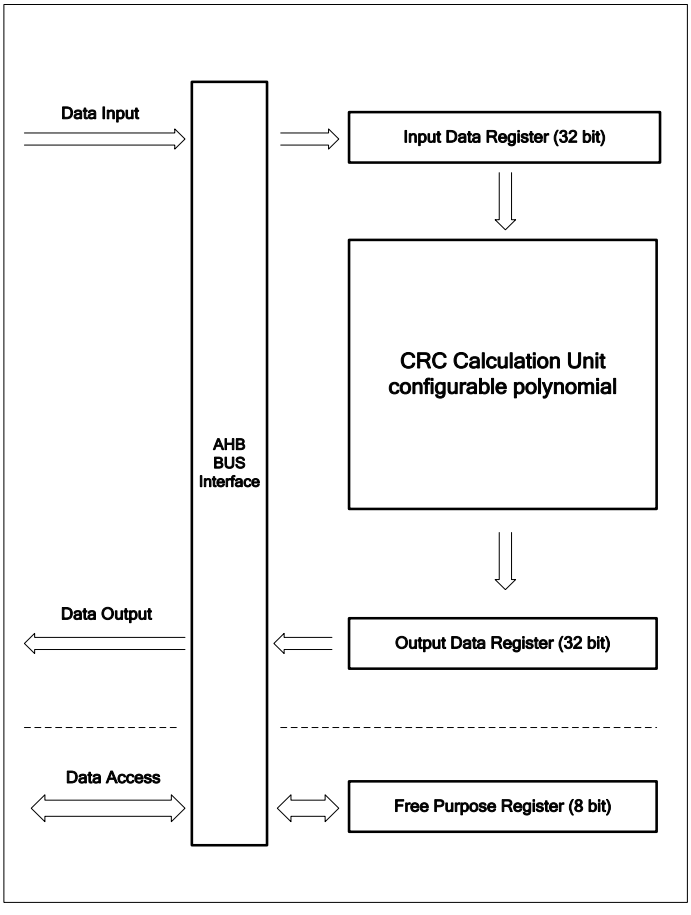
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

This CRC management unit can be used to calculate 7 / 8 / 16 / 32 bits CRC code within user configurable polynomial.

16.2. Characteristics

- Supports 8 / 16 / 32 bits data input.
- For 8 / 16 / 32 bits input data length, the calculation cycles are 1 / 2 / 4 AHB clock cycles.
- User configurable polynomial value and size (7 / 8 / 16 / 32 bits).
- User can configure CRC initial value.
- Free 8-bit register is unrelated to calculation and can be used for any other goals by any other peripheral devices.

Figure 16-1. Block diagram of CRC calculation unit



16.3. Function overview

- CRC calculation unit is used to calculate the 32-bit raw data, and CRC_DATA register will receive the raw data and store the calculation result.

  If the CRC_DATA register has not been cleared by setting the CRC_CTL register, the new input raw data will be calculated based on the result of previous value of CRC_DATA.

  CRC calculation will spend 4 / 2 / 1 AHB clock cycles for 32 / 16 / 8 bits data size. During this period, AHB will not be hanged because of the existence of the 32-bit input buffer.

- This module supplies an 8-bit free register CRC_FDATA.

  CRC_FDATA is unrelated to the CRC calculation. Independent read and write operations can be performed at any time.

- Reversible function can reverse the input data and output data.

  For input data, 3 reverse types can be selected.

  Original data is 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.

- User configurable initial calculation data is available.

  When RST bit is set or write operation to CRC_IDATA register, the CRC_DATA register will be automatically initialized to the value in CRC_IDATA.

- User configurable polynomial.

  Depends on PS[1:0] bits, the valid polynomial and output bit width can be selected by user.It is strongly recommend to reset the CRC calculation unit after changing the PS[1:0] bits or polynomial.

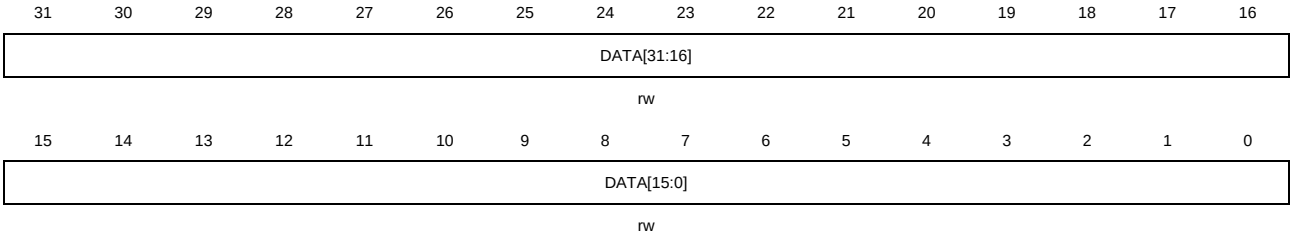
16.4. Register definition

CRC base address: 0x4002 3000

16.4.1. Data register (CRC_DATA)

Address offset: 0x00
Reset value: 0xFFFF FFFF

This register is read accessed by word (32-bit), and written accessed in byte (8-bit), half-word (16-bit), or word (32-bit).

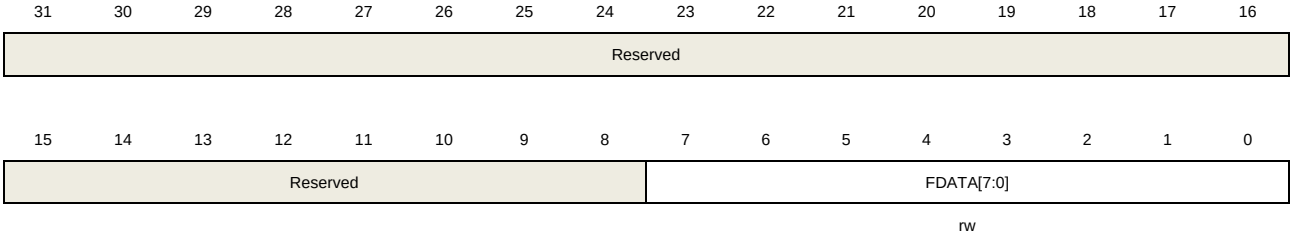


Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software writes and reads. This register is used to calculate new data, and the register can be written the new data directly. Write value cannot be read because the read value is the previous CRC calculation result.

16.4.2. Free data register (CRC_FDATA)

Address offset: 0x04
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free data register bits Software writes and reads. These bits are unrelated with CRC calculation. This byte can be used for any goal

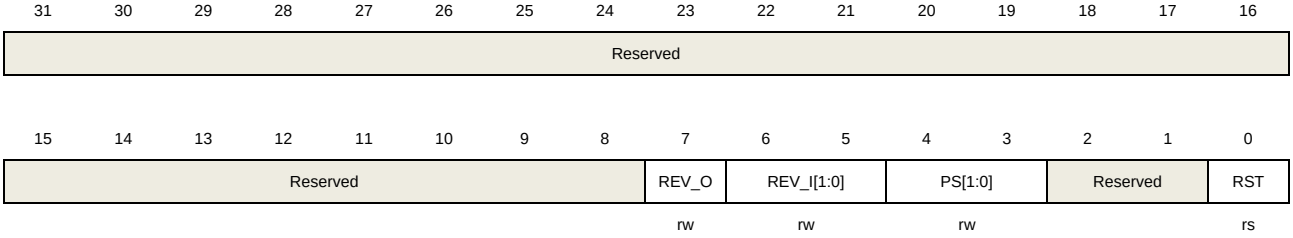
by any other peripheral. The CRC_CTL register will generate no effect to the byte.

16.4.3. Control register (CRC_CTL)

Address offset: 0x08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	REV_O	Reverse output data value in bit order 0:Not bit reversed for output data 1:Bit reversed for output data
6:5	REV_I[1:0]	Reverse type for input data 0: Dot not use reverse for input data 1: Reverse input data with every 8-bits length 2: Reverse input data with every 16-bits length 3: Reverse input data with whole 32-bits length
4:3	PS[1:0]	Size of polynomial 0: 32 bits 1: 16 bits (POLY [15:0] is used for calculation.) 2: 8 bits (POLY [7:0] is used for calculation.) 3: 7 bits (POLY [6:0] is used for calculation.)
2:1	Reserved	Must be kept at reset value.
0	RST	Software writes and reads. Set this bit can reset the CRC_DATA register. When set, the value of the CRC_DATA register is automatically initialized to the value in the CRC_IDATA register and then automatically cleared by hardware. This bit will take no effect to CRC_FDATA.

16.4.4. 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.

16.4.5. Polynomial register (CRC_POLY)

Address offset: 0x14

Reset value: 0x04C1 1DB7

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

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

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

Bits	Fields	Description
31:0	POLY[31:0]	User configurable polynomial value This value is used together with PS[1:0] bits.

17. True random number generator (TRNG)

17.1. Overview

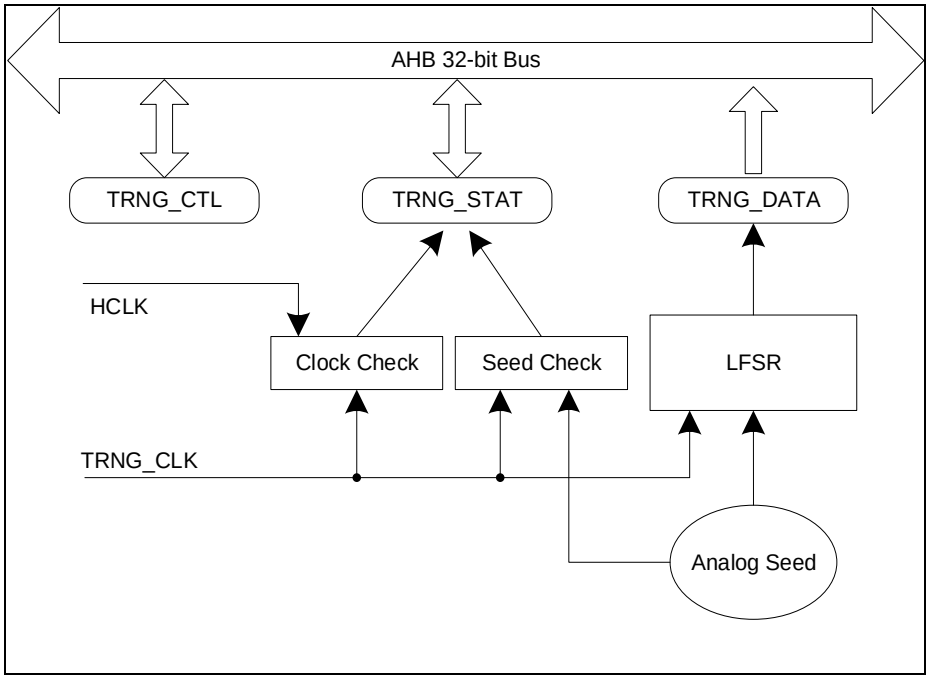
The true random number generator (TRNG) module can generate a 32-bit random value by using continuous analog noise.

17.2. Characteristics

- About 40 periods of TRNG_CLK are needed between two consecutive random numbers.
- 32-bit random value seed is generated from analog noise, so the random number is a true random number.

17.3. Function overview

Figure 17-1. TRNG block diagram



The random number seed comes from analog circuit. This analog seed is then plugged into a linear feedback shift register (LFSR), where a 32-bit width random number is generated.

The analog seed is generated by several ring oscillators. The LFSR is driven by a configurable TRNG_CLK (refer to ***Reset and clock unit (RCU)*** chapter), so that the quality of the generated random number depends on TRNG_CLK exclusively, no matter what HCLK frequency was set or not.

The 32-bit value of LFSR will transfer into TRNG_DATA register after a sufficient number of

seeds have been sent to the LFSR.

At the same time, the analog seed and TRNG_CLK clock are monitored. When an analog seed error or a clock error occurs, the corresponding status bit in TRNG_STAT will be set and an interrupt will generate if the IE bit in TRNG_CTL is set.

17.3.1. Operation flow

The following steps are recommended for using TRNG block:

1). Enable the interrupt as necessary, so that when a random number or an error occurs, an interrupt will be generated.

2). Enable IRC48M clock or select CK_PLL0 for USBFSPSC[1:0] to enable CK_TRNG clock.

3). Enable the TRNGEN bit.

4). When an interrupt occurs, check the status register TRGN_STAT, if SEIF=0, CEIF=0 and DRDY=1, then the random value in the data register could be read.

As required by the FIPS PUB 140-2, the first random data in data register should be saved but not be used. Every subsequent new random data should be compared to the previously random data. The data can only be used if it is not equal to the previously one.

17.3.2. Error flags

- Clock error

When the TRNG_CLK frequency is lower than the 1/16 of HCLK, the CECS and CEIF bit will be set. In this case, the application should check TRNG_CLK and HCLK frequency configurations and then clear CEIF bit. Clock error will not impact the previous random data.

- Seed error

When the analog seed is not changed or always changing during 64 TRNG_CLK periods, the SECS and SEIF bit will be set. In this case, the random data in data register should not be used. The application needs to clear the SEIF bit, then clear and set TRNGEN bit for restarting the TRNG.

17.4. Register definition

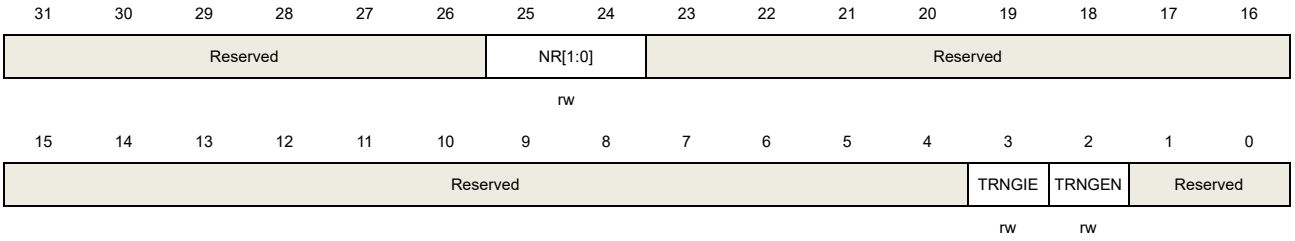
TRNG base address: 0x4002 3C00

17.4.1. Control register (TRNG_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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



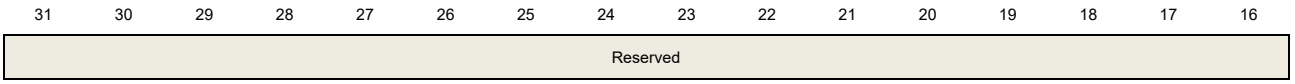
Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	NR[1:0]	analog rng power mode. 00: ultra low; may lead to weak randomness 01: low 10: medium 11: high
23:4	Reserved	Must be kept at reset value.
3	TRNGIE	Interrupt enabled bit. This bit controls the generation of an interrupt when DRDY, SEIF or CEIF was set. 0: Disable TRNG interrupt 1: Enable TRNG interrupt
2	TRNGEN	TRNG enabled bit. 0: Disable TRNG module (reduce power consuming) 1: Enable TRNG module
1:0	Reserved	Must be kept at reset value.

17.4.2. Status register (TRNG_STAT)

Address offset: 0x04

Reset value: 0x0000 0000

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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									SEIF	CEIF	Reserved		SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	SEIF	Seed error interrupt flag This bit will be set if more than 64 consecutive same bit or more than 32 consecutive 01 (or 10) changing are detected. 0: No fault detected 1: Seed error has been detected. The bit is cleared by writing 0.
5	CEIF	Clock error interrupt flag This bit will be set if TRNG_CLK frequency is lower than 1 / 16 HCLK frequency. 0: No fault detected 1: Clock error has been detected. The bit is cleared by writing 0.
4:3	Reserved	Must be kept at reset value.
2	SECS	Seed error current status 0: Seed error is not detected at current time. In case of SEIF = 1 and SECS = 0, it means seed error has been detected before but now is recovered. 1: Seed error is detected at current time if more than 64 consecutive same bits or more than 32 consecutive 01(or 10) changing are detected
1	CECS	Clock error current status 0: Clock error is not detected at current time. In case of CEIF = 1 and CECS = 0, it means clock error has been detected before but now is recovered. 1: Clock error is detected at current time. TRNG_CLK frequency is lower than 1 / 16 HCLK frequency.
0	DRDY	Random data ready status bit. This bit is cleared by reading the TRNG_DATA register and set when a new random number is generated. 0: The content of TRNG data register is not available. 1: The content of TRNG data register is available.

17.4.3. Data register (TRNG_DATA)

Address offset: 0x08
Reset value: 0x0000 0000

Application must make sure DRDY is set before reading 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
TRNGDATA[31:16]															

r

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

r

Bits	Fields	Descriptions
31:0	TRNGDATA[31:0]	32-bit random data

18. Cryptographic Acceleration Unit (CAU)

18.1. Overview

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with AES (128, 192, or 256) algorithm. This module follows the following standards:

- The Advanced Encryption Standard (AES) is announced by Federal Information Processing Standards Publication 197, November 26, 2001.

AES algorithms with different key sizes are supported to perform data encryption and decryption in the CAU in multiple modes.

The CAU is a 32-bit peripheral, DMA transfer is supported and data can be accessed in the input and output FIFO.

18.2. Characteristics

- AES encryption/decryption algorithm is supported.
- DMA transfer for incoming and outgoing data is supported.
- Supports the ECB algorithm.
- Supports 128-bit, 192-bit and 256-bit keys.
- 8*32-bit input and output FIFO.
- Multiple data types are supported, including No swapping, Half-word swapping Byte swapping and Bit swapping.
- Data can be transferred by DMA, CPU during interrupts, or without both of them.

18.3. CAU data type

The cryptographic acceleration unit receives data of 32 bits at a time, while they are processed in 128 bits for AES algorithms. For each data block, according to the data type, the data could be bit / byte / half-word / no swapped before they are transferred into the cryptographic acceleration processor. The same swapping operation should be also performed on the processor output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian.

Figure 18-1. DATAM No swapping and Half-word swapping and *Figure 18-2. DATAM Byte swapping and Bit swapping* illustrate the 128-bit AES block data swapping according to different data types.

Figure 18-1. DATAM No swapping and Half-word swapping

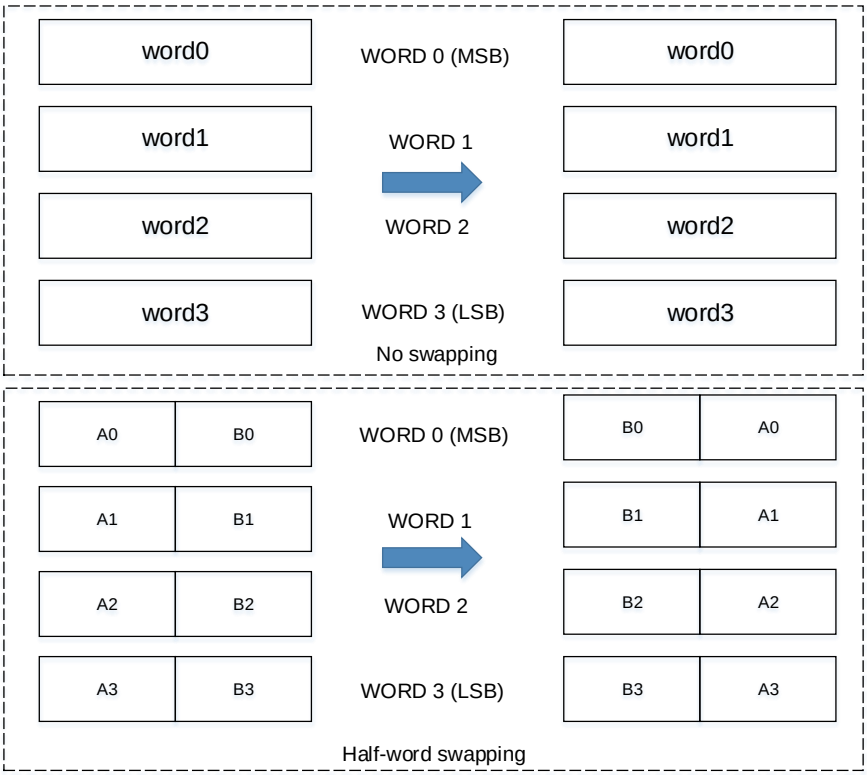
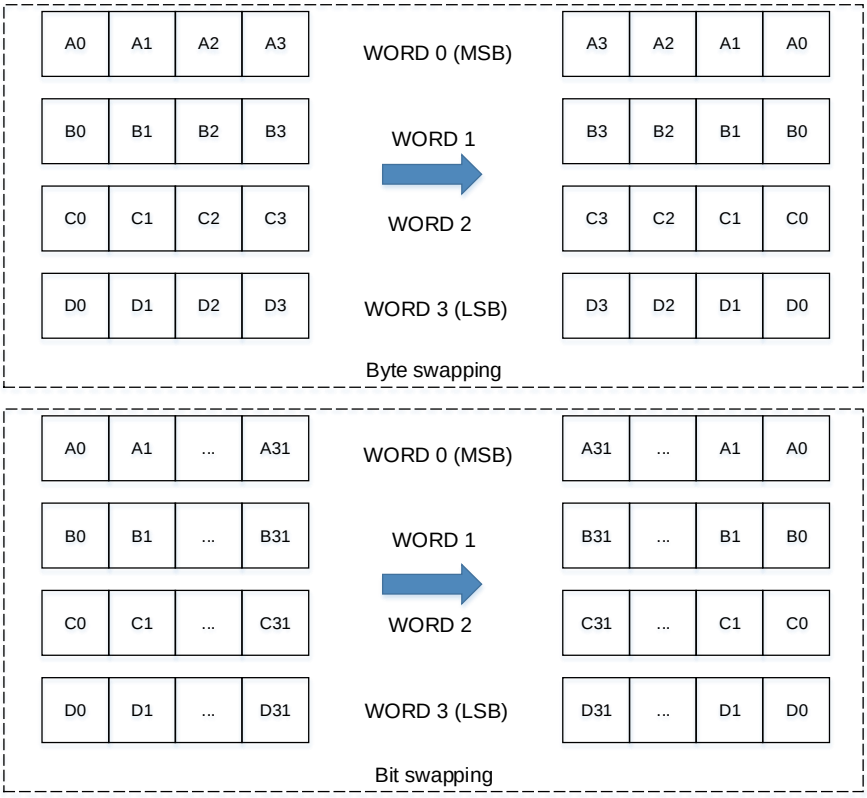


Figure 18-2. DATAM Byte swapping and Bit swapping

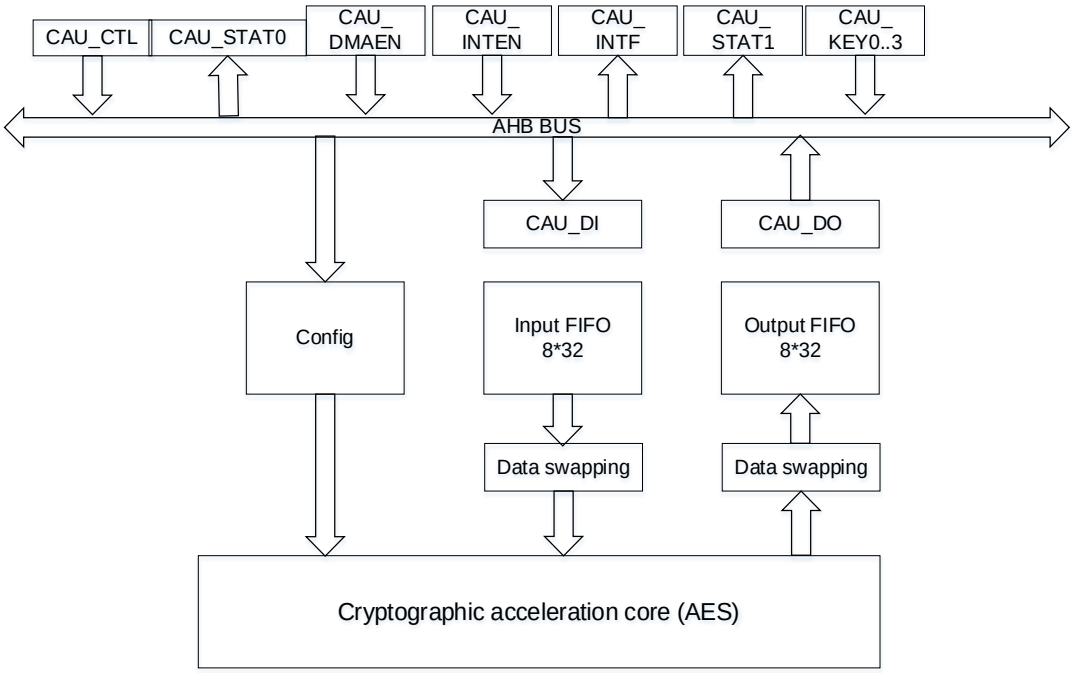


18.4. Cryptographic acceleration processor

The cryptographic acceleration unit implements AES acceleration processors, which are detailed described in section ***AES cryptographic acceleration processor***.

Figure 18-3. CAU diagram shows the block diagram of the cryptographic acceleration unit.

Figure 18-3. CAU diagram



18.4.1. AES cryptographic acceleration processor

The AES cryptographic acceleration processor consists of three components, including the AES algorithm (AEA), multiple keys and the initialization vectors or Nonce.

Three lengths of AES keys are supported: 128, 192 and 256 bits, and different initialization vectors or nonce are used depends on the operation mode.

The AES key is used as [KEY3 KEY2] when the key size is configured as 128, [KEY3 KEY2 KEY1] when the key size is configured as 192 and [KEY3 KEY2 KEY1 KEY0] when the key size is configured as 256.

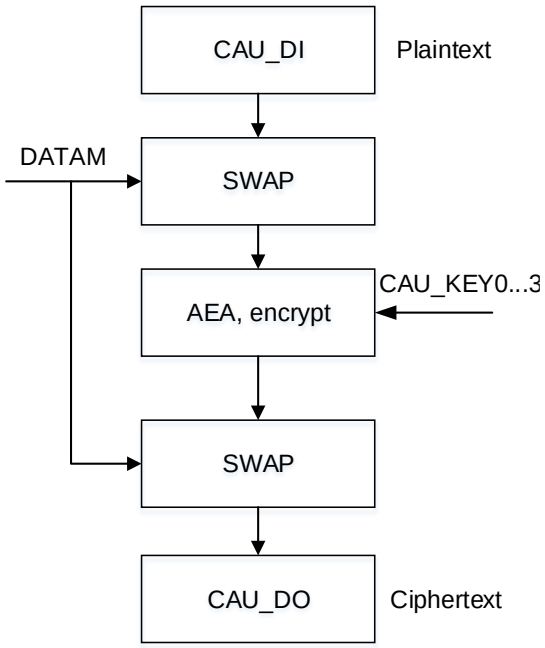
The thorough explanation of the key used in the AES is provided in FIPS PUB 197 (November 26, 2001), and the explanation process is omitted in this manual.

AES-ECB mode encryption

The 128-bit input plaintext is first obtained after data swapping according to the data type. The input data block is read in the AEA and encrypted using the 128, 192 or 256 -bit key. The output after above process is then swapped back according to the data type again, and a

128-bit ciphertext is produced and stored in the out FIFO. The procedure of AES ECB mode encryption is illustrated in ***Figure 18-4. AES ECB encryption***.

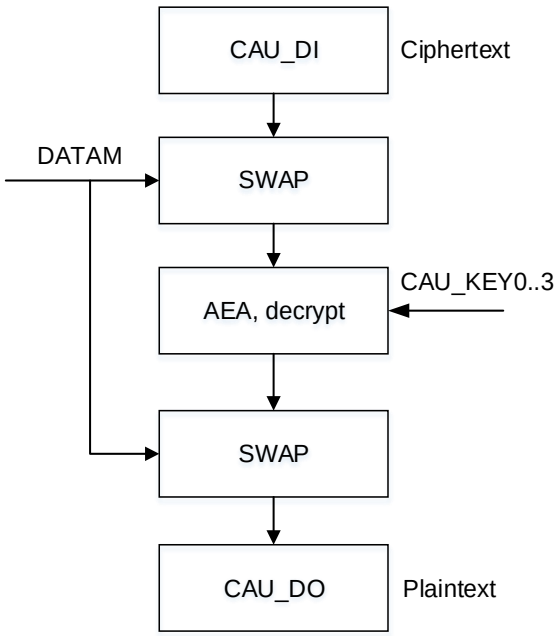
Figure 18-4. AES ECB encryption



AES-ECB mode decryption

First of all, the key derivation must be completed to prepare the decryption keys, the input key of the key schedule is the same to that used in encryption. The last round key obtained from the above operation is then used as the first round key in the decryption. After the key derivation, the 128-bit input ciphertext is first obtained after data swapping according to the data type. The input data block is read in the AEA and decrypted using keys prepared above. The output is then swapped back according to the data type again, and a 128-bit plaintext is produced. The procedure of AES ECB mode decryption is illustrated in ***Figure 18-5. AES ECB decryption***.

Figure 18-5. AES ECB decryption



18.5. Operating modes

Encryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Configure the CAU_KEY0..3(H / L) registers according to the algorithm.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the algorithm (AES) and the chaining mode (ECB) by writing the ALGM[2:0] bit in the CAU_CTL register.
6. Configure the encryption direction by writing 0 to the CAUDIR bit in the CAU_CTL register.
7. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
8. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
9. If the INF bit in the CAU_STAT0 register is 1, then write data blocks into the CAU_DI register. The data can be transferred by DMA / CPU during interrupts / no DMA or interrupts.
10. Wait for ONE bit in the CAU_STAT0 register is 1 then read the CAU_DO register. The output data can also be transferred by DMA / CPU during interrupts / no DMA or interrupts.
11. Repeat steps 9, 10 until all data blocks has been encrypted.

Decryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.

2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Configure the CAU_KEY0..3(H / L) registers according to the algorithm.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the ALGM[2:0] bits to “111” in the CAU_CTL register to complete the key derivation.
6. Enable the CAU by set the CAUEN bit as 1.
7. Wait until the BUSY and CAUEN bit return to 0 to make sure that the decryption keys are prepared.
8. Configure the algorithm (AES) and the chaining mode (ECB) by writing the ALGM[2:0] bit in the CAU_CTL register.
9. Configure the decryption direction by writing 1 to the CAUDIR bit in the CAU_CTL register.
10. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
11. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
12. If the INF bit in the CAU_STAT0 register is 1, then write data blocks into the CAU_DI register. The data can be transferred by DMA / CPU during interrupts/no DMA or interrupts.
13. Wait for ONE bit in the CAU_STAT0 register is 1, then read the CAU_DO register. The output data can also be transferred by DMA / CPU during interrupts / no DMA or interrupts.
14. Repeat steps 12, 13 until all data blocks has been decrypted.

18.6. CAU DMA interface

The DMA can be used to transfer data blocks with the interface of the cryptographic acceleration unit. The operations can be controlled by the CAU_DMAEN register. DMAIEN is used to enable the DMA request during the input phase, then a word is written into CAU_DI from DMA. DMAOEN is used to enable the DMA request during the output phase, then a word is read from the CAU.

Single and Burst transfers are both supported to ensure the data transfer if the number of words is not an integral multiple of burst size. Note the DMA controller should be configured to perform burst of 4 words or less to make sure no data will be lost. DMA channel for output data has a higher priority than that channel for input data so that the output FIFO can be empty earlier than that the input FIFO is full.

18.7. CAU interrupts

There are two types of interrupt registers in CAU, which are CAU_STAT1 and CAU_INTF. In CAU, the interrupt is used to indicate the situation of the input and output FIFO.

Any of input and output FIFO interrupt can be enabled or disabled by configuring the Interrupt

Enable register CAU_INTEN. Value 1 of the register enable the interrupts.

Input FIFO interrupt

The input FIFO interrupt is asserted when the number of words in the input FIFO is less than four words, then ISTA is asserted. And if the input FIFO interrupt is enabled by IINTEN with a 0 value, the IINTF is also asserted. Note if the CAUEN is low, then the ISTA and IINTF are also always low.

Output FIFO interrupt

The output FIFO interrupt is asserted when the number of words in the output FIFO is more than one words, then OSTA is asserted. And if the output FIFO interrupt is enabled by OINTEN with a 0 value, the OINTF is also asserted. Note Unlike that of Input FIFO interrupt, the value of CAUEN will never affect the situation of OSTA and OINTF.

18.8. Register definition

CAU base address: 0x4002 3400

18.8.1. Control register (CAU_CTL)

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
CAUEN	FFLUSH	Reserved				KEYM[1:0]		DATAM[1:0]		ALGM[2:0]			CAUDIR	Reserved	
rw	w					rw		rw		rw			rw		

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	CAUEN	CAU Enable 0: CAU is disabled 1: CAU is enabled **Note:** the CAUEN can be cleared automatically when the key derivation (ALGM=111b) is finished.
14	FFLUSH	Flush FIFO 0: No effect 1: When CAUEN=0, flush the input and output FIFO Reading this bit always returns 0
13:10	Reserved	Must be kept at reset value.
9:8	KEYM[1:0]	AES key size mode configuration, must be configured when BUSY=0 00: 128-bit key length 01: 192-bit key length 10: 256-bit key length 11: never use
7:6	DATAM[1:0]	Data swapping type mode configuration, must be configured when BUSY=0 00: No swapping 01: Half-word swapping 10: Byte swapping 11: Bit swapping
5:3	ALGM[2:0]	Encryption / decryption algorithm mode bit 0 to bit 2 000: All modes are disabled.

		100: AES-ECB with CAU_KEY0, 1, 2, 3. 111: AES key derivation for decryption mode. The input key must be same to that used in encryption. The BUSY bit is set until the process has been finished, and CAUEN is then cleared.
2	CAUDIR	CAU direction, must be configured when BUSY=0 0: encryption 1: decryption
1:0	Reserved	Must be kept at reset value.

18.8.2. Status register 0 (CAU_STAT0)

Address offset: 0x04

Reset value: 0x0000 0003

This register 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											BUSY	OFU	ONE	INF	IEM
											r	r	r	r	r

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	BUSY	Busy bit 0: No processing. This is because: - CAU is disabled by CAUEN=0 or the processing has been completed. - No enough data or no enough space in the input / output FIFO to perform a data block 1: CAU is processing data or key derivation.
3	OFU	Output FIFO is full 0: Output FIFO is not full 1: Output FIFO is full
2	ONE	Output FIFO is not empty 0: Output FIFO is empty 1: Output FIFO is not empty
1	INF	Input FIFO is not full 0: Input FIFO is full 1: Input FIFO is not full
0	IEM	Input FIFO is empty 0: Input FIFO is not empty

1: Input FIFO is empty

18.8.3. Data input register (CAU_DI)

Address offset: 0x08
Reset value: 0x0000 0000

The data input register is used to transfer plaintext or ciphertext blocks into the input FIFO for processing. The MSB is firstly written into the FIFO and the LSB is the last one. If the CAUEN is 0 and the input FIFO is not empty, when it is read, then the first data in the FIFO is popped out and returned. If the CAUEN is 1, the returned value is undefined. Once it is read, then the FIFO must be flushed.

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

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

Bits	Fields	Descriptions
31:0	DI[31:0]	Data input Write these bits will write data to input FIFO, read these bits will return input FIFO value if CAUEN is 0, or it will return an undefined value

18.8.4. Data output register (CAU_DO)

Address offset: 0x0C
Reset value: 0x0000 0000

The data output register is a read only register. It is used to receive plaintext or ciphertext results from the output FIFO. Similar to CAU_DI, the MSB is read at first while the LSB is read at last.

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

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

Bits	Fields	Descriptions
31:0	DO[31:0]	Data output

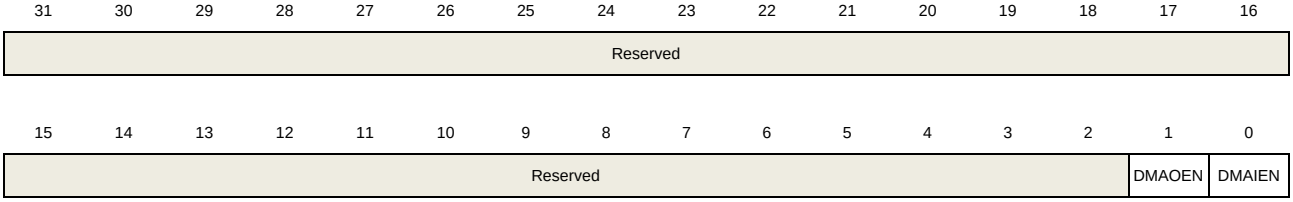
These bits are read only, read these bits return output FIFO value.

18.8.5. DMA enable register (CAU_DMAEN)

Address offset: 0x10

Reset value: 0x0000 0000

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



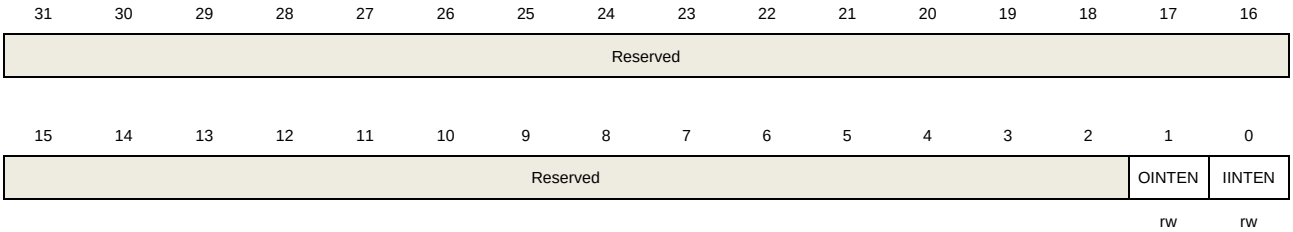
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DMAOEN	DMA output enable 0: DMA for OUT FIFO data is disabled 1: DMA for OUT FIFO data is enabled
0	DMAIEN	DMA input enable 0: DMA for IN FIFO data is disabled 1: DMA for IN FIFO data is enabled

18.8.6. Interrupt enable register (CAU_INTEN)

Address offset: 0x14

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	OINTEN	OUT FIFO interrupt enable 0: OUT FIFO interrupt is disable 1: OUT FIFO interrupt is enable
0	IINTEN	IN FIFO interrupt enable

0: IN FIFO interrupt is disable

1: IN FIFO interrupt is enable

18.8.7. Status register 1 (CAU_STAT1)

Address offset: 0x18

Reset value: 0x0000 0001

This register 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														OSTA	ISTA
														r	r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OSTA	OUT FIFO interrupt status 0: OUT FIFO interrupt status not pending 1: OUT FIFO interrupt status pending
0	ISTA	IN FIFO interrupt status 0: IN FIFO interrupt not pending 1: IN FIFO interrupt flag pending

18.8.8. Interrupt flag register (CAU_INTF)

Address offset: 0x1C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														OINTF	IINTF
														r	r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTF	OUT FIFO enabled interrupt flag 0: OUT FIFO Interrupt not pending 1: OUT FIFO Interrupt pending

0	IINTF	IN FIFO enabled interrupt flag 0: IN FIFO Interrupt not pending 1: IN FIFO Interrupt pending when CAUEN is 1

18.8.9. Key registers (CAU_KEY0..3(H / L))

Address offset: 0x20 to 0x3C
Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In AES-128 mode, KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[0:63], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[64:127].

In AES-192 mode, KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[0:63], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[64:127], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[128:191].

In AES-256 mode, KEY0H[31:0] || KEY0L[31:0] is used as AES_KEY[0:63], KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[64:127], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[128:191], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[192:255].

Note: "||" is a concatenation operator. For example, X || Y denotes the concatenation of two bit strings X and Y.

CAU_KEY0H

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
KEY0H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0H[15:0]															
w															

CAU_KEY0L

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
KEY0L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

KEY0L[15:0]

w

CAU_KEY1H

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
KEY1H[31:16]															

w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1H[15:0]															

w

CAU_KEY1L

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
KEY1L[31:16]															

w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1L[15:0]															

w

CAU_KEY2H

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
KEY2H[31:16]															

w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2H[15:0]															

w

CAU_KEY2L

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

KEY2L[31:16]
w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2L[15:0]															
w															

CAU_KEY3H

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
KEY3H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3H[15:0]															
w															

CAU_KEY3L

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
KEY3L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3L[15:0]															
w															

Bits	Fields	Descriptions
31:0	KEY0...3(H / L)	The key for AES

19. Hash Acceleration Unit (HAU)

19.1. Overview

The hash acceleration unit is used for information security. The secure hash algorithm (SHA-256) is supported for various applications. The digest will be computed and the length is 256 bits for a message up to (2^{64} - 1) bits computed by SHA-256 algorithm respectively.

The HAU is fully compliant implementation of the following standards:

- Federal Information Processing Standards Publication 180-4(FIPS PUB 180-4).
- Secure Hash Standard specifications (SHA-256).

19.2. Characteristics

- 32-bit AHB slave peripheral.
- High performance of computation of hash algorithms.
- Little-endian data representation.
- Multiple data types are supported, including no swapping, half-word swapping, byte swapping, and bit swapping with 32-bit data words.
- Automatic data padding to fill the 512-bit message block for digest computation.
- DMA transfer is supported.

19.3. HAU data type

The hash acceleration unit receives data words of 32 bits at a time, while they are processed in 512-bits blocks. For each input word, according to the data type, the data could be bit / byte / half-word / no swapped before they are transferred into the hash acceleration core. The same swapping operation should be also performed on the core output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian. However, the computation of SHA-256 are big-endian.

Figure 19-1. DATAM No swapping and Half-word swapping and *Figure 19-2. DATAM Byte swapping and Bit swapping* illustrate the data swapping according to different data types.

Figure 19-1. DATAM No swapping and Half-word swapping

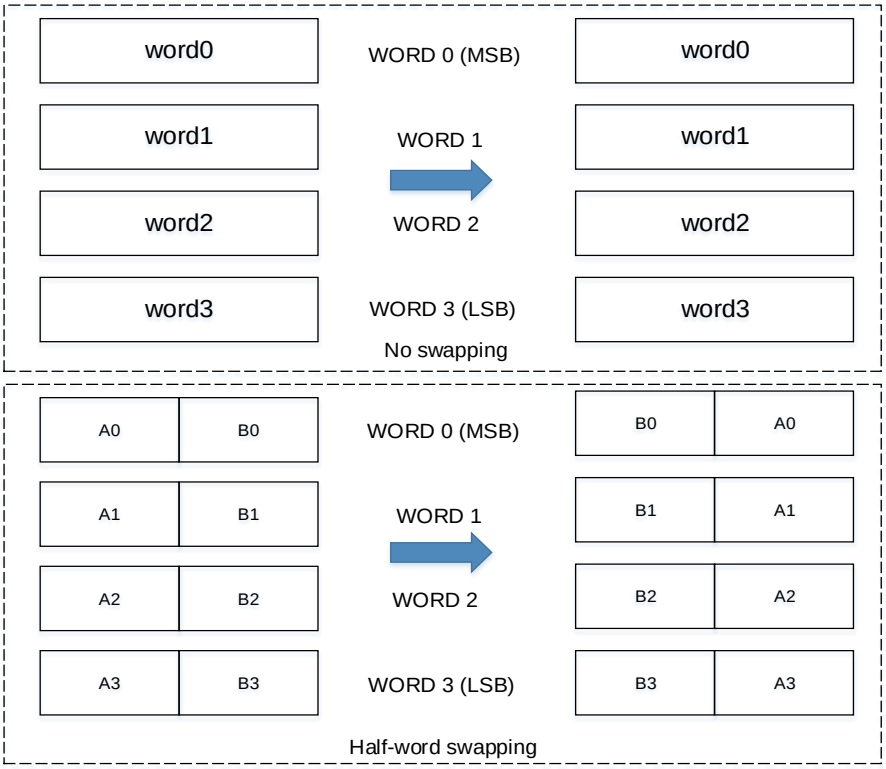
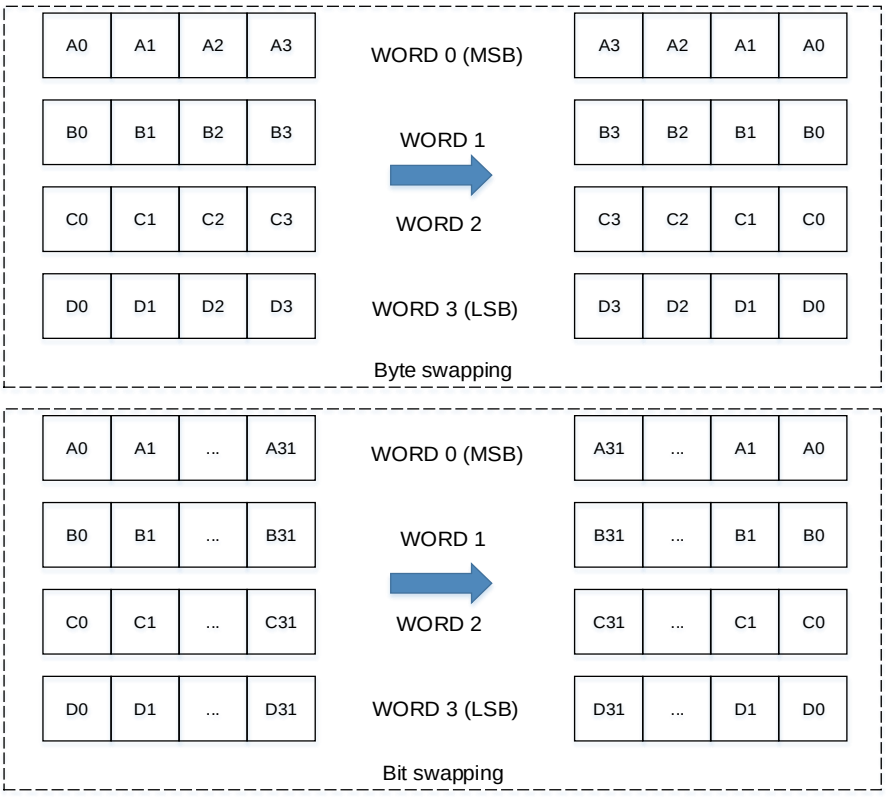


Figure 19-2. DATAM Byte swapping and Bit swapping

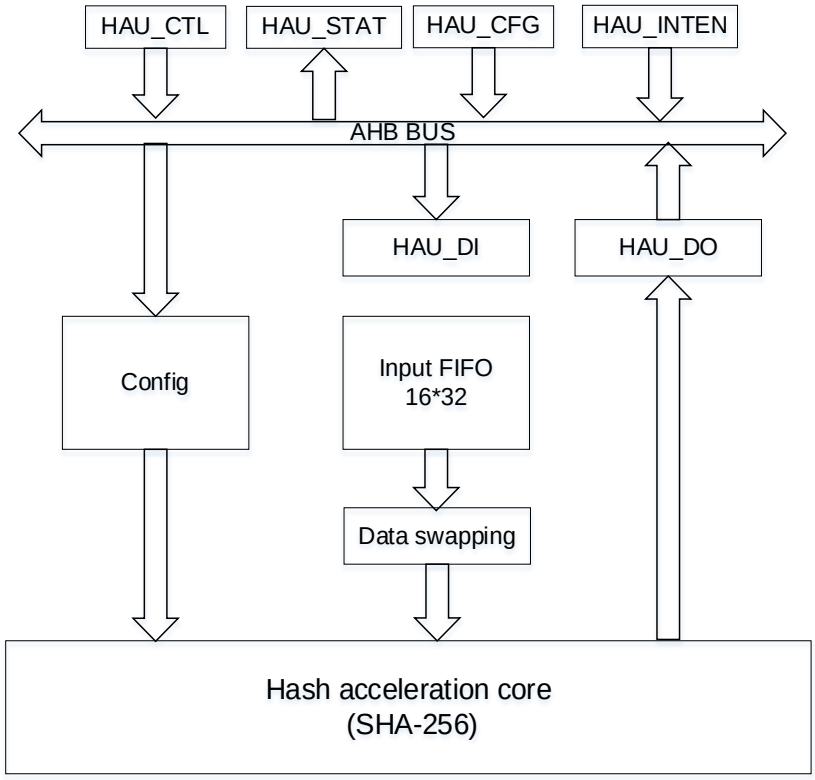


19.4. HAU core

The hash acceleration unit is used to compute condensed information of input messages with secure hash algorithms. The digest result has a length of 256 bits for a message up to (2^{64}-1) bits computed by SHA256 algorithm respectively. It can be used to generate or verify the signature of a message with a higher efficiency because of the much simpler of the information.

A message which need to be processed in the HAU should be considered as bit information. And the length is the number of bits of the message. The information security is ensured because that, to find the original message using the digest is computationally impossible and, the result will be completely different with any change to the input message.

Figure 19-3. HAU block diagram



19.4.1. Automatic data padding

The input message should be padded first so that the number of bits in the input of the HAU core can be an integral multiple of 512. First of all, a “1” is added to follow the last bit of the input message, and then several “0” should be padded to ensure the result modulo 512 is 448, at last, a 64-bit length information of input is added.

After the message padding is correctly performed, the VBL bits in the HAU_CFG register is configured as the 64-bit length value above, and CALEN bit in the HAU_CFG register can be set 1 to start the calculation of the digest of the last block.

Data Padding Example: The input message is “HAU”, which ASCII hexadecimal code is:

484155

Then the VBL bits in the HAU_CFG register is set as decimal 24 because of the valid bit length. A “1” is added at bit location 24 then, and several “0” are padded so that the result modulo 512 is 448, the hexadecimal result is as follows:

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000

After that, a 64-bit length information of the input message is padded, which hexadecimal value is 18, and the final result will be:

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000018

19.4.2. Digest computing

After data padding, for each block calculation of HAU, 512 bits are written into the HAU core by DMA or CPU. To start the processing of the HAU core, the peripheral must obtain the information as to whether the HAU_DI register contains the last bits of the message or not. This can be confirmed with the status of the input FIFO and the HAU_DI register.

When DMA is used to transfer data:

The status of the block transfer is automatically interpreted with the information from the DMA controller. And padding and digest computation are performed automatically as if CALEN bit in the HAU_CFG register is set as 1.

Note: If hash message is large files and multiple DMA transfers are needed, then MDS bit should be set as 1. And the VBL bits need to be set before the transfer. The CALEN bit is not set automatically after an intermediate DMA transfer completed. Only when the last DMA transfer is processing, the MDS bit is cleared so that the CALEN bit is automatically set after data transferring.

Otherwise, the MDS bit is set as 0. And the CALEN bit is set automatically after a DMA transfer. Also, VBL bits need to set before the DMA transfer.

When CPU is used to transfer data without DMA:

- The intermediate block computing can be started when HAU_DI is filled with another new word of the next block.
- The last block computing can be started when CALEN bit in the HAU_CFG register is 1.

19.4.3. Hash mode

After a message block of 512 bit has been received through the HAU_DI register and the input FIFO, the processor starts the calculation with the information from DMA or the status of the CALEN bit.

The results can be finally read from the HAU_DO0...7 registers.

19.5. HAU interrupt

There are two types of interrupt registers in HAU, which are both in HAU_STAT register. In HAU, the interrupt is used to indicate the situation of the input FIFO and the status of whether the digest calculation is completed.

Any of interrupts can be enabled or disabled by configuring the HAU interrupt enable register HAU_INTEN. Value 1 of the register bits enable the interrupts.

19.5.1. Input FIFO interrupt

When the processing of data pushed in the input FIFO is completed, then DIF is asserted. If input FIFO interrupt is enabled, when DIF is asserted, input FIFO interrupt will be asserted.

19.5.2. Calculation completion interrupt

When the digest calculation is finished, then CCF is asserted. If calculation completion interrupt is enabled, when CCF is asserted, calculation completion interrupt will be asserted.

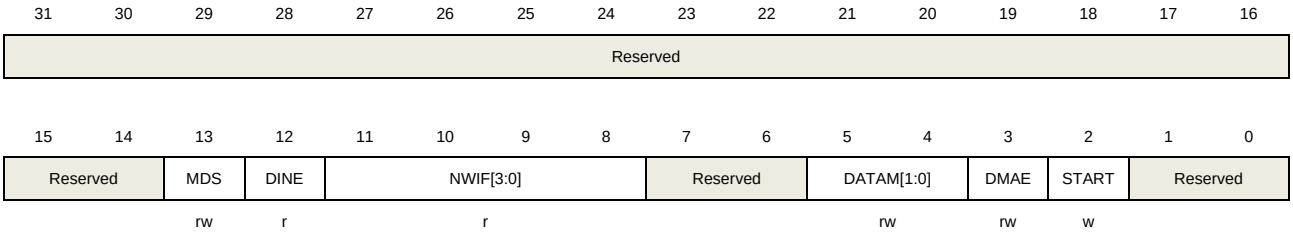
19.6. Register definition

HAU base address: 0x4002 3800

19.6.1. Control register (HAU_CTL)

Address offset: 0x00
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	MDS	Multiple DMA Selection Set this bit if hash message is large files and multiple DMA transfers are needed. 0: Single DMA transfers needed and CALEN bit is automatically set at the end of a DMA transfer 1: Multiple DMA transfers needed and CALEN bit is not automatically set at the end of a DMA transfer
12	DINE	DI register not empty 0: The DI register is empty 1: The DI register is not empty **Note:** This bit is cleared when START bit or CALEN bit is set as 1.
11:8	NWIF[3:0]	Number of words in the input FIFO **Note:** These bits are cleared when START bit set or a digest calculation starts (CALEN bit is set as 1, or DMA end of transfer)
7:6	Reserved	Must be kept at reset value.
5:4	DATAM[1:0]	Data type mode Defines the format of the data entered into the HAU_DI register: 00: No swapping. The data written to HAU_DI is direct write to FIFO without swapping. 01: Half-word swapping. The data written into HAU_DI need half-word swapping before write to FIFO. 10: Bytes swapping. The data written into HAU_DI need bytes swapping before write to FIFO. 11: Bit swapping. The data written into HAU_DI need bytes swapping before write

		to FIFO.
3	DMAE	DMA enable 0: DMA disabled 1: DMA enabled **Note:** 1. This bit is cleared when transferring the last data of the message, but not cleared because of START. 2. When DMA is transferring, writing 0 to this bit will not stop the current transfer until the transfer is completed or START is set as 1.
2	START	Start the digest calculation 1: Start the digest of a new message 0: No effect **Note:** Reading this bit always returns 0.
1:0	Reserved	Must be kept at reset value.

19.6.2. Data input register (HAU_DI)

Address offset: 0x04
Reset value: 0x0000 0000

The data input register is used to transfer message with 512-bit blocks into the input FIFO for processing. Any new write operation to this register will be extended while the digest calculation is in process until it has been finished.

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

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

Bits	Fields	Descriptions
31:0	DI[31:0]	Message data input When write to these registers, the current content pushed to IN FIFO and new value updates. When read, returns the current content.

19.6.3. Configuration register (HAU_CFG)

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							CALEN	Reserved			VBL[4:0]				
							w				rw				

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	CALEN	Digest calculation enable 0: No calculation 1: Start data padding with VBL prepared previously. Start the calculation of the last digest **Note:** Reading this bit always returns 0.
7:5	Reserved	Must be kept at reset value.
4:0	VBL[4:0]	Valid bits length in the last word 0x00: All 32 bits of the last data written to HAU_DI after data swapping are valid 0x01: Only bit [31] of the last data written to HAU_DI after data swapping are valid 0x02: Only bits [31:30] of the last data written to HAU_DI after data swapping are valid 0x03: Only bits [31:29] of the last data written to HAU_DI after data swapping are valid ... 0x1F: Only bits [31:1] of the last data written to HAU_DI after data swapping are valid **Note:** These bits must be configured before setting the CALEN bit.

19.6.4. Data output register (HAU_DO0..7)

Address offset: 0x0C
Reset value: 0x0000 0000

The data output registers are read only registers. They are used to receive results from the output FIFO. And they are reset by the START bit. Any read access when calculating will be extended until the calculation is completed.

HAU_DO0

Address offset: 0x0C and 0x310
Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO0[31:16]															
r															

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

DO0[15:0]

r

HAU_DO1

Address offset: 0x10 and 0x314

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO1[31:16]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO1[15:0]															

r

HAU_DO2

Address offset: 0x14 and 0x318

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO2[31:16]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO2[15:0]															

r

HAU_DO3

Address offset: 0x18 and 0x31C

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO3[31:16]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO3[15:0]															

r

HAU_DO4

Address offset: 0x1C and 0x320

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

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

DO4[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO4[15:0]															
r															

HAU_DO5

Address offset: 0x324

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO5[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO5[15:0]															
r															

HAU_DO6

Address offset: 0x328

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO6[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO6[15:0]															
r															

HAU_DO7

Address offset: 0x32C

Reset value: 0x0000 0000

These registers have to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO7[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO7[15:0]															
r															

Bits	Fields	Descriptions
31:0	DO0..7[31:0]	Message digest result of hash algorithm

19.6.5. Interrupt enable register (HAU_INTEN)

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														CCIE	DIIE
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CCIE	Calculation completion interrupt enable 0: Calculation completion interrupt is disabled 1: Calculation completion interrupt is enabled
0	DIIE	Data input interrupt enable 0: Data input interrupt is disabled 1: Data input interrupt is enabled

19.6.6. Status and flag register (HAU_STAT)

Address offset: 0x24

Reset value: 0x0000 0001

This register 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												BUSY	DMAS	CCF	DIF
												r	r	rc_w0	rc_w0

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	BUSY	Busy bit 0: No processing 1: Data block is in process
2	DMAS	DMA status 0: DMA is disabled (DMAE = 0) and no transfer is processing

		1: DMA is enabled (DMAE = 1) or a transfer is processing
1	CCF	Digest calculation completion flag 0: Digest calculation is not completed 1: Digest calculation is completed
0	DIF	Data input flag 0: A data is written to data input register 1: A data processing is completed (only the data in input FIFO will be processed)

20. Analog-to-digital converter (ADC)

20.1. Overview

A 12-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip. ADC0 has 16 external channels, 2 internal channel (temperature sensor, V_{REFINT} inputs channel), ADC1 has 18 external channels, ADC2 has 17 external channels.

Analog watchdog allows the application to detect if the input voltages exceed the user-set high and low thresholds.

All ADC sampling channels support a variety of operation modes. After sampling and conversion, the conversion results can be stored in the corresponding data registers according to the least significant bit (LSB) alignment or the most significant bit (MSB) alignment.

An on-chip hardware oversample scheme improves performances and reduces the computational burden of MCU.

20.2. Characteristics

- High performance.
  - ADC sampling resolution: 12-bit, 10-bit, 8-bit or 6-bit.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit.
  - DMA support for routine sequence and inserted sequence.
- Analog input channels.
  - 16 external analog inputs for ADC0, 18 external analog inputs for ADC1, 17 external analog inputs for ADC2.
  - 1 channel for internal temperature sensor (V_{SENSE}) for ADC0.
  - 1 channel for internal reference voltage (V_{REFINT}) for ADC0.
- Start-of-conversion can be initiated.
  - By software.
  - By TRIGSEL.
- Operation modes.
  - Converts a single channel or scans a sequence of channels.
  - Single operation mode converts selected inputs once per trigger.
  - Continuous operation mode converts selected inputs continuously.
  - Discontinuous operation mode.
  - SYNC mode(the device with two or more ADCs).
- Conversion result threshold monitor function: analog watchdog.
- Interrupt generation.
  - At the end of sequence conversion.
  - Analog watchdog event.

- Oversampler.
  - 16-bit data register.
  - Oversampling ratio adjustable from 2x to 256x.
  - Programmable data shift up to 8-bit.
- Channel input voltage range: $V_{REFN} \leq V_{IN} \leq V_{REFP}$.

20.3. Pins and internal signals

Figure 20-1. ADC module block diagram shows the ADC block diagram. *Table 20-1. ADC internal input* gives the ADC internal signals and *Table 20-2. ADC input pins definition* gives the ADC pin description.

Table 20-1. ADC internal input channels

Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{REFINT}	Internal voltage reference output voltage

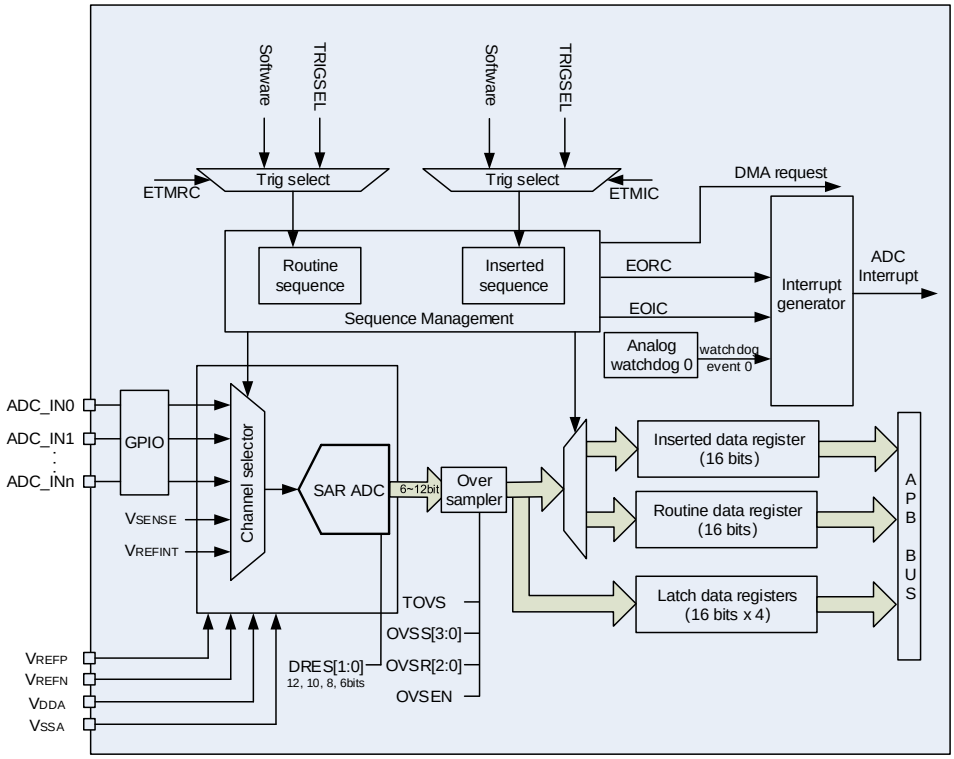
Table 20-2. ADC input pins definition

Name	Description
V_{DDA}	Analog power supply equals to V_{DD}
V_{SSA}	Ground for analog power supply equals to V_{SS}
V_{REFP}	The positive reference voltage for the ADC.
V_{REFN}	The negative reference voltage for the ADC.
ADCx_IN[17:0]	Up to 18 external analog channels

Note: V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS} respectively.

20.4. Function overview

Figure 20-1. ADC module block diagram



20.4.1. ADC clock

The maximum frequency of ADC is 42MHz. The CK_ADC clock is synchronous with the AHB and APB2 clock and provided by the clock controller. ADC clock can be divided and configured by RCU controller.

Refer to ***Reset and clock unit (RCU)*** for more information on ADC clock generation.

20.4.2. ADC enable

The ADCON bit on the ADC_CTL1 register is the enable switch of the ADC module. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is reset, the analog submodule will be put into power down mode. After ADC is enabled, you need delay t_{SU} time for sampling (the value of t_{SU} please refer to the device datasheet).

20.4.3. Routine and inserted sequence

The channel management circuit can organize the sampling conversion channels into two sequences: a routine sequence and an inserted 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.

The inserted sequence supports up to 4 channels, and each channel is called inserted channel. The IL[1:0] bits in the ADC_ISQ register specify the total conversion sequence length. The ADC_ISQ register specifies the selected channels of the inserted sequence.

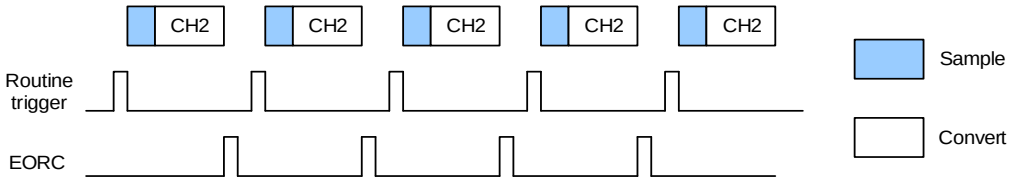
Note: Although the ADC supports 18 multiplexed channels, the maximum length of the routine sequence is only 16, and the maximum length of the inserted sequence is only 4.

20.4.4. Operation modes

Single operation mode

This mode can be running on both routine and inserted sequence. In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits of ADC_RSQ2 or the channel specified in the ISQ3[4:0] bits of ADC_ISQ. When the ADCON has been set high, the ADC samples and converts a single channel, once the corresponding software trigger or TRIGSEL trigger is active.

Figure 20-2. Single operation mode



After conversion of a single routine channel, the conversion data will be stored in the ADC_RDATA register, the EORC will be set. An interrupt will be generated if the EORCIE bit is set.

After conversion of a single inserted channel, the conversion data will be stored in the ADC_IDATA register, the EOIC will be set. An interrupt will be generated if the EOICIE bit is set.

Software procedure for single operation mode of routine sequence:

1. Make sure the DISRC, SM in the ADC_CTL0 register and CTN bit in the ADC_CTL1 register are reset.
2. Configure RSQ0[4:0] in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
5. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence.
6. Wait the EORC flag to be set.
7. Read the converted data in the ADC_RDATA register.
8. Clear the EORC flag by writing 0 to it.

Software procedure for single operation mode of inserted sequence:

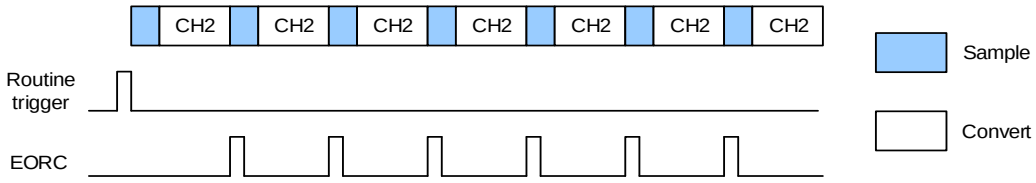
1. Make sure the DISIC, SM in the ADC_CTL0 register are reset.

2. Configure ISQ3[4:0] with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
5. Set the SWICST bit, or generate an TRIGSEL trigger for the inserted sequence.
6. Wait the EOIC flags to be set.
7. Read the converted data in the ADC_IDATA register.
8. Clear the EOIC flag by writing 0 to it.

Continuous operation mode

This mode can be running on routine sequence. The continuous operation mode will be enabled when CTN bit in the ADC_CTL1 register is set. In this mode, the ADC performs conversion on the channel specified in the RSQ0[4:0]. When the ADCON has been set high, the ADC samples and converts specified channel, once the corresponding software trigger or TRIGSEL trigger is active. The conversion data will be stored in the ADC_RDATA register.

Figure 20-3. 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[4:0] in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
5. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence.
6. Wait the EORC flag to be set.
7. Read the converted data in the ADC_RDATA register.
8. Clear the EORC flag by writing 0 to it.
9. Repeat steps 6~8 as soon as the conversion is in need.

To get rid of checking EORC bit, DMA can be used to transfer the converted data:

1. Set the CTN and RDMA bit in the ADC_CTL1 register.
2. Configure RSQ0[4:0] in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
5. Prepare the DMA module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence.

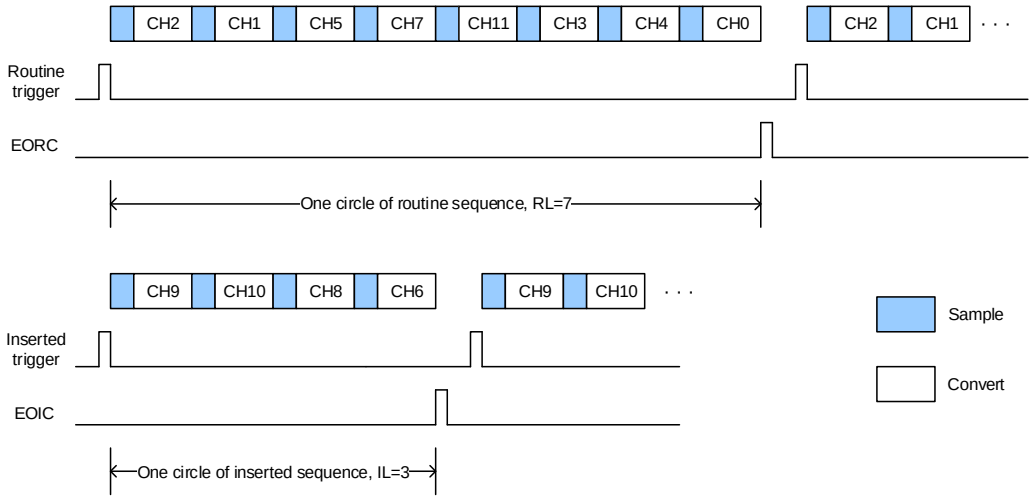
Scan operation mode

The scan operation mode will be enabled when SM bit in the ADC_CTL0 register is set. In this mode, the ADC performs conversion on the channels with a specific sequence specified

in the ADC_RSQ0~ADC_RSQ2 registers or ADC_ISQ register. When the ADCON has been set high, the ADC samples and converts specified channels one by one in the routine or inserted sequence till the end of the routine or inserted sequence, once the corresponding software trigger or external trigger is active. The conversion data will be stored in the ADC_RDATA or ADC_IDATA register. After conversion of the routine or inserted sequence, the EORC or EOIC will be set. An interrupt will be generated if the EORCIE or EOICIE bit is set. The RDMA or IDMA bit in ADC_CTL1 register can be set when the routine or inserted 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 20-4. Scan operation mode, continuous disable



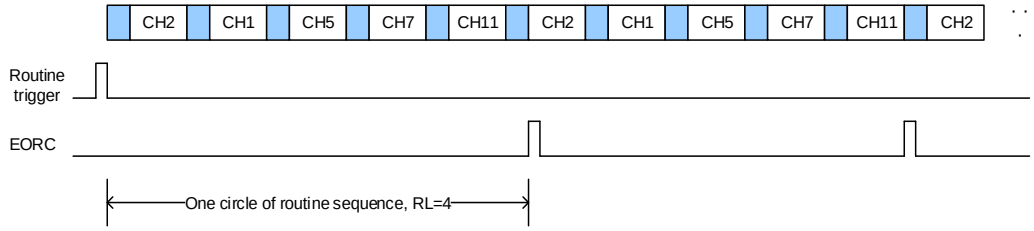
Software procedure for scan operation mode on a routine sequence:

1. Set the SM bit in the ADC_CTL0 register and the RDMA bit in the ADC_CTL1 register.
2. Configure ADC_RSQx and ADC_SAMPTx registers.
3. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_RDATA.
5. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence.

Software procedure for scan operation on an inserted sequence:

1. Set the SM bit in the ADC_CTL0 register and the IDMA bit in the ADC_CTL1 register.
2. Configure ADC_ISQ and ADC_SAMPTx registers.
3. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_IDATA;
5. Set the SWICST bit, or generate an TRIGSEL trigger for the inserted sequence.

Figure 20-5. Scan operation mode, continuous enable



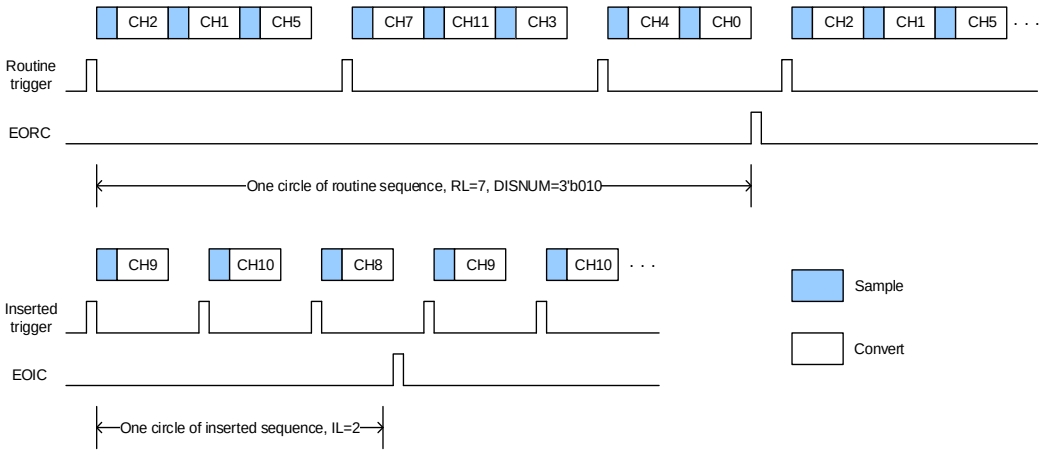
Discontinuous operation mode

For routine sequence, the discontinuous operation mode will be enabled when DISRC bit in the ADC_CTL0 register is set. In this mode, the ADC performs a short sequence of n conversions (n does not exceed 8) which is a part of the conversions selected in the ADC_RSQ0~ADC_RSQ2 registers. The value of n is configured by the DISNUM[2:0] bits in the ADC_CTL0 register. When the corresponding software trigger or 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 EORC will be set after every circle of the routine sequence. An interrupt will be generated if the EORCIE bit is set.

For inserted sequence, the discontinuous operation mode will be enabled when DISIC bit in the ADC_CTL0 register is set. In this mode, the ADC performs one conversion which is a part of the conversions selected in the ADC_ISQ register. When the corresponding software trigger or external trigger is active, the ADC samples and converts the next channel selected in the ADC_ISQ register until all the channels in the inserted sequence are done. The EOIC will be set after every circle of the inserted sequence. An interrupt will be generated if the EOICIE bit is set.

The routine and inserted sequences cannot both work in discontinuous operation mode. Only one sequence conversion can be set in discontinuous operation mode at a time.

Figure 20-6. 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 RDMA 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 ETMRC[1:0] bits in the ADC_CTL1 register if in need.
5. Prepare the DMA module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence.
7. Repeat step6 if in need.

Software procedure for discontinuous operation mode on an inserted sequence:

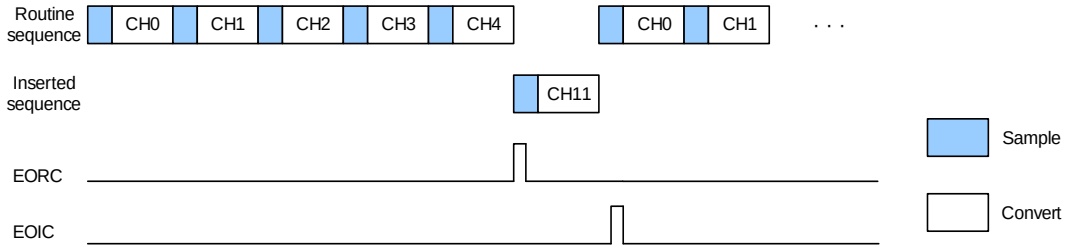
1. Set the DISIC bit in the ADC_CTL0 register and the IDMA bit in the ADC_CTL1 register.
2. Configure ADC_ISQ and ADC_SAMPTx registers.
3. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_IDATA;
5. Set the SWICST bit, or generate an TRIGSEL trigger for the inserted sequence.
6. Repeat step4 if in need.

20.4.5. Inserted sequence management

Auto-insertion

The inserted sequence is automatically converted after the routine sequence when the ICA bit in ADC_CTL0 register is set. In this mode, external trigger on inserted sequence cannot be enabled. A sequence of up to 20 conversions programmed in the ADC_RSQ0~ADC_RSQ2 and ADC_ISQ registers can be used to convert in this mode. In addition to the ICA bit, if the CNT bit is also set, routine sequence followed by inserted sequence are continuously converted.

Figure 20-7. Auto-insertion, CNT = 1

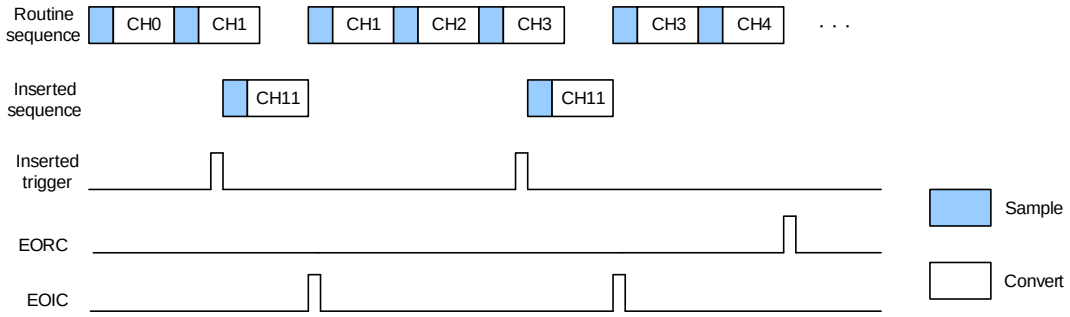


The auto insertion mode cannot be enabled when the discontinuous operation mode is set.

Triggered insertion

If the ICA bit is cleared, the triggered insertion occurs if a software or external trigger occurs during the routine sequence conversion. In this situation, the ADC aborts from the current conversion and starts the conversion of inserted sequence. After the inserted sequence is done, the routine sequence channel conversion is resumed from the last aborted conversion.

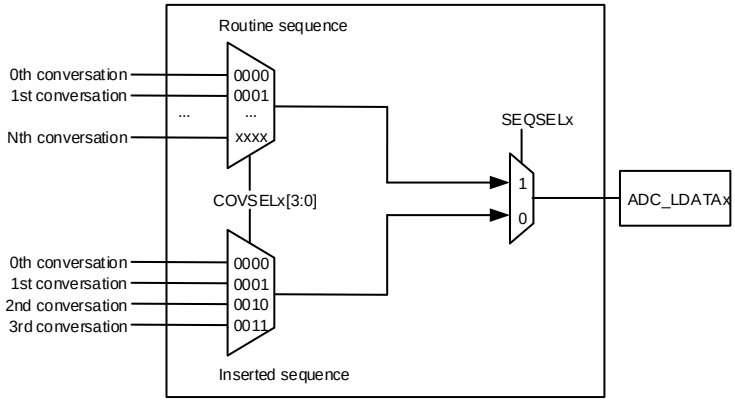
Figure 20-8. Triggered insertion



20.4.6. Convert data latch

The ADC has 4 latch data registers, ADC_LDATAx (x=0…3), which can latch the data from a completed conversion in a routine or inserted sequence. The control logic for the latch data registers is shown in ***Figure 20-9. Data latch***.

Figure 20-9. Data latch

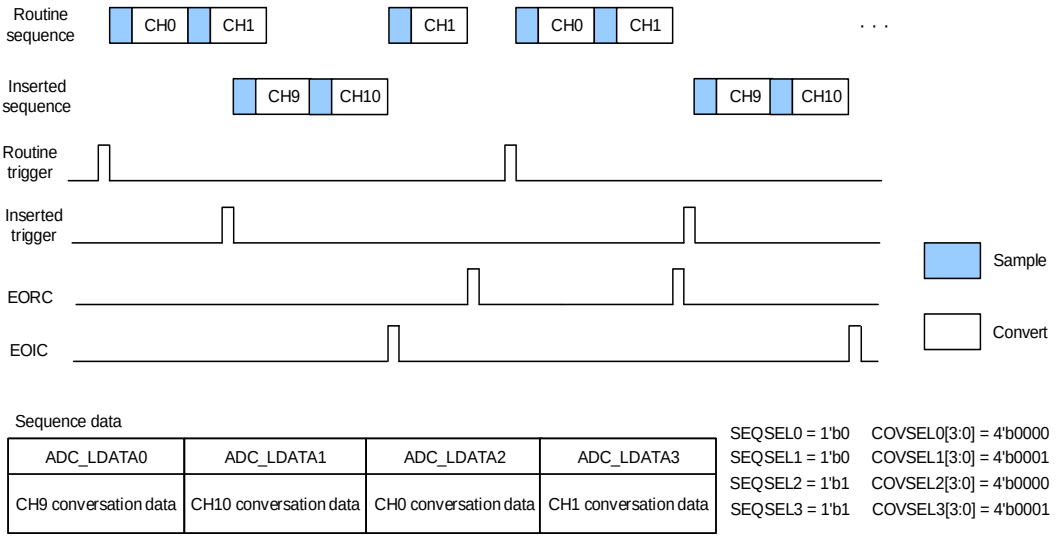


The data to be latched from the routine or inserted sequence is selected through the SEQSELx (x=0…3) bits in the ADC_LDCTL register, while the COVSELx[3:0] bits select which conversion result in the sequence to latch. Since the maximum length of the inserted sequence is 4, COVSELx[3:0] can only be configured up to 4'b0011 for the inserted sequence. The maximum length of the routine sequence is 16, COVSELx[3:0] can be configured up to 4'b1111 for the routine sequence.

ADC_LDATAx (x=0…3) registers are set to store the conversion results of the inserted sequence or routine sequence and default store inserted sequence xth conversion results.

Therefore, in addition to using DMA to transfer conversion data from the ADC_RDATA or ADC_IDATA registers, data can also be directly read from the latch data registers, offering greater flexibility for consecutive conversions in a conversion sequence.

Figure 20-10. Sequence data from ADC_LDATAx



Software procedure for triggered insertion mode and read the sequence data from ADC_LDATAx:

1. Set the SM bit in the ADC_CTL0 register;
2. Configure ADC_RSQx, ADC_ISQ and ADC_SAMPTx registers;
3. Configure SEQSEL2, COVSEL2[3:0], SEQSEL3 and COVSEL3[3:0] bits in the ADC_LDCTL register for the routine sequence;
4. Configure SEQSEL0, COVSEL0[3:0], SEQSEL1 and COVSEL1[3:0] bits in the ADC_LDCTL register for the inserted sequence;
5. Configure ETMRC[1:0] and ETMIC[1:0] bits in the ADC_CTL1 register if it is needed;
6. Set the SWRCST bit, or generate an TRIGSEL trigger for the routine sequence;
7. Set the SWICST bit, or generate an TRIGSEL trigger for the inserted sequence;
8. Wait for EORC or EOIC flags to be set;
9. If EORC flag is set, read the converted result of the routine sequence in the ADC_LDATA2 and ADC_LDATA3 register;
10. Clear the EORC flag by writing 0.
11. If EOIC flag is set, read the converted result of the inserted sequence in the ADC_LDATA0 and ADC_LDATA1 register;
12. Clear the EOIC flag by writing 0.

20.4.7. Conversion result threshold monitor

Analog watchdog 0

The analog watchdog 0 is enabled when the RWD0EN and IWD0EN bits in the ADC_CTL0 register are set for routine and inserted sequences respectively. This function is used to monitor whether the conversion result exceeds the thresholds. The WD0E bit in ADC_STAT register will be set if the conversion result exceeds the thresholds. An interrupt will be generated if the WD0EIE bit is set. The ADC_WD0HT and ADC_WD0LT registers are used to specify the high and low threshold. The comparison is done before the alignment, so the

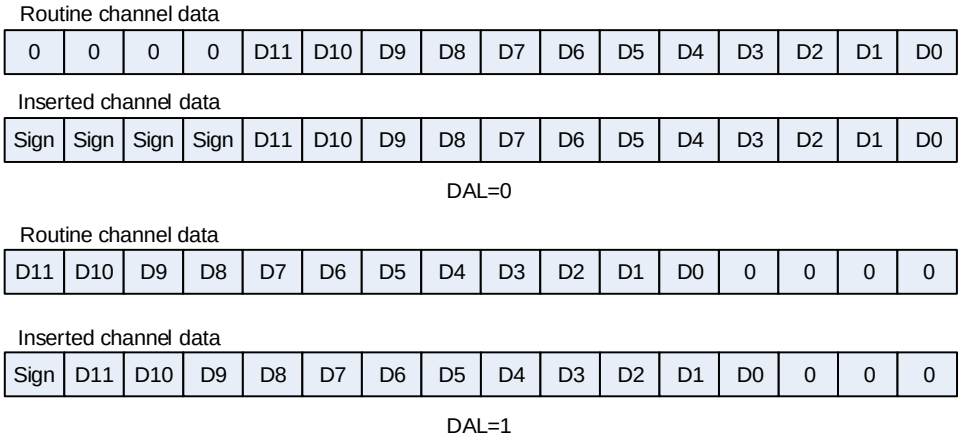
threshold value is independent of the alignment, which is specified by the DAL bit in the ADC_CTL1 register. One or more channels, which are select by the RWD0EN, IWD0EN, WD0SC and WD0CHSEL[4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog 0.

20.4.8. Data storage mode

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

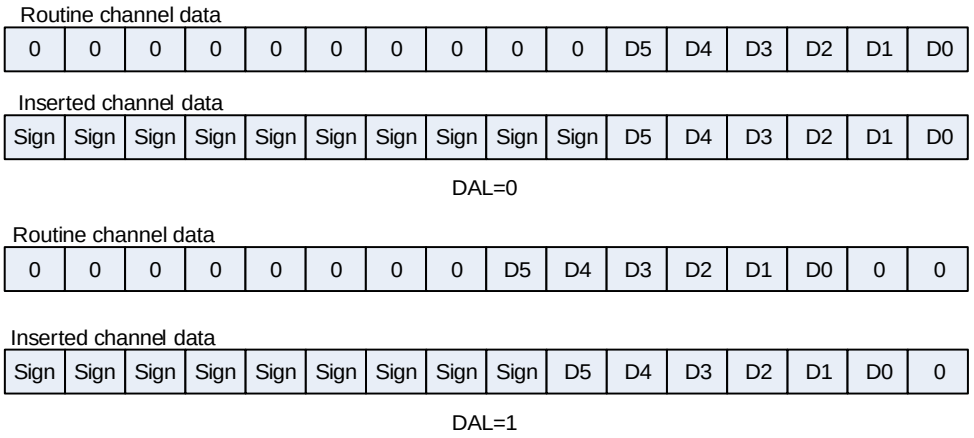
After being decreased by the user-defined offset written in the ADC_IOFFx registers, the inserted sequence data value may be a negative value. The sign bit is extended.

Figure 20-11. Data storage mode of 12-bit resolution



6-bit resolution data alignment is different from 12-bit/10-bit/8-bit resolution data alignment, shown as ***Figure 20-12. Data storage mode of 6-bit resolution***.

Figure 20-12. Data storage mode of 6-bit resolution



20.4.9. Sample time configuration

The number of CK_ADC cycles which is used to sample the input voltage can be specified by the SPTn[2:0] bits in the ADC_SAMPT0 and ADC_SAMPT1 registers. A different sample

time can be specified for each channel. For 12-bits resolution, the total sampling and conversion time is “sampling time + 12.5” CK_ADC cycles.

For example:

CK_ADC = 42MHz and sample time is 1.5 cycles, the total conversion time is “1.5+12.5” CK_ADC cycles, that means 0.333us.

20.4.10. External trigger configuration

The conversion of routine or inserted sequence can be triggered by rising edge of TRIGSEL triggers or software. The ETMRC[1:0] and ETMIC[1:0] bits in the ADC_CTL1 register control the trigger modes of routine and inserted sequence respectively.

Table 20-3. External trigger mode and type

ETMRC[1:0]/ETMIC[1:0]	Trigger Source	Trigger Type
00, 01, 10	TRIGSEL	Signal from TRIGSEL
11	SWRCST	Software trigger

20.4.11. DMA request

The DMA request, which is enabled by the RDMA or IDMA bit of ADC_CTL1 register, is used to transfer data of routine or inserted sequence for conversion of more than one channel. The ADC generates a DMA request at the end of conversion of a routine or inserted channel. When this request is received, the DMA will transfer the converted data from the ADC_RDATA or ADC_IDATA register to the destination location which is specified by the user.

20.4.12. ADC internal channels

When the TSVEN bit of ADC_CTL1 register is set, the temperature sensor channel (ADC0_CH16) is enabled. When the INREFEN bit of ADC_CTL1 register is set, the V_{REFINT} channel (ADC0_CH17) is enabled. The temperature sensor can be used to measure the ambient temperature of the device. The sensor output voltage can be converted into a digital value by ADC. The sampling time for the temperature sensor is recommended to be set to at least t_{s_temp} μs (please refer to the datasheet). When this sensor is not in use, it can be put in power down mode by resetting the TSVEN bit.

The output voltage of the temperature sensor changes linearly with temperature. Because 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 instead of absolute temperature. When it is used to detect accurate temperature, an external temperature sensor part should be used to calibrate the offset error.

To use the temperature sensor:

1. Configure the conversion sequence (ADC0_IN16) and the sampling time (t_{s_temp} μs) for the channel.

2. Enable the temperature sensor by setting the TSVEN 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 equation ***(20-1)***:

$$\text{Temperature (°C)} = \frac{V_{25} - V_{temperature}}{\text{Avg_Slope}} + 25 \quad (20\text{-}1)$$

$V_{temperature}$: The output voltage of temperature sensor.

V_{25}: Internal temperature sensor output voltage at 25°C, and the typical value please refer to the datasheet.

Avg_Slope: Average slope for curve between temperature vs. internal temperature sensor output voltage, the typical value please refer to the datasheet.

The internal reference voltage (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and Comparators. V_{REFINT} is internally connected to the ADC0_IN17 input channel.

20.4.13. Programmable resolution (DRES)

The resolution is configured by programming the DRES[1:0] bits in the ADC_OVSAMPCTL register. For applications that do not require high data accuracy, lower resolution allows faster conversion time. The DRES[1:0] bits must only be changed when the ADCON bit is reset. The result of the conversion is always 12 bits wide and any unused LSB bits are read as zeroes. Lower resolution reduces the conversion time needed for the successive approximation steps as shown in ***Table 20-4. tCONV timings depending on resolution***.

Table 20-4. t_{CONV} timings depending on resolution

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=42MHz	t_s(min) (ADC clock cycles)	t_{ADC} (ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=42MHz
12	12.5	298 ns	1.5	14	333 ns
10	10.5	250 ns	1.5	12	286 ns
8	8.5	202 ns	1.5	10	238 ns
6	6.5	155 ns	1.5	8	190 ns

20.4.14. On-chip hardware oversampling

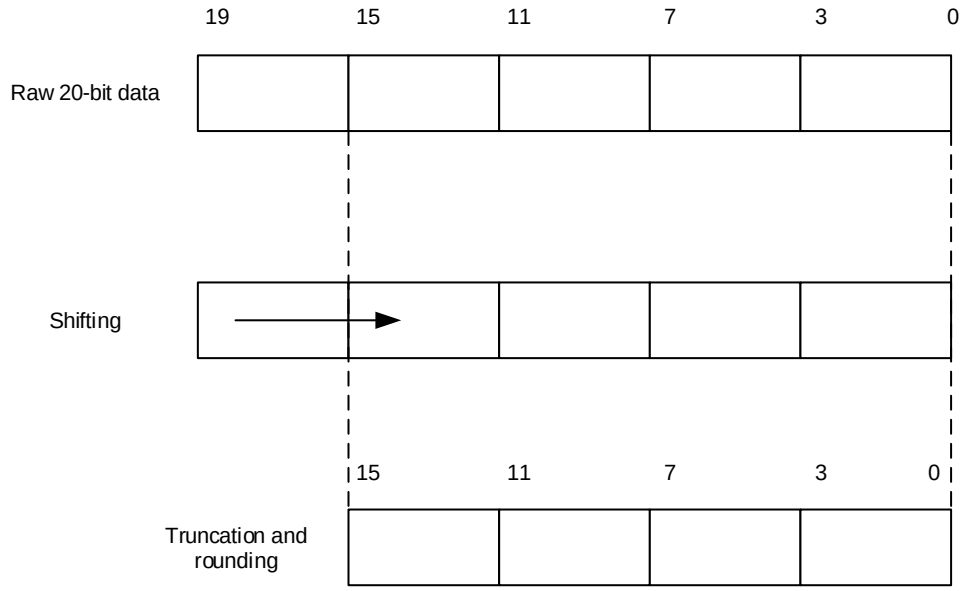
The on-chip hardware oversampling circuit performs data preprocessing to offload the CPU. It can handle multiple conversions and average them into a single data with increased data width, up to 16-bit. It provides a result with the following form, where N and M can be adjusted, and $D_{out}(n)$ is the n-th output digital signal of the ADC:

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{N-1} D_{out}(n) \quad (20\text{-}2)$$

The on-chip hardware oversampling circuit performs the following functions: summing and bit right shifting. The oversampling ratio N is defined by the OVSR[2:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to 256x. The division coefficient M means bit right shifting up to 8-bit. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

Summation units can produce up to 20 bits (256 x 12-bit), which is first shifted right. The upper bits of the result are then truncated, keeping only the 16 least significant bits rounded to the nearest value using the least significant bits left apart by the shifting, before being finally transferred into the data register, as shown in ***Figure 20-13. 20-bit to 16-bit result truncation***.

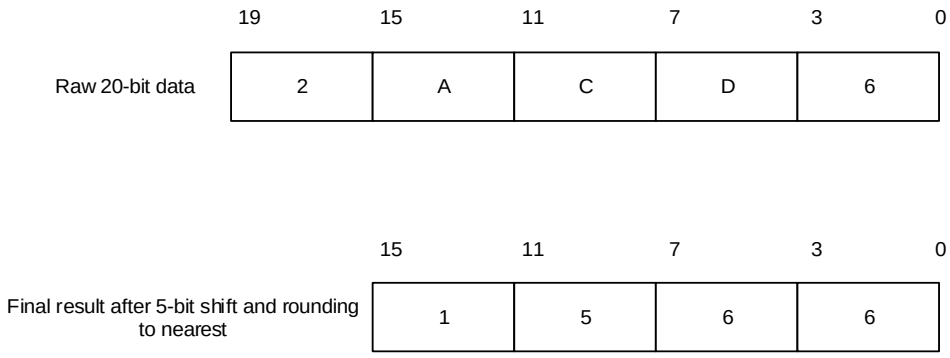
Figure 20-13. 20-bit to 16-bit result truncation



Note: If the intermediate result after the shifting exceeds 16 bits, the upper bits of the result are simply truncated.

Figure 20-14. Numerical example with 5-bits shift and rounding shows a numerical example of the processing, from a raw 20-bit accumulated data to the final 16-bit result.

Figure 20-14. Numerical example with 5-bits shift and rounding



The ***Table 20-5. Maximum output results for N and M (Grayed values indicates truncation)*** gives the data format for the various N and M combination, for a raw conversion data equal to 0xFFF.

Table 20-5. Maximum output results for N and M (Grayed values indicates truncation)

Oversampling ratio	Max Raw data	No-shift OVSS=0000	1-bit shift OVSS=0001	2-bit shift OVSS=0010	3-bit shift OVSS=0011	4-bit shift OVSS=0100	5-bit shift OVSS=0101	6-bit shift OVSS=0110	7-bit shift OVSS=0111	8-bit shift OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

The conversion timings in oversampled mode do not change compared to standard conversion mode: the sampling time remains equal throughout the oversampling sequence. New data is supplied every N conversions, and the equivalent delay is equal to:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (20\text{-}3)$$

20.5. ADC sync mode

In devices with more than one ADC, the ADC sync mode can be used.

In ADC sync mode, the conversion starts alternately or simultaneously triggered by ADC0 to ADC1, according to the sync mode configured by the SYNCM[3:0] bits in ADC0_CTL0 register.

In sync mode, when the conversion is configured to be triggered by an external event, the ADC1 must be configured as triggered by the software.

The modes in ***Table 20-6. ADC sync mode table*** can be configured.

In ADC sync mode, the RDMA bit must be set even if it is not used. The converted data of ADC1 routine channel can be read from the ADC0 routine data register (ADC0_RDATA).

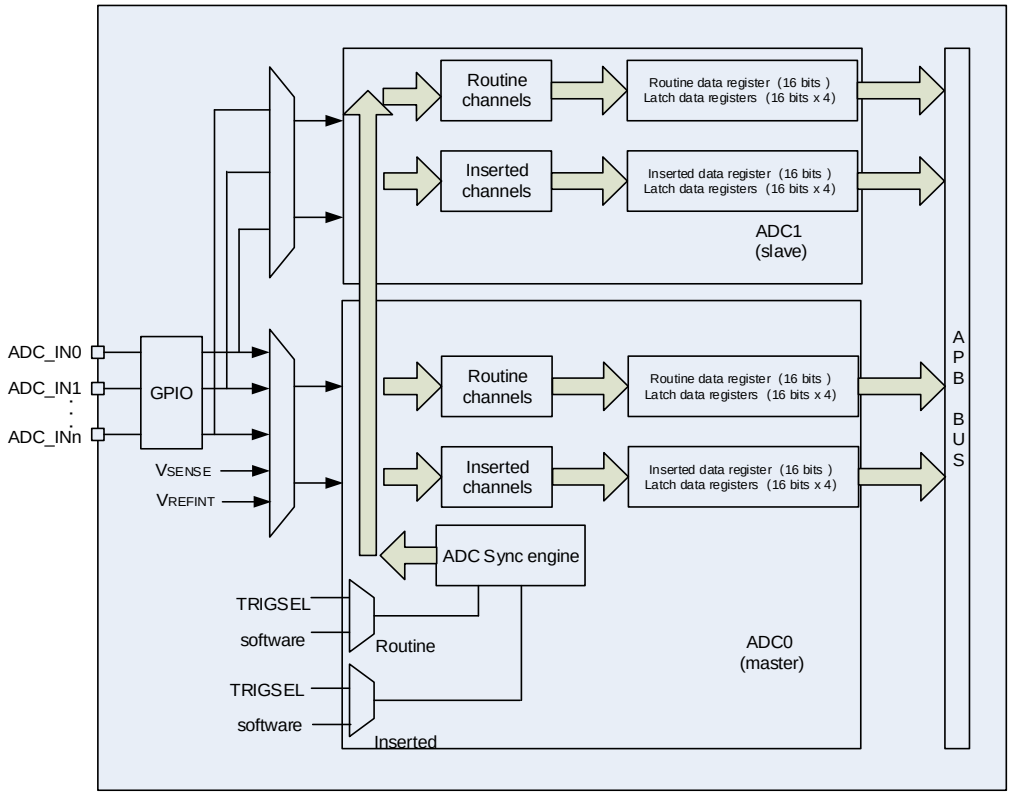
Table 20-6. ADC sync mode table

SYNCM[3: 0]	Mode
0000	Free mode
0001	ADC0 and ADC1 work in combined routine parallel & inserted parallel mode
0010	ADC0 and ADC1 work in combined routine parallel & inserted trigger rotation mode
0011	ADC0 and ADC1 work in combined inserted parallel & routine follow-up fast mode
0100	ADC0 and ADC1 work in combined inserted parallel & routine follow-up slow mode
0101	ADC0 and ADC1 work in inserted parallel mode
0110	ADC0 and ADC1 work in routine parallel mode
0111	ADC0 and ADC1 work in routine follow-up fast mode
1000	ADC0 and ADC1 work in routine follow-up slow mode
1001	ADC0 and ADC1 work in inserted trigger rotation mode

When the ADCs are in a sync mode other than free mode, they should be configured to free mode before being configured to another sync mode.

The ADC sync scheme is shown in ***Figure 20-15. ADC sync block diagram*** .

Figure 20-15. ADC sync block diagram



20.5.1. Free mode

In this mode, each ADC works independently and does not interfere with each other.

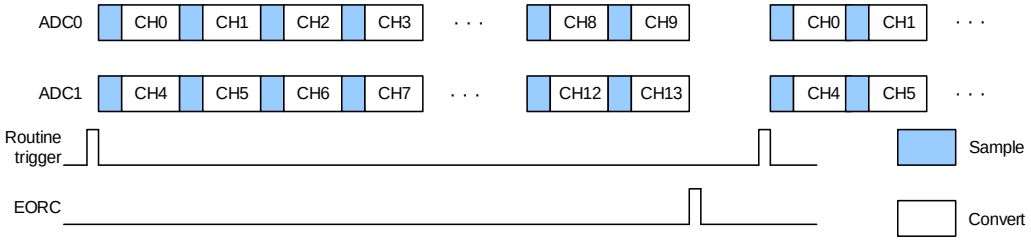
20.5.2. Routine parallel mode

The routine parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0110. This mode converts the routine sequence simultaneously. The source of external trigger comes from the ADC0 routine sequence (configured by the ETMRC[1:0] bits in the ADC_CTL1 register) , and ADC1 routine sequence is configured as software trigger mode.

At the end of conversion event on ADC0 or ADC1, an EORC interrupt is generated (if enabled on one of the two ADC interrupt) when the ADC0/ADC1 routine channels are all converted. The behavior of routine parallel mode is shown in the ***Figure 20-16. Routine parallel mode on 10 channels*** .

A 32-bit DMA is used, which transfers ADC0_RDATA 32-bit register (the ADC0_RDATA 32-bit register containing the ADC1 converted data in the upper half-word and the ADC0 converted data in the lower half-word) to SRAM.

Figure 20-16. Routine parallel mode on 10 channels



Note:

1. If two ADCs use the same sampling channel, it should be ensured that the channel is not used at the same time.
2. Two channels sampled by two ADCs at the same time should be configured with the same sampling time.
3. Make sure to do ADC peripheral reset before ADCON is set to 1.

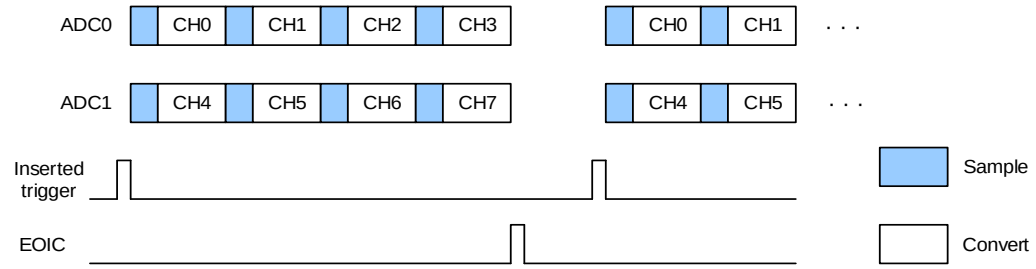
20.5.3. Inserted parallel mode

The inserted parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0101. In the inserted parallel mode, ADC0 and ADC1 convert the inserted sequence parallel at the selected external trigger of ADC0. The trigger is selected by configuring the ETMIC[1:0] bits in the ADC_CTL1 register of ADC0.

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences. The converted data are stored in the ADC_IDATA registers of each ADC interface. The behavior of inserted parallel mode is shown in the ***Figure 20-17. Inserted parallel mode***

on 4 channels .

Figure 20-17. Inserted parallel mode on 4 channels



Note:

1. If two ADCs use the same sampling channel, it should be ensured that the channel is not used at the same time.
2. Two channels sampled by two ADCs at the same time should be configured with the same sampling time.

20.5.4. Routine follow-up fast mode

The routine follow-up fast mode is applicable to sample the same channel of two ADCs. The routine follow-up fast mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0111. The source of external trigger comes from the ADC0 routine channel (selected by the ETMRC[1:0] bits in the ADC_CTL1 register). When the trigger occurs, ADC1 runs immediately and ADC0 runs after 7 ADC clock cycles.

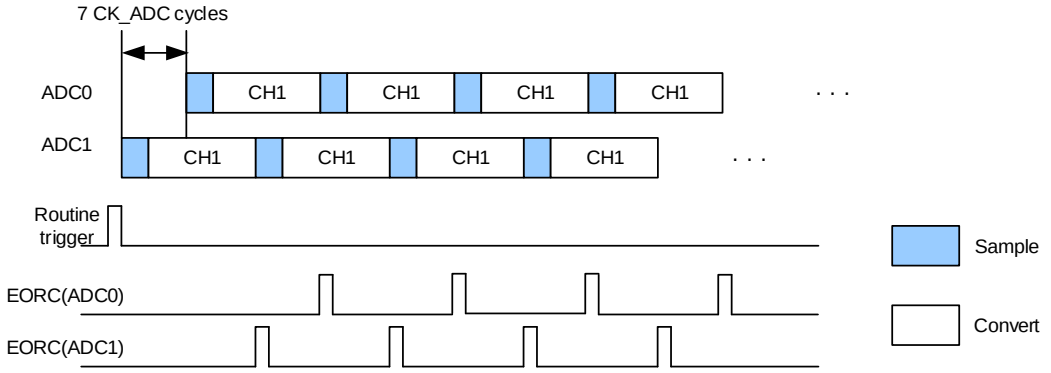
If the continuous mode is enabled for both ADC0 and ADC1, the selected routine channels of two ADCs are continuously converted. The behavior of follow-up fast mode shows in the *Figure 20-18. Routine follow-up fast mode (the CTN bit of ADCs are set)*.

After an EORC interrupt is generated by ADC0 in case of setting the EORCIE bit, we can use a 32-bit DMA, which transfers to SRAM the ADC0_RDATA register containing the ADC1 converted data in the [31: 16] bits field and the ADC0 converted data in the [15: 0] bits field.

Note:

1. The sampling time of the routine channel of the two ADCs should be less than 7 ADC clock cycles
2. Make sure to do ADC peripheral reset before ADCON is set to 1.

Figure 20-18. Routine follow-up fast mode (the CTN bit of ADCs are set)



20.5.5. Routine follow-up slow mode

The routine follow-up slow mode is applicable to sample the same channel of two ADCs. The routine follow-up fast mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b1000. The source of external trigger comes from the ADC0 routine channel (selected by the ETMRC[1:0] bits in the ADC_CTL1 register). When the trigger occurs, ADC1 runs immediately, ADC0 runs after 14 ADC clock cycles, after the second 14 ADC clock cycles the ADC1 runs again.

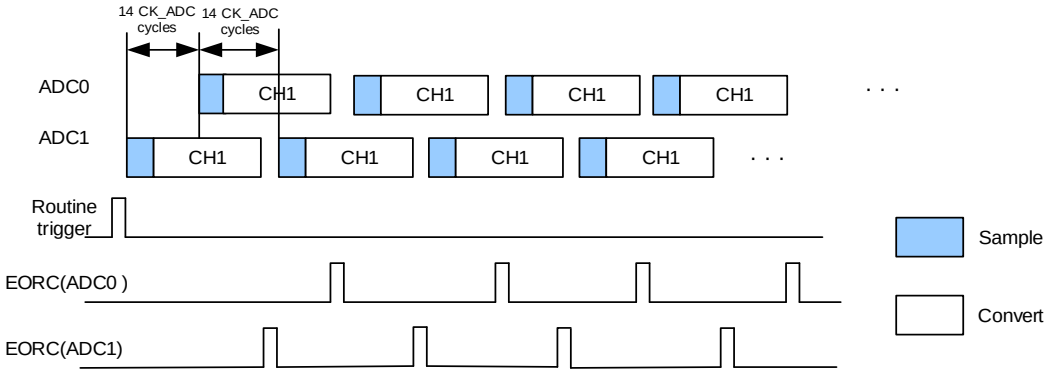
Continuous mode can't be used in this mode, because it continuously converts the routine channel. The behavior of follow-up slow mode shows in the ***Figure 20-19. Routine follow-up slow mode***.

After an EORC interrupt is generated by ADC0 (if EORCIE bit is set), we can use a 32-bit DMA, which transfers to SRAM the ADC0_RDATA register containing the ADC1 converted data in the [31: 16] bits field and the ADC0 converted data in the [15: 0] bits field.

Note:

1. The maximum sampling time allowed is <14 CK_ADC cycles to avoid the overlap between ADC0 and ADC1 sampling phases in the event that they convert the same channel.
2. Make sure to do ADC peripheral reset before ADCON is set to 1.

Figure 20-19. Routine follow-up slow mode

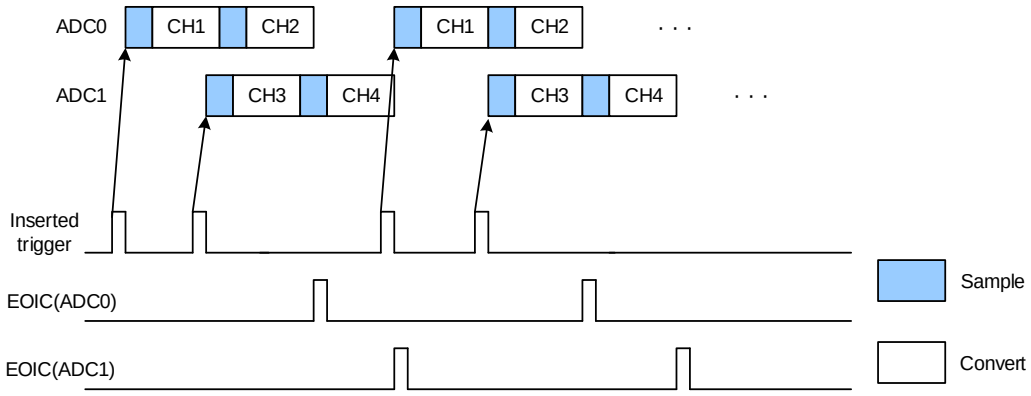


20.5.6. Inserted trigger rotation mode

The inserted trigger rotation mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b1001. In the inserted trigger rotation mode, the inserted sequence of the ADC is triggered in turn by the selected external trigger. When the first trigger valid, all the inserted channels of ADC0 are converted. When the second trigger valid, all the inserted channels of ADC1 are converted. The triggers are selected by configuring the ETMIC[1:0] bits in the ADC_CTL1 register of ADC0.

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences. The behavior of inserted trigger rotation mode is shown in the ***Figure 20-20. Inserted trigger rotation: DISIC=0, IL=1*** when DISIC bit is 0.

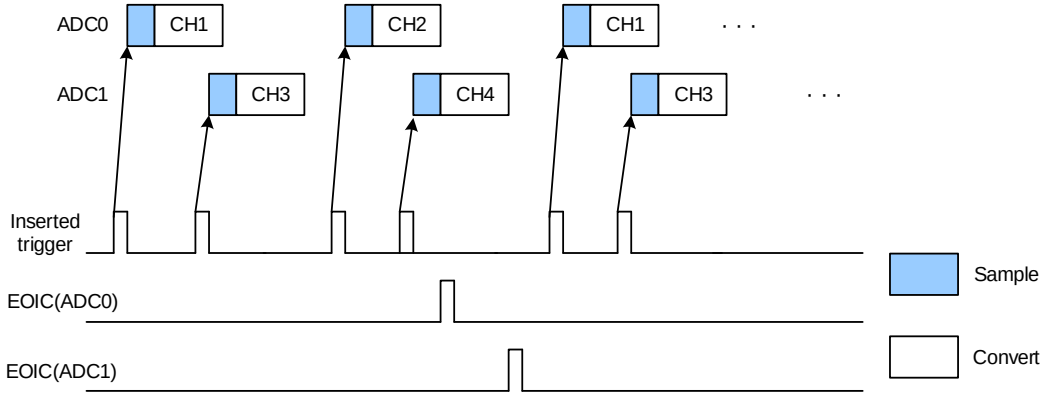
Figure 20-20. Inserted trigger rotation: DISIC=0, IL=1



If the discontinuous operation mode is enabled for both ADC0 and ADC1, when the first trigger valid, the first inserted channel in ADC0 is converted. When the second trigger valid, the first inserted channel in ADC1 is converted. Then the second channel in ADC0, the second channel in ADC1, and so on.

The behavior of trigger rotation mode is shown in the ***Figure 20-21. Inserted trigger rotation: DISIC=1, IL=1*** when DISIC bit is 1.

Figure 20-21. Inserted trigger rotation: DISIC=1, IL=1



Note:

1. Do not convert the same channel on two ADCs at a given time (no overlapping sampling

times for the two ADCs when converting the same channel).

20.5.7. Combined routine parallel & inserted parallel mode

The combined routine parallel & inserted parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0001. In the combined routine parallel & inserted parallel sync mode, it is also possible to interrupt parallel conversion of a routine sequence to insert parallel conversion of an inserted sequence.

EORC interrupts (if enabled on the ADC interfaces) are generated at the end of routine sequences.

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences.

Note:

1. Two channels sampled by two ADCs at the same time should be configured with the same sampling time.
2. Make sure to do ADC peripheral reset before ADCON is set to 1.

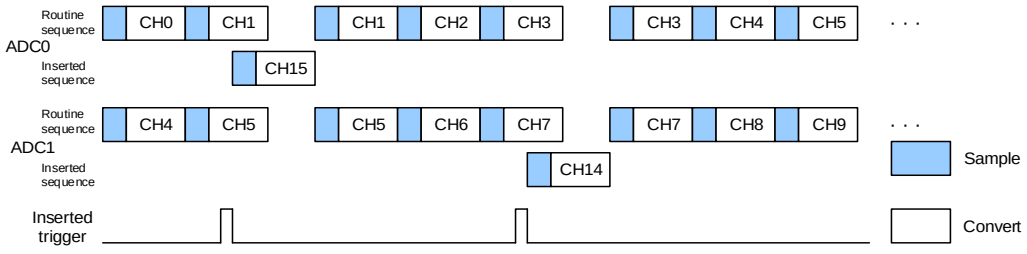
20.5.8. Combined routine parallel & inserted trigger rotation mode

The combined routine parallel & inserted trigger rotation mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0010. In the combined routine parallel & trigger rotation mode, it is possible to interrupt routine sequence parallel conversion by rotation triggering conversion of an inserted sequence. When the inserted sequence trigger event occurs, the inserted conversion is immediately started. When the routine conversions are interrupted, the routine conversion of the ADCs is stopped at the inserted trigger and resumed parallel at the end of the inserted conversion. The behavior of routine parallel conversion interrupted by inserted triggers rotation is shown in the ***Figure 20-22. Routine parallel & inserted trigger rotation mode: SYNCM = 4'b0010***.

EORC interrupts (if enabled on the ADC interfaces) are generated at the end of routine sequences .

EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences.

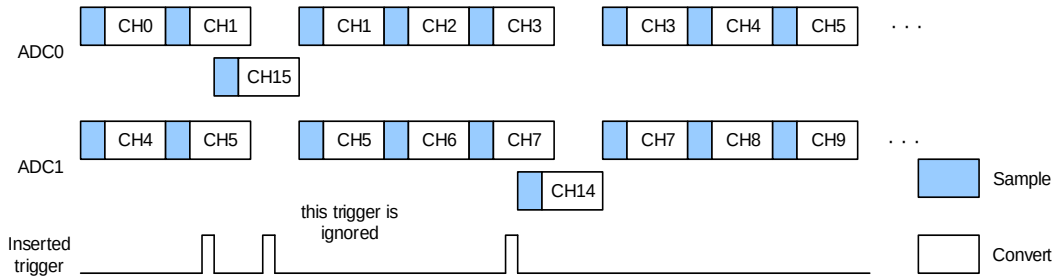
Figure 20-22. Routine parallel & inserted trigger rotation mode: SYNCM = 4'b0010



If another inserted trigger occurs during an inserted conversion, the latter trigger will be

ignored, as shown in ***Figure 20-23. Trigger occurs during inserted conversion: SYNCM = 4'b0010***.

Figure 20-23. Trigger occurs during inserted conversion: SYNCM = 4'b0010



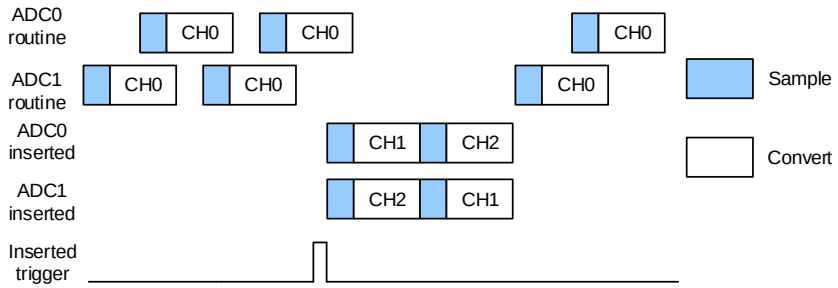
Note:

1. Make sure to do ADC peripheral reset before ADCON is set to 1.

20.5.9. Combined inserted parallel & routine follow-up mode

The combined inserted routine parallel & routine follow-up fast/slow mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0011/4'b0100. It is possible to interrupt a follow-up conversion (both fast and slow) with an inserted event. When the inserted trigger occurs, the follow-up conversion is interrupted and the inserted conversion starts. At the end of the inserted sequence the follow-up conversion is resumed. ***Figure 20-24. Follow-up single channel with inserted sequence CH1, CH2*** shows the behavior of this mode.

Figure 20-24. Follow-up single channel with inserted sequence CH1, CH2



Note:

1. Make sure to do ADC peripheral reset before ADCON is set to 1.

20.6. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for routine and inserted sequences.
- The analog watchdog event.

Separate interrupt enable bits are available for flexibility.

The interrupts of ADC0 and ADC1 are mapped into the same interrupt vector IRQ18. The interrupts of ADC2 are mapped into the interrupt vector IRQ47.

20.7. Register definition

ADC0 base address: 0x4001 2400

ADC1 base address: 0x4001 2800

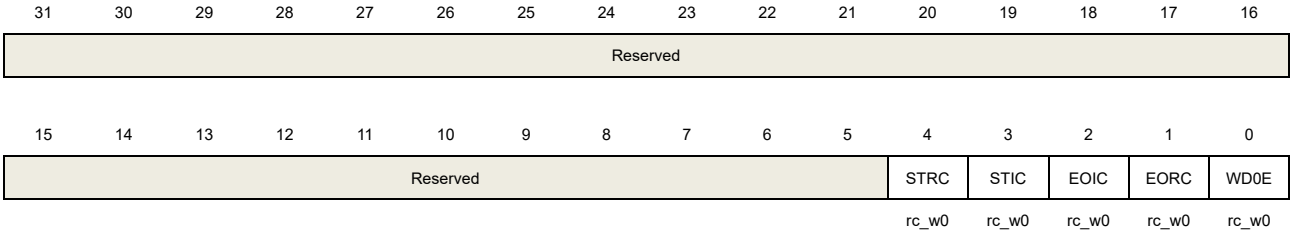
ADC2 base address: 0x4001 3C00

20.7.1. Status register (ADC_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	STRC	Start flag of routine sequence 0: Routine sequence conversion is not started 1: Routine sequence conversion is started Set by hardware when routine sequence conversion starts. Cleared by software writing 0 to it.
3	STIC	Start flag of inserted sequence 0: Inserted sequence conversion is not started 1: Inserted sequence conversion is started Set by hardware when inserted sequence conversion starts. Cleared by software writing 0 to it.
2	EOIC	End flag of inserted sequence conversion 0: No end of inserted sequence conversion 1: End of inserted sequence conversion Set by hardware at the end of all inserted sequence conversion. Cleared by software writing 0 to it.
1	EORC	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 routine sequence conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.

0	WD0E	Analog watchdog 0 event flag 0: Analog watchdog 0 event is not happened 1: Analog watchdog 0 event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD0LT and ADC_WD0HT registers. Cleared by software writing 0 to it.

20.7.2. Control register 0 (ADC_CTL0)

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								RWD0EN	IWD0EN	Reserved		SYNCM[3:0]			
								rw	rw					rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM[2:0]			DISIC	DISRC	ICA	WD0SC	SM	EOICIE	WD0EIE	EORCIE	WD0CHSEL[4:0]				
	rw		rw	rw	rw	rw	rw	rw	rw	rw			rw		

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	RWD0EN	Routine sequence analog watchdog 0 enable 0: Routine sequence analog watchdog 0 disable 1: Routine sequence analog watchdog 0 enable
22	IWD0EN	Inserted sequence analog watchdog 0 enable 0: Inserted sequence analog watchdog 0 disable 1: Inserted sequence analog watchdog 0 enable
21:20	Reserved	Must be kept at reset value.
19:16	SYNCM[3:0]	Sync mode selection These bits use to select the operating mode. 0000: Free mode. 0001: Combined routine parallel + inserted parallel mode 0010: Combined routine parallel + inserted trigger rotation mode 0011: Combined inserted parallel + routine follow-up fast mode 0100: Combined inserted parallel + routine follow-up slow mode 0101: Inserted parallel mode 0110: Routine parallel mode 0111: Routine follow-up fast mode 1000: Routine follow-up slow mode 1001: Inserted trigger rotation mode 1010~1111: Reserved

		Note: 1) These bits are only used in ADC0. 2) Users must disable sync mode before any configuration change.
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	DISIC	Discontinuous mode on inserted sequence 0: Discontinuous operation mode on inserted sequence disable 1: Discontinuous operation mode on inserted sequence enable
11	DISRC	Discontinuous mode on routine sequence 0: Discontinuous operation mode on routine sequence disable 1: Discontinuous operation mode on routine sequence enable
10	ICA	Inserted sequence convert automatically 0: Inserted sequence convert automatically disable 1: Inserted sequence convert automatically enable
9	WD0SC	When in scan mode, analog watchdog 0 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	EOICIE	Interrupt enable for EOIC 0: EOIC interrupt disable 1: EOIC interrupt enable
6	WD0EIE	Interrupt enable for WD0E 0: WD0E interrupt disable 1: WD0E interrupt enable
5	EORCIE	Interrupt enable for EORC 0: EORC interrupt disable 1: EORC interrupt enable
4:0	WD0CHSEL[4:0]	Analog watchdog 0 channel select 00000: ADC channel 0 00001: ADC channel 1 00010: ADC channel 2 00011: ADC channel 3 00100: ADC channel 4 00101: ADC channel 5 00110: ADC channel 6 00111: ADC channel 7 01000: ADC channel 8

01001: ADC channel 9
01010: ADC channel 10
01011: ADC channel 11
01100: ADC channel 12
01101: ADC channel 13
01110: ADC channel 14
01111: ADC channel 15
10000: ADC channel 16
10001: ADC channel 17
Other values are reserved.

Note: ADC0 analog inputs channel16 and channel17 are internally connected to the temperature sensor and V_{REFINT} analog inputs.

20.7.3. 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							TSVEN	INREFEN	SWRCST	SWICST	Reserved	ETMRC[1:0]		Reserved	
							rw	rw	rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ETMIC[1:0]		Reserved	DAL	Reserved.		RDMA	IDMA	Reserved					CTN	ADCON
	rw			rw			rw	rw						rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	TSVEN	Channel 16 (temperature sensor) enable of ADC0. 0: Temperature sensor channel disable 1: Temperature sensor channel enable
23	INREFEN	Channel17 (internal reference voltage V_{REFINT}) enable of ADC0 0: Internal reference voltage channel disable 1: Internal reference voltage channel enable
22	SWRCST	Software start on routine channel Setting 1 on this bit starts a conversion of a sequence of routine channels. It is set by software and cleared by software or by hardware immediately after the conversion starts.
21	SWICST	Software start on inserted channel Setting 1 on this bit starts a conversion of a sequence of inserted channels. It is set by software and cleared by software or by hardware immediately after the conversion starts.

20	Reserved	Must be kept at reset value.
19:18	ETMRC[1:0]	External trigger mode for routine sequence conversion 00, 01, 10: Rising edge of external trigger for routine sequence enable 11: External trigger for routine sequence disable
17:15	Reserved	Must be kept at reset value
14:13	ETMIC[1:0]	External trigger mode for inserted sequence conversion 00, 01, 10: Rising edge of external trigger for inserted sequence enable 11: External trigger for inserted sequence disable
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	RDMA	DMA request enable for routine channel 0: DMA request disable 1: DMA request enable
7	IDMA	DMA request enable for inserted channel 0: DMA request disable 1: DMA request enable
6:2	Reserved	Must be kept at reset value.
1	CTN	Continuous mode 0: Continuous operation mode disable 1: Continuous operation mode enable
0	ADCON	ADC ON The ADC will be waked up when this bit is changed from low to high and take a stabilization time. 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

20.7.4. Sample time register 0 (ADC_SAMPT0)

Address offset: 0x0C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
								rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]		
rw	rw			rw			rw			rw			rw		

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:21	SPT17[2:0]	Refer to SPT10[2:0] description
20:18	SPT16[2:0]	Refer to SPT10[2:0] description
17:15	SPT15[2:0]	Refer to SPT10[2:0] description
14:12	SPT14[2:0]	Refer to SPT10[2:0] description
11:9	SPT13[2:0]	Refer to SPT10[2:0] description
8:6	SPT12[2:0]	Refer to SPT10[2:0] description
5:3	SPT11[2:0]	Refer to SPT10[2:0] description
2:0	SPT10[2:0]	Channel sample time 000: Channel sampling time is 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

20.7.5. Sample time register 1 (ADC_SAMPT1)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

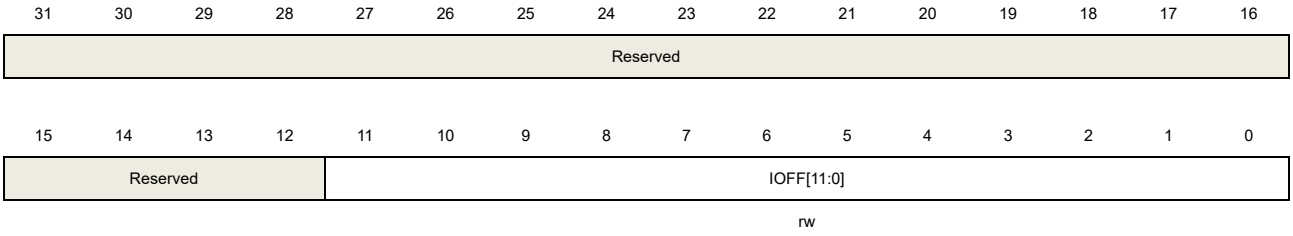
26:24	SPT8[2:0]	Refer to SPT0[2:0] description
23:21	SPT7[2:0]	Refer to SPT0[2:0] description
20:18	SPT6[2:0]	Refer to SPT0[2:0] description
17:15	SPT5[2:0]	Refer to SPT0[2:0] description
14:12	SPT4[2:0]	Refer to SPT0[2:0] description
11:9	SPT3[2:0]	Refer to SPT0[2:0] description
8:6	SPT2[2:0]	Refer to SPT0[2:0] description
5:3	SPT1[2:0]	Refer to SPT0[2:0] description
2:0	SPT0[2:0]	Channel sample 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

20.7.6. Inserted channel data offset register x (ADC_IOFFx) (x = 0..3)

Address offset: 0x14 + 0x04 * x

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	IOFF[11:0]	Data offset for inserted channel x These bits will be subtracted from the raw converted data when converting inserted channels. The conversion result can be read from in the ADC_IDATA register.

20.7.7. Watchdog 0 high threshold register (ADC_WD0HT)

Address offset: 0x24

Reset value: 0x000F 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
Reserved												WD0HT[19:16]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD0HT[15:0]															
rw															

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD0HT[19:0]	High threshold for analog watchdog 0 These bits define the high threshold for the analog watchdog 0.

20.7.8. Watchdog 0 low threshold register (ADC_WD0LT)

Address offset: 0x28

Reset value: 0x000F 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
Reserved												WD0LT[19:16]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD0LT[15:0]															
rw															

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD0LT[19:0]	Low threshold for analog watchdog 0 These bits define the low threshold for the analog watchdog 0.

20.7.9. Routine sequence register 0 (ADC_RSQ0)

Address offset: 0x2C

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RL[3:0]				RSQ15[4:1]			

								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]	RSQ14[4:0]					RSQ13[4:0]					RSQ12[4:0]				
rw	rw					rw					rw				

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	RL[3:0]	Routine sequence length. The total number of conversion in routine sequence equals to RL[3:0]+1.
19:15	RSQ15[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ14[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ13[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ12[4:0]	Refer to RSQ0[4:0] description

20.7.10. Routine sequence register 1 (ADC_RSQ1)

Address offset: 0x30

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		RSQ11[4:0]					RSQ10[4:0]					RSQ9[4:1]			
		rw					rw					rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ9[0]	RSQ8[4:0]					RSQ7[4:0]					RSQ6[4:0]				
rw	rw					rw					rw				

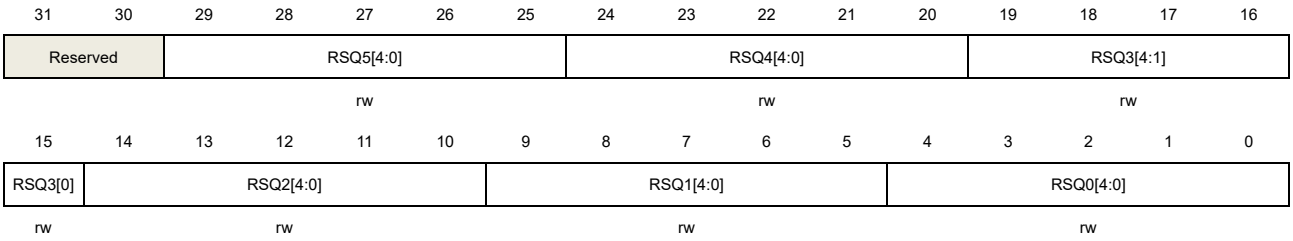
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ11[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ10[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ9[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ8[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ7[4:0]	Refer to RSQ0[4:0] description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

20.7.11. Routine sequence register 2 (ADC_RSQ2)

Address offset: 0x34

Reset value: 0x0000 0000

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



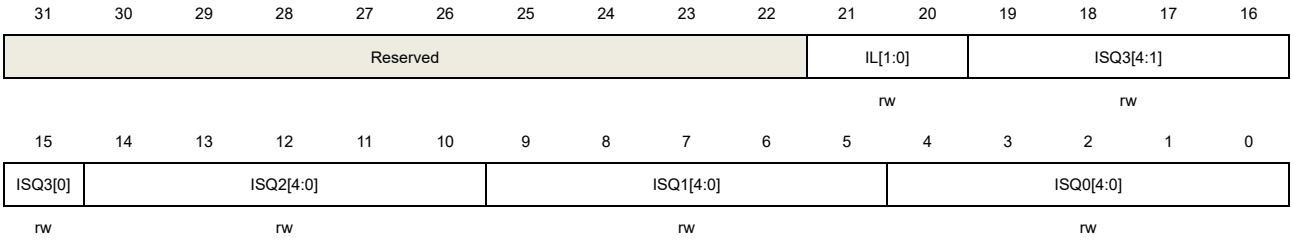
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ5[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ4[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ3[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ2[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ1[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ0[4:0]	The channel number (0..17) is written to these bits to select a channel as the nth conversion in the routine sequence.

20.7.12. Inserted sequence register (ADC_ISQ)

Address offset: 0x38

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:20	IL[1:0]	Inserted sequence length The total number of conversion in inserted sequence equals to IL[1:0] + 1.
19:15	ISQ3[4:0]	Refer to ISQ0[4:0] description
14:10	ISQ2[4:0]	Refer to ISQ0[4:0] description

9:5	ISQ1[4:0]	Refer to ISQ0[4:0] description
4:0	ISQ0[4:0]	The channel number (0..17) is written to these bits to select a channel at the nth conversion in the inserted sequence. Unlike the routine conversion sequence, the inserted channels are converted starting from (4 - IL[1:0] - 1), if IL[1:0] length is less than 4. IL: Inserted channel order 3: ISQ0 >> ISQ1 >> ISQ2 >> ISQ3 2: ISQ1 >> ISQ2 >> ISQ3 1: ISQ2 >> ISQ3 0: ISQ3

20.7.13. Latch data register x (ADC_LDATAx) (x= 0..3)

Address offset: 0x3C + 0x04 * x

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	LDATAn[15:0]	Inserted or routine number n conversion data. These bits contain the number n conversion result, which is read only.

20.7.14. Routine data register (ADC_RDATA)

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
ADC1RDATA[15:0]															
r															

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

Bits	Fields	Descriptions
31:16	ADC1RDATA[15:0]	ADC1 routine channel data

		In ADC0: In sync mode, these bits contain the routine data of ADC1. In ADC1: These bits are reserved.
15:0	RDATA[15:0]	Routine channel data These bits contain routine channel conversion value, which is read only.

20.7.15. Inserted data register (ADC_IDATA)

Address offset: 0x50

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	IDATA[15:0]	Inserted channel data These bits contain the inserted channel conversion value, which is read only.

20.7.16. Latch data control register(ADC_LDCTL)

Address offset: 0x54

Reset value: 0x0001 0203

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEQSEL0	Reserved			COVSEL0[3:0]				SEQSEL1	Reserved			COVSEL1[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQSEL2	Reserved			COVSEL2[3:0]				SEQSEL3	Reserved			COVSEL3[3:0]			
rw				rw				rw				rw			

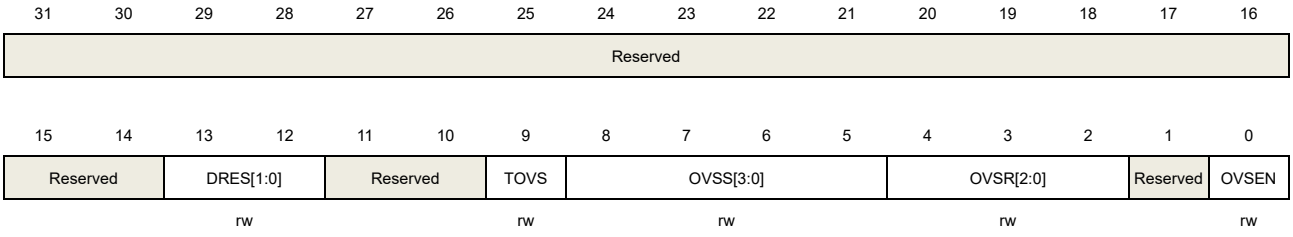
Bits	Fields	Descriptions
31	SEQSEL0	The ADC_LDATA0 register sequence source selection 0: Select inserted sequence 1: Select routine sequence **Note:** The software allows this bit to be written only when ADCON=0 (this ensures

		that no conversion is progress).
30:28	Reserved	Must be kept at reset value.
27:24	COVSEL0[3:0]	The ADC_LDATA0 register conversion source selection When ADC_LDATA0 register is used for sequence conversion, the nth conversion data of the inserted sequence (SEQSEL0 = 0) or the routine sequence (SEQSEL0 = 1) can be selected to be stored in ADC_LDATA0 register. 0000: The 0th conversion of the sequence. 0001: The 1st conversion of the sequence. 0010: The 2nd conversion of the sequence. … 1111: the 15th conversion of the sequence. **Note:** When used in inserted sequence (SEQSEL0 = 0), it can only be 0000,0001,0010,0011, and the default value is 0000. The software allows this bit to be written only when ADCON=0 (this ensures that no conversion is progress).
23	SEQSEL1	Refer to SEQSEL0 description.
22:20	Reserved	Must be kept at reset value.
19:16	COVSEL1[3:0]	Refer to COVSEL0[3:0] description.
15	SEQSEL2	Refer to SEQSEL0 description.
14:12	Reserved	Must be kept at reset value.
11:8	COVSEL2[3:0]	Refer to COVSEL0[3:0] description.
7	SEQSEL3	Refer to SEQSEL0 description.
6:4	Reserved	Must be kept at reset value.
3:0	COVSEL3[3:0]	Refer to COVSEL0[3:0] description.

20.7.17. Oversample control register (ADC_OVSAMPCTL)

Address offset: 0x80
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions

31:14	Reserved	Must be kept at reset value.
13:12	DRES[1:0]	ADC resolution 00: 12bit 01: 10bit 10: 8bit 11: 6bit
11:10	Reserved	Must be kept at reset value.
9	TOVS	Triggered oversampling This bit is set and cleared by software. 0: All oversampled conversions for a channel are done consecutively after a trigger 1: Each conversion needs a trigger for a oversampled channel and the number of triggers is determined by the oversampling ratio(OVSR[2:0]). **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
8:5	OVSS[3:0]	Oversampling shift This bit is set and cleared by software. 0000: No shift 0001: Shift 1-bit 0010: Shift 2-bit 0011: Shift 3-bit 0100: Shift 4-bit 0101: Shift 5-bit 0110: Shift 6-bit 0111: Shift 7-bit 1000: Shift 8-bit Other values are reserved. **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
4:2	OVSR[2:0]	Oversampling ratio This bit filed defines the number of oversampling ratio. 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
1	Reserved	Must be kept at reset value.

0	OVSEN	Oversampler enable This bit is set and cleared by software. 0: Oversampler disabled 1: Oversampler enabled **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).

21. Digital-to-analog converter (DAC)

21.1. 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.

21.2. 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.
- Conversion update synchronously.
- Conversion triggered by external triggers.
- Configurable internal buffer.
- Extern voltage reference, V_{REFP}.
- Noise wave generation (LFSR noise mode and triangle noise mode).

Figure 21-1. DAC block diagram and *Table 21-1. DAC I/O description* show the block

diagram of DAC and the pin description of DAC, respectively.

Figure 21-1. DAC block diagram

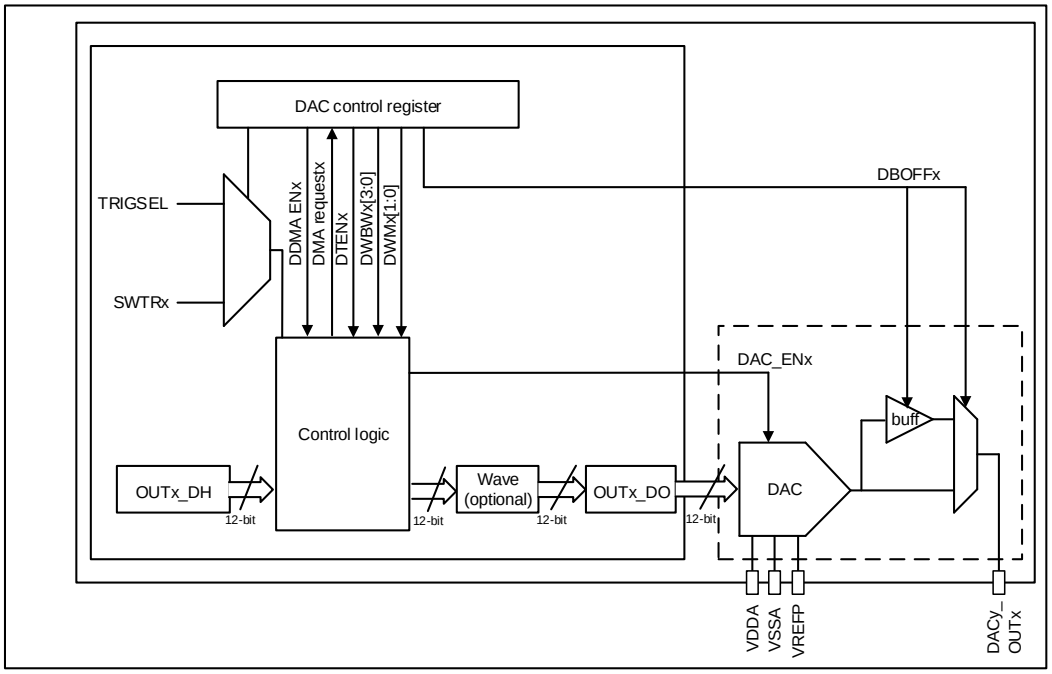


Table 21-1. DAC I/O description

Name	Description	Signal type
V_{DDA}	Analog power supply	Input, analog supply
V_{SSA}	Ground for analog power supply	Input, analog supply ground
V_{REFP}	Positive reference voltage of DAC	Input, analog positive reference
DACy_OUTx	DAC analog output	Analog output signal

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

Table 21-2. DAC triggers and outputs summary

	DAC0
Channel	**Channel0**
DAC outputs connected to I / Os	PA4 / PA5
DAC output buffer	•
DAC software trigger	•
DAC trigger signals from TRIGSEL	•

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

21.3. Function overview

21.3.1. 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.

21.3.2. 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.

21.3.3. DAC data configuration

The 12-bit DAC holding data (OUTx_DH) can be configured by writing any one of the DAC_OUTx_R12DH, DAC_OUTx_L12DH and DAC_OUTx_R8DH registers. When the data is loaded by DAC_OUTx_R8DH register, only the MSB 8 bits are configurable, the LSB 4 bits are forced to 4'b0000.

21.3.4. 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 21-3. Triggers of DAC***.

Table 21-3. Triggers of DAC

DTSELx[2:0]	Trigger Source	Trigger Type
2b'00	TRIGSEL	Hardware trigger
2b'01 /2b'10	Reserved	Reserved
2b'11	SWTR	Software trigger

The external trigger is generated from the TRIGSEL, while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

Note：The DAC module can only be triggered after the DAC module is enabled (DEN0=1).

21.3.5. DAC conversion

If the external trigger is enabled by setting the DTENx bit in DAC_CTL0 register, the DAC holding data is transferred to the DAC output data (DAC_OUTx_DO) register when the selected trigger event happened. When the external trigger is disabled, the transfer is performed automatically.

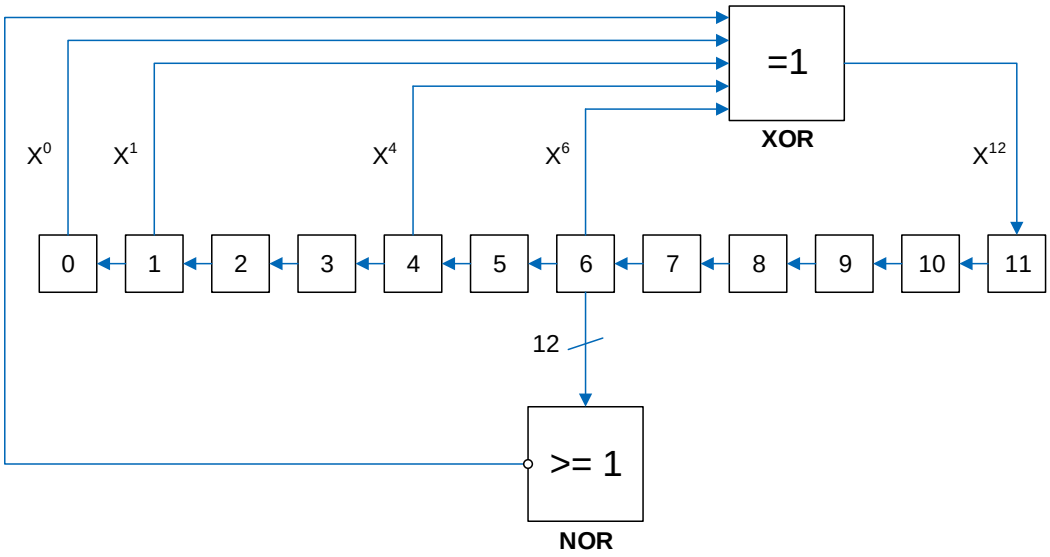
When the DAC holding data (OUTx_DH) is loaded into the DAC_OUTx_DO register, after the time $t_{SETTLING}$ which is determined by the analog output load and the power supply voltage, the analog output is valid.

21.3.6. DAC noise wave

There are two methods of adding noise wave to the DAC output data: LFSR noise wave mode and Triangle wave mode. The noise wave mode can be selected by the DWMx bits in the DAC_CTL0 register. The amplitude of the noise can be configured by the DAC noise wave bit width (DWBWx) bits in the DAC_CTL0 register.

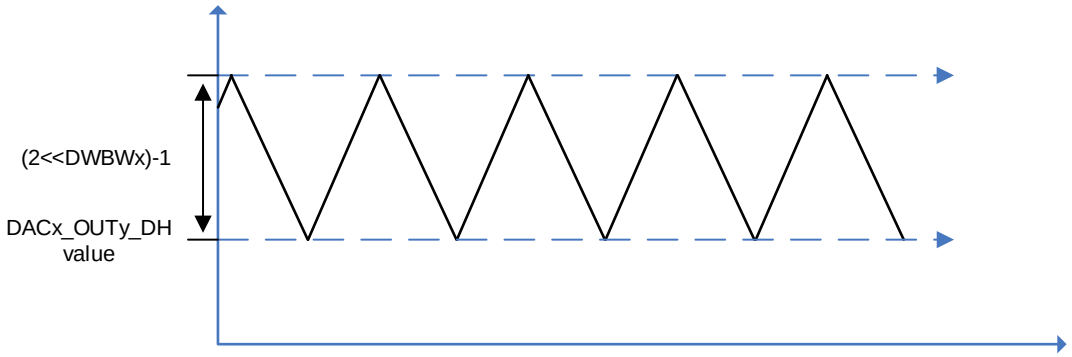
LFSR noise wave mode: there is a Linear Feedback Shift Register (LFSR) in the DAC control logic, it controls the LFSR noise signal which is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. When the configured DAC noise wave bit width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

Figure 21-2. DAC LFSR algorithm



Triangle noise mode: a triangle signal is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. The minimum value of the triangle signal is 0, while the maximum value of the triangle signal is (2 << DWBWx) - 1.

Figure 21-3. DAC triangle noise wave



21.3.7. DAC output voltage

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

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (21\text{-}1)$$

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

21.3.8. DMA request

When the external trigger is enabled, the DMA request is enabled by setting the DDMAENx bit of the DAC_CTL0 register. A DMA request will be generated when an external hardware trigger (not a software trigger) occurs.

21.4. Register definition

DAC0 base address: 0x4000 7400

21.4.1. DACx control register 0 (DAC_CTL0)

Address offset: 0x00
Reset value: 0x0000 8000

This register 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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTSEL0	DCSEL0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		Reserved	DTSEL0[1:0]		DTEN0	DBOFF0	DEN0
rw	rw	rw	rw	rw				rw			rw		rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	OUTSEL0	DACx_OUT0 DAC output pin select 0: DACx_OUT0 connected to PA5 1: DACx_OUT0 connected to PA4 (default is 1)
14	DCSEL0	When DACx_OUT0 is connected to CMP, connection selection of DACx_OUT0 with GPIO: 0: DACx_OUT0 is connected to GPIO 1: DACx_OUT0 is not connected to GPIO
13	DDUDRIE0	DACx_OUT0 DMA underrun interrupt enable 0: DACx_OUT0 DMA underrun interrupt disabled 1: DACx_OUT0 DMA underrun interrupt enabled
12	DDMAEN0	DACx_OUT0 DMA enable 0: DACx_OUT0 DMA mode disabled 1: DACx_OUT0 DMA mode enabled
11:8	DWBW0[3:0]	DACx_OUT0 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT0. These bits indicate that unmask LFSR bit [n-1, 0] in LFSR noise mode or the amplitude of the triangle is ((2<<(n-1))-1) in triangle noise mode, where n is the bit width of wave. 0000: The bit width of the wave signal is 1 0001: The bit width of the wave signal is 2 0010: The bit width of the wave signal is 3 0011: The bit width of the wave signal is 4 0100: The bit width of the wave signal is 5

		0101: The bit width of the wave signal is 6 0110: The bit width of the wave signal is 7 0111: The bit width of the wave signal is 8 1000: The bit width of the wave signal is 9 1001: The bit width of the wave signal is 10 1010: The bit width of the wave signal is 11 ≥1011: The bit width of the wave signal is 12 Note：This bit field cannot be written when the DAC is enabled (DEN0=1).
7:6	DWM0[1:0]	DACx_OUT0 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT0 when external trigger of DACx_OUT0 is enabled (DTEN0=1). 00: Wave disabled 01: LFSR noise mode 1x: Triangle noise mode
5	Reserved	Must be kept at reset value
4:3	DTSEL0[1:0]	DACx_OUT0 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 00: EXTRIG (external trigger from TRIGSEL) 11: Software trigger All other values: reserved.
2	DTEN0	DACx_OUT0 trigger enable 0: DACx_OUT0 trigger disabled 1: DACx_OUT0 trigger enabled
1	DBOFF0	DACx_OUT0 output buffer turn off 0: DACx_OUT0 output buffer turns on to reduce the output impedance and improve the driving capability 1: DACx_OUT0 output buffer turns off
0	DEN0	DACx_OUT0 enable 0: DACx_OUT0 disabled 1: DACx_OUT0 enabled

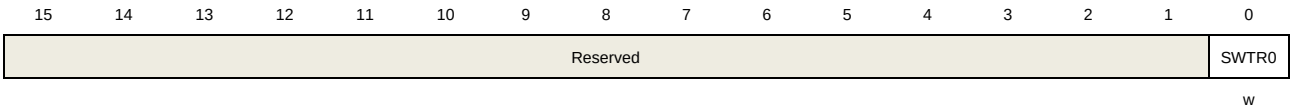
21.4.2. DACx software trigger register (DAC_SWT)

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															



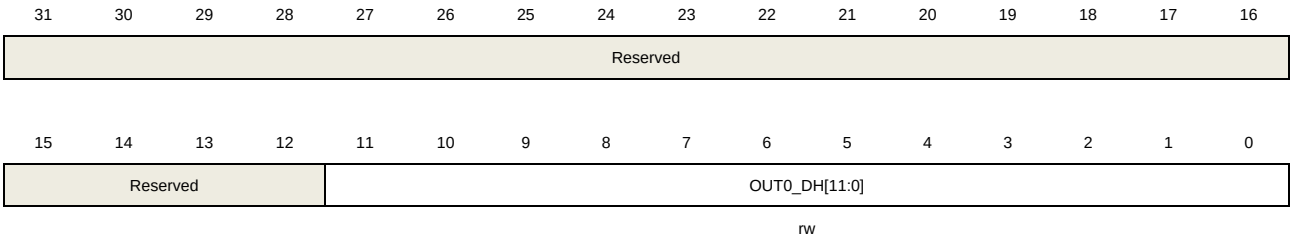
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SWTR0	DACx_OUT0 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled

21.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH)

Address offset: 0x08

Reset value: 0x0000 0000

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



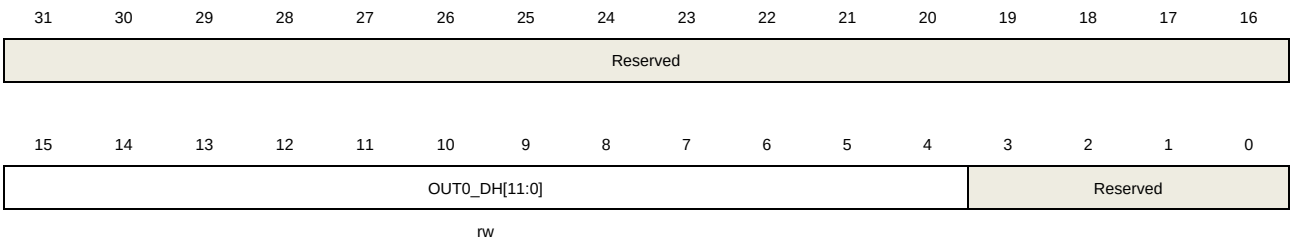
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT0.

21.4.4. 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).



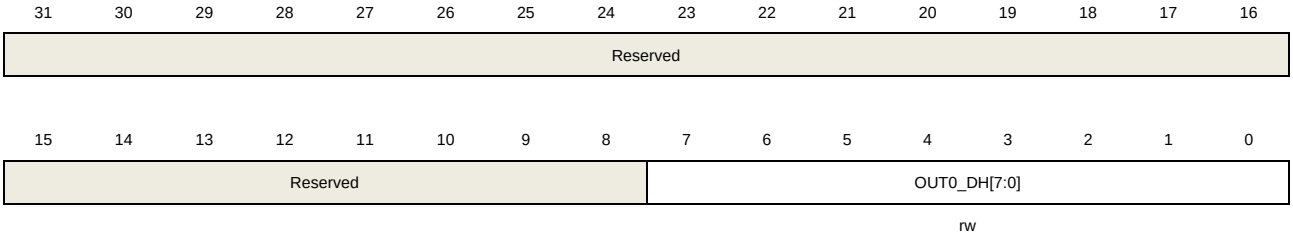
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

21.4.5. 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).



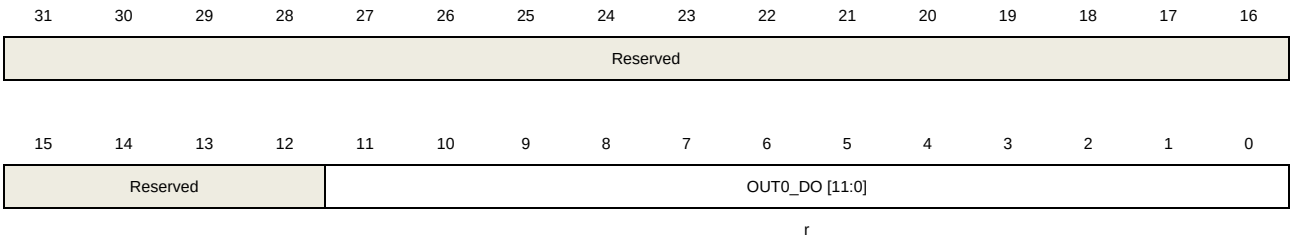
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

21.4.6. 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

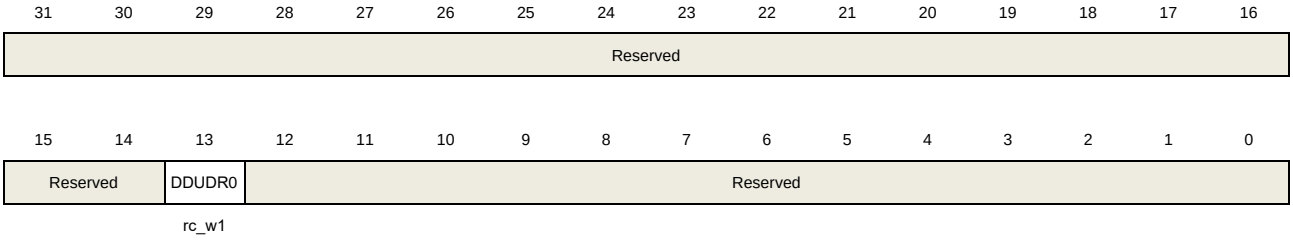
DACx_OUT0.

21.4.7. DACx_OUT0 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. 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is higher than the DMA transfer).
12:0	Reserved	Must be kept at reset value.

22. Comparator (CMP)

22.1. Overview

The general purpose CMP can work either standalone (all terminal are available on I/Os) or together with the timers.

It can be used to wake up the MCU from low-power mode by an analog signal, provide a trigger source when an analog signal is in a certain condition, achieve some current control by working together with a PWM output of a timer and the DAC. It blanking function can be used for false overcurrent detection in motor control applications.

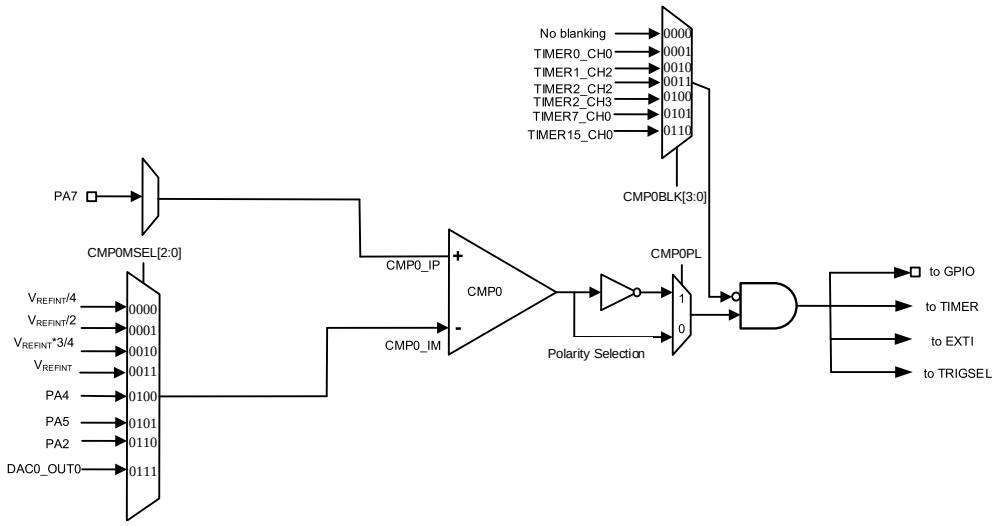
22.2. Characteristic

- Rail-to-rail comparators.
- Configurable hysteresis.
- Each comparator has configurable analog input source.
  - DAC output.
  - Multiplexed I/O pins.
  - The whole or sub-multiple values of internal reference voltage.
- Outputs with blanking source.
- Outputs to I/O.
- Outputs to timers for triggering.
- Outputs to EXTI.
- Outputs to NVIC.
- Outputs to TRIGSEL.

22.3. Function overview

The block diagrams of CMP are shown in ***Figure 22-1. CMP block diagram***.

Figure 22-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

22.3.1. CMP clock and reset

The CMP clock is synchronous with the PCLK by the clock controller. The CMP has private reset and clock enable bits.

22.3.2. CMP I/O configuration

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

The CMP output can be redirected internally and externally simultaneously.

Refer to pin definitions in datasheet, and the CMP output can be connected to the corresponding I/O port via the alternate function of the GPIO.

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

- CMP output to the TIMER break.

In order to work even in deep-sleep mode, either polarity selection logic or output redirection to the port are independent of PCLK.

Table 22-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 22-1. CMP inputs and outputs summary

	CMP0
CMP non inverting inputs connected to I/Os	PA7
CMP inverting inputs connected to I/Os	PA4 PA5 PA2

	CMP0
CMP inverting inputs connected to internal signals	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0
CMP outputs connected to I/Os	PA0 PA1 PA6 PB0 PB1
CMP outputs connected to EXTI	•
CMP outputs connected to TRIGSEL	•
CMP outputs connected to NVIC	•
CMP outputs connected to TIMER break	•

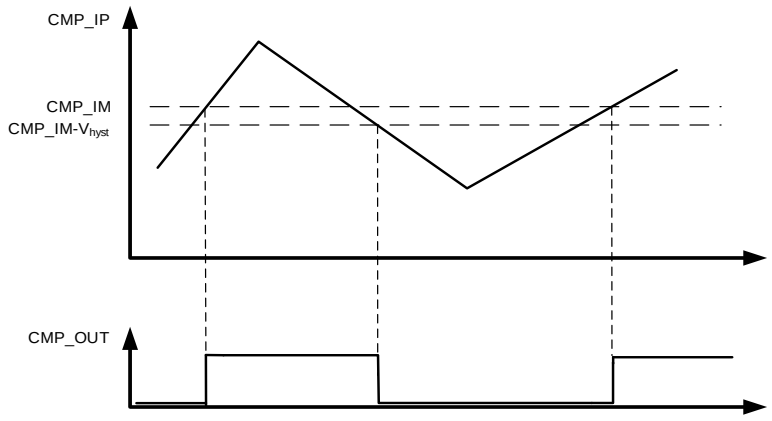
Note:

1. Refer to the TIMER module for details on CMP outputs connected to TIMER break.

22.3.3. 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 CMP0_CS register. This function could be shut down if it is unnecessary.

Figure 22-2. CMP hysteresis



22.3.4. CMP register write protection

The CMP control and status register (CMP0_CS) can be protected from writing by setting CMP0LK bit to 1. The CMP0_CS register, including the CMP0LK bit will be read-only, and

can only be reset by the MCU reset.

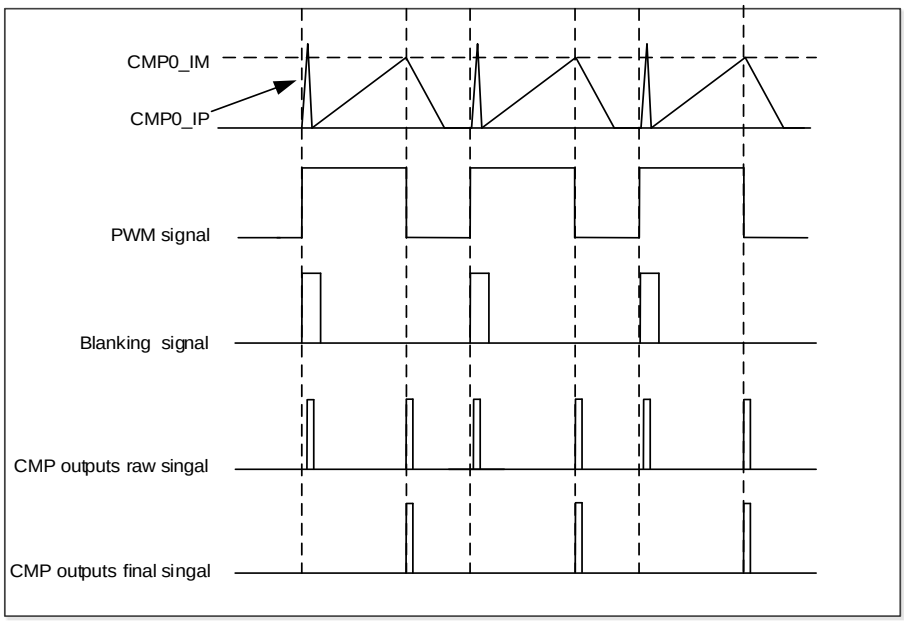
This write protection function is useful in some applications, such as thermal protection and over-current protection.

22.3.5. CMP output blanking

CMP output blanking function can be used to avoid interference of short pulses in the input signal to CMP output signal. If the CMP0BLK[3:0] bits in the CMP0_CS register are setting to an available value, the CMP output final signal is obtained by ANDing the complementary signal of the selected blanking signal with the raw output of the comparator. The blanking function can be used for false overcurrent detection in motor control applications.

Figure 22-3. The CMP outputs signal blanking shows the comparator output blank function.

Figure 22-3. The CMP outputs signal blanking



22.3.6. CMP digital filter

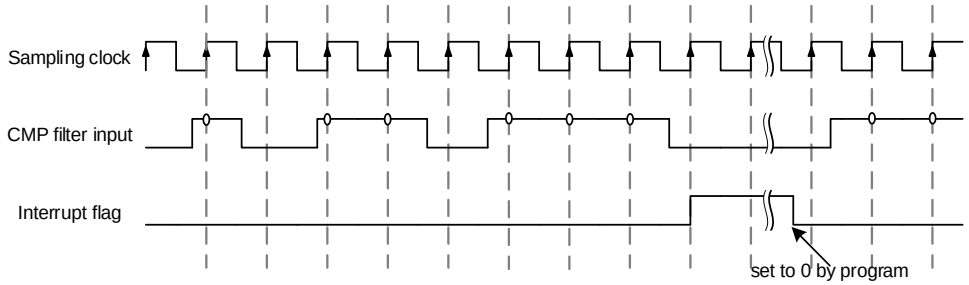
The sampling clock of digital filter can be selected by the CMP0DFSCDIV[1:0] bits and the number of matches can be selected by the CMP0DFSNUM bit of CMP0_CS register. The function of digital filter function are as follows:

- One of three sampling periods can be selected.
- The filter function can also be disabled.
- A noise-filtered signal can be used to generate the interrupt request output, TIMER internal trigger source output, and comparison result output.

The comparator output is sampled every sampling clock, and if the same value is sampled two or three times, that value is determined as the noise filter output at the next sampling clock. Details of digital noise filter are shown in *Figure 22-4. CMP noise filter and interrupt*

operation.

Figure 22-4. CMP noise filter and interrupt operation



The above operation example is to enable digital noise filtering, and applies when the CMP0DFSCDIV[1:0] bits are 2'b01,2'b10 and 2'b11 and CMP0DFSNUM bit is 0. The noise canceling width can be changed by setting the value of the CMP0DFSCDIV[1:0] bits to select the sampling clock.

22.3.7. CMP voltage scaler function

The voltage scaler function can provide 1 / 4, 1 / 2, 3 / 4 reference voltage to CMP input for selection. It is controlled by CMP0SEN and CMP0BEN bits in CMP0_CS register, which enable or disable the analog amplifier and the resistor bridge, respectively.

22.3.8. CMP interrupt

The CMP output is connected to the EXTI and the EXTI line is exclusive to each CMP. With this function, CMP can generate either interrupts or events which can used to exit from low-power modes or stop modes.

The CMP also can generate an interrupt to NVIC. It is a sequential logic signal, so the PCLK is needed.

22.4. Register definition

CMP base address: 0x4000 7800

22.4.1. Status register (CMP_STAT)

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															CMP0IF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															CMP0O
															r

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CMP0IF	CMP0 interrupt flag 0: No CMP0 output interrupt 1: CMP0 output interrupt Set by hardware when the CMP0 output is set. Cleared by software writing 1 to CMP0IC bit in the CMP_IFC register.
15:1	Reserved	Must be kept at reset value.
0	CMP0O	CMP0 output state This 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

22.4.2. Interrupt flag clear register (CMP_IFC)

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															CMP0IC
															w

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CMP0IC	CMP0 interrupt flag clear 0: Not clear CMP0 interrupt flag 1: Clear CMP0 interrupt flag
15:0	Reserved	Must be kept at reset value.

22.4.3. CMP0 control/status register (CMP0_CS)

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
CMP0LK	CMP0DFSNUM	CMP0DFSCDIV[1:0]		CMP0BLK[3:0]				Reserved				CMP0MSEL[3:0]			
rwo	rw	rw		rw								rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CMP0HST[1:0]		Reserved	CMP0INTEN	Reserved		CMP0PL	CMP0SEN	CMP0BEN	CMP0EN
						rw			rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP0LK	CMP0 lock This bit allows to have all control bits of CMP0 as read-only. This bit is write once. It can only be cleared by a system reset once it is set by software. 0: CMP0_CS bits are read-write 1: CMP0_CS bits are read-only
30	CMP0DFSNUM	CMP0 digital filter sampling number 0: Output changes when the same value is sampled 2 times. 1: Output changes when the same value is sampled 3 times
29:28	CMP0DFSCDIV[1:0]	CMP0 digital filter sampling clock division These bits are used to select the sampling frequency of digital filter. 00: Digital filter is not used 01: Sampling frequency is f_{PCLK} / 8. 10: Sampling frequency is f_{PCLK} / 16. 11: Sampling frequency is f_{PCLK} / 32.
27:24	CMP0BLK[3:0]	CMP0 output blanking source This bit is used to select which timer output controls the CMP0 output blanking. 0000: No blanking 0001: Select TIMER0_CH0 as blanking source 0010: Select TIMER1_CH2 as blanking source

		0011: Select TIMER2_CH2 as blanking source 0100: Select TIMER2_CH3 as blanking source 0101: Select TIMER7_CH0 as blanking source 0110: Select TIMER15_CH0 as blanking source All other values: reserved
23:20	Reserved	Must be kept at reset value.
19:16	CMP0MSEL[3:0]	CMP0_IM internal input selection These bits are used to select the source connected to the CMP0_IM input of the CMP0. 0000: V_{REFINT} / 4 0001: V_{REFINT} / 2 0010: V_{REFINT}*3 / 4 0011: V_{REFINT} 0100: PA4 0101: PA5 0110: PA2 0111: DAC0_OUT0 All other values: reserved
15:10	Reserved	Must be kept at reset value.
9:8	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
7	Reserved	Must be kept at reset value.
6	CMP0INTEN	CMP0 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	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
2	CMP0SEN	Voltage scaler enable bit When CMP0 is enabled, the bit must be set. 0: Disable V_{REFINT} voltage scaler in case that CMP0SEN bit of CMP0_CS is also reset 1: Enable voltage scaler

1	CMP0BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge disable in case that CMP0BEN bit of CMP0_CS is also reset 1: Enable scaler resistor bridge
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

23. Universal synchronous/asynchronous receiver /transmitter (USART)

23.1. Overview

The Universal Synchronous/Asynchronous Receiver/Transmitter (USART) provides a flexible serial data exchange interface. Data frames can be transferred in full duplex or half duplex mode, synchronously or asynchronously through this interface. A programmable baud rate generator divides the UCLK (PCLK1 or PCLK2) to produce a dedicated baud rate 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 independently and flexibly.

The USART supports DMA function for high-speed data communication, except UART4.

23.2. Characteristics

- NRZ standard format.
- Asynchronous, full duplex communication.
- Half duplex single wire communications.
- Programmable baud-rate generator:
  - Divided from the peripheral clocks, PCLK2 for USART0, PCLK1 for USART1/2 and UART3/4.
  - Oversampling by 8 or 16.
  - Maximum speed up to 35 MBits/s (PCLK2 280M and oversampling by 8).
- Fully programmable serial interface characteristics:
  - Even, odd or no-parity bit generation/detection.
  - A data word length can be 8 or 9 bits.
  - 0.5, 1, 1.5 or 2 stop bit generation.
- Transmitter and receiver can be enabled separately.
- Hardware flow control protocol (CTS/RTS).
- DMA request for data buffer access.
- 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 frame or address match detection.
- Various status flags:
  - Flags for transfer detection: receive buffer not empty (RBNE), transmit buffer empty (TBE), transfer complete (TC), and busy (BSY).
  - 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).
  - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF).
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set.

While USART0/1/2 is fully implemented, UART3/4 is only partially implemented with the following features not supported.

- Smartcard mode
- Synchronous mode
- Hardware flow control protocol (CTS/RTS)
- Configurable data polarity
- Receive timeout

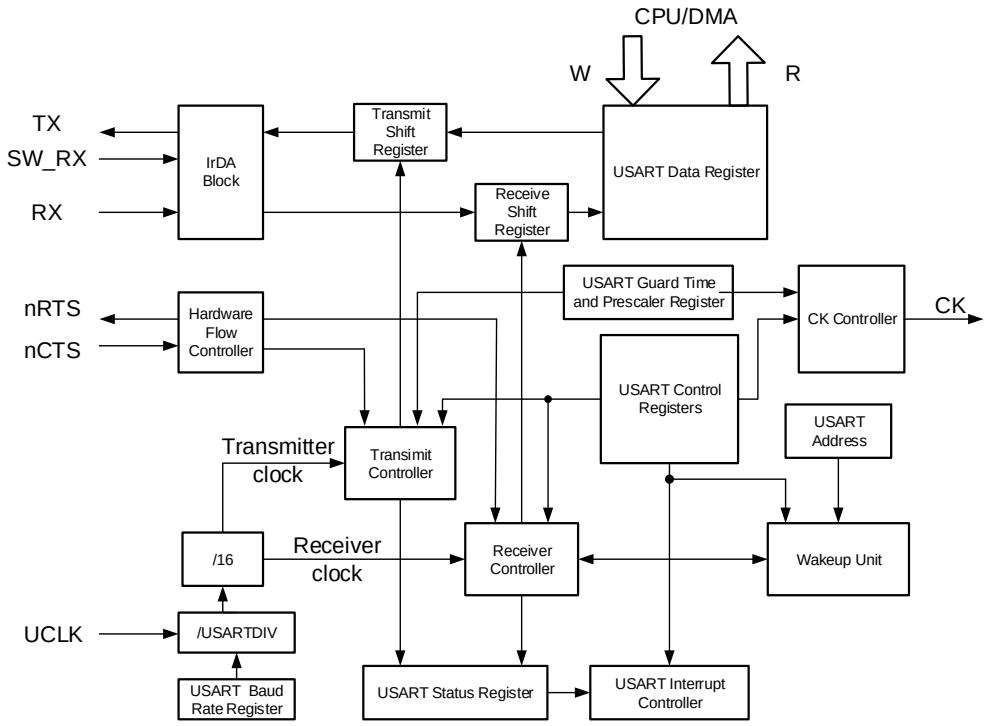
23.3. Function overview

The interface is externally connected to another device by the main pins listed in ***Table 23-1. Description of USART important pins***.

Table 23-1. Description of USART important pins

Pin	Type	Description
RX	Input	Receive data
TX	Output I/O (single-wire/Smartcard mode)	Transmit data. High level when enabled but nothing to be transmitted
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

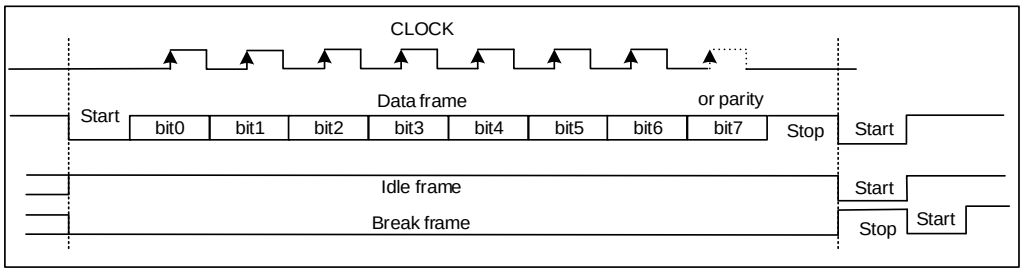
Figure 23-1. USART module block diagram



23.3.1. 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 23-2. 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 23-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	default value
01	0.5	Smartcard mode for receiving
10	2	normal USART and single-wire modes

STB[1:0]	stop bit length (bit)	usage description
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK and the configuration of the baud rate generator.

23.3.2. Baud rate generation

The baud-rate divider is a 18-bit number which consists of a 14-bit integer and a 4-bit fractional part. The number formed by these two values is used by the baud rate generator to determine the bit period. Having a fractional baud-rate divider allows the USART to generate all the standard baud rates.

The baud-rate divider (USARTDIV) has the following relationship with the peripheral clock:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (23\text{-}1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (23\text{-}2)$$

For example, when oversampled by 16:

1. Get USARTDIV by caculating the value of USART_BUAD:
   If USART_BUAD=0x21D, then INTDIV=33 (0x21), FRADIV=13 (0xD).
   USARTDIV=33+13/16=33.81.
2. Get the value of USART_BUAD by calculating the value of USARTDIV:
   If USARTDIV=30.37, then INTDIV=30 (0x1E).
   16*0.37=5.92, the nearest integer is 6, so FRADIV=6 (0x6).
   USART_BUAD=0x1E6.

Note: If the roundness of FRADIV is 16 (overflow), the carry must be added to the integer part.

23.3.3. 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_CTL3 register. Clock pulses can be output through the CK pin.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while

the transmission is ongoing.

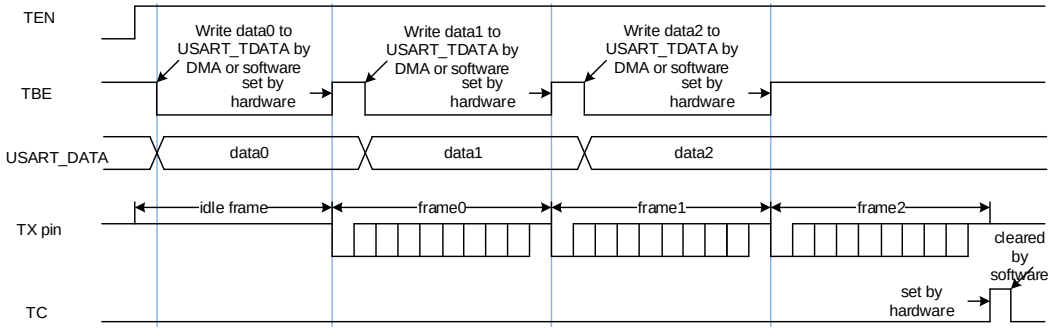
After power on, the TBE bit is high by default. Data can be written to the USART_DATA when the TBE bit in the USART_STAT0 register is asserted. The TBE bit is cleared by writing to the USART_DATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the USART_DATA 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_DATA 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_STAT0 register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in ***Figure 23-3. USART transmit procedure***. The software operating process is as follows:

1. Set the UEN bit in USART_CTL0 to enable the USART.
2. Write the WL bit in USART_CTL0 to set the data bits length.
3. Set the STB[1:0] bits in USART_CTL1 to configure the number of stop bits.
4. Enable DMA (DENT bit) in USART_CTL2 if multibuffer communication is selected.
5. Set the baud rate in USART_BAUD.
6. Set the TEN bit in USART_CTL0.
7. Wait for the TBE to be asserted.
8. Write the data to the USART_DATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 23-3. USART transmit procedure



It is necessary to wait for the TC bit to be asserted before disabling the USART or entering the power saving mode. This bit can be cleared by a software sequence: reading the USART_STAT0 register and then writing the USART_DATA register. If the multibuffer communication is selected (DENT=1), this bit can also be cleared by writing 0 directly.

23.3.4. USART receiver

After power on, the USART receiver can be enabled by the following procedure:

1. Set the UEN bit in USART_CTL0 to enable the USART.
2. Write the WL bit in USART_CTL0 to set the data bits length.
3. Set the STB[1:0] bits in USART_CTL1.
4. Enable DMA (DENR bit) in USART_CTL2 if multibuffer communication is selected.
5. Set the baud rate in USART_BAUD.
6. Set the REN bit in USART_CTL0.

After being enabled, the receiver receives a bit stream after a valid start pulse has been detected. Detection on noisy error, parity error, frame error and overrun error is performed during the reception of a frame.

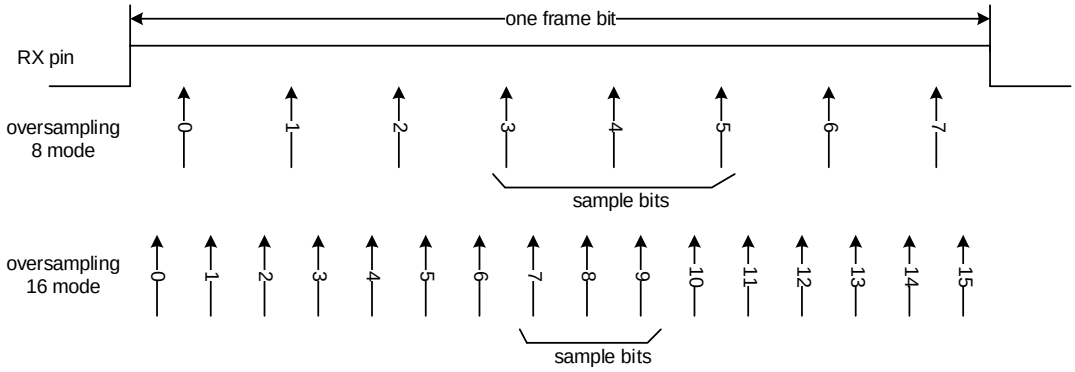
When a frame is received, the RBNE bit in USART_STAT0 is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status bits of the reception are stored in the USART_STAT0 register.

The software can get the received data by reading the USART_DATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_DATA 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 there are two or more samples of a frame bit is 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 get only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 23-4. Receiving a frame bit by oversampling method (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_STAT0 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_STAT0 register will be set. An interrupt will be 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_STAT0 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.

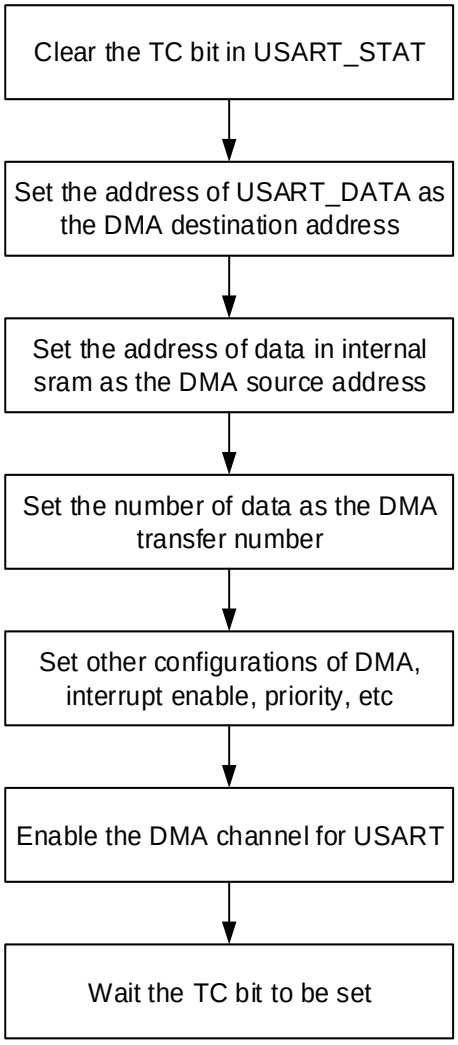
If a noise error (NERR), parity error (PERR), frame error (FERR) or overrun error (ORERR) is generated during a receiving process, then NERR, PERR, FERR or ORERR will be set at same time with RBNE. If DMA is disabled, the software needs to check whether the RBNE interrupt is caused by noise error, parity error, framing error or overflow error when the RBNE interrupt occurs.

23.3.5. 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 steps are shown in ***Figure 23-5. Configuration step when using DMA for USART transmission***.

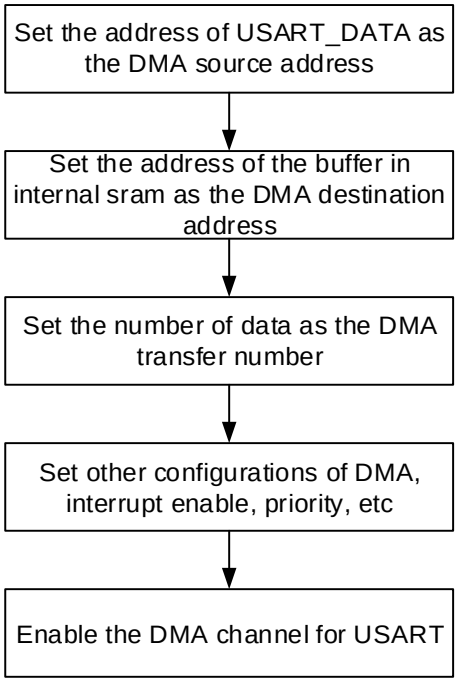
Figure 23-5. Configuration step when using DMA for USART transmission



After all of the data frames are transmitted, the TC bit in USART_STAT0 is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

When DMA is used for USART reception, DMA transfers data from the receive data buffer of the USART to the internal SRAM. The configuration steps are shown in ***Figure 23-6. Configuration steps 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_STAT0.

Figure 23-6. Configuration steps when using DMA for USART reception

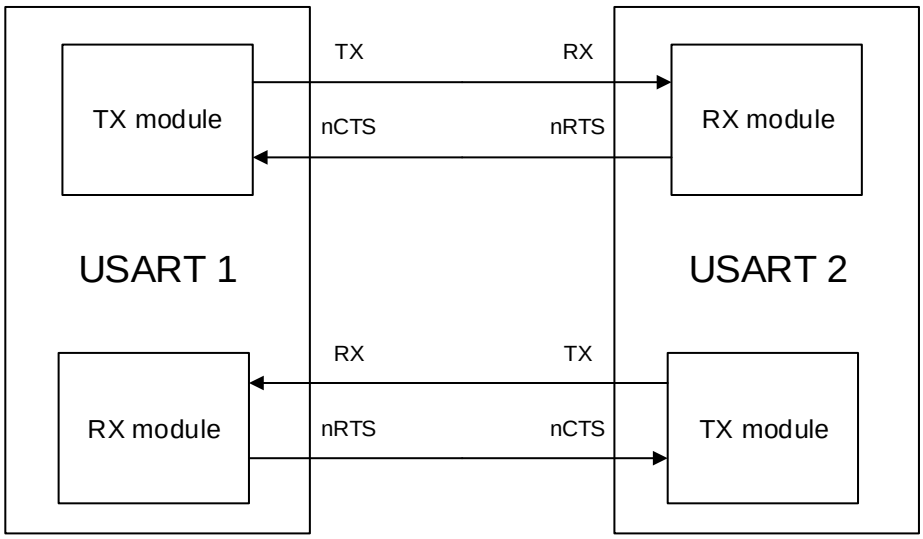


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt will be generated in the DMA module.

23.3.6. 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 23-7. Hardware flow control between two USARTs



RTS flow control

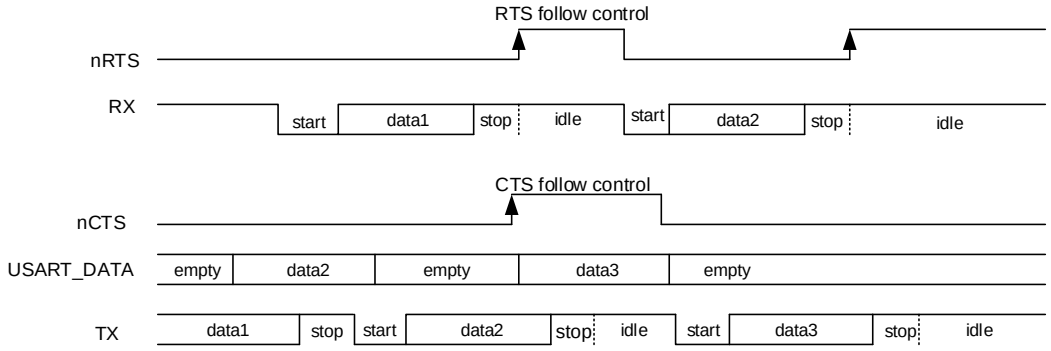
The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When

data frame is received, the nRTS signal goes high to prevent the transmitter from sending next frame. The nRTS signal keeps high when the receive buffer is full, and can be cleared by reading the USART_DATA register.

CTS flow control

The USART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in USART_STAT0 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 23-8. Hardware flow control



If the CTS flow control is enabled, the CTSF bit in USART_STAT0 is set when the nCTS pin toggles. An interrupt is generated if the CTSIE bit in USART_CTL2 is set.

23.3.7. 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 setting the RWU bit in USART_CTL0 register.

If a USART is in mute mode, all of the receive status bits cannot be set. Software can wake up the USART by clearing the RWU bit.

The USART can also be woken 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 it is woken up by an idle frame, the IDLEF bit in USART_STAT0 will not 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 bits of an address frame are the same as the ADDR[3:0] bits in the USART_CTL1 register, the hardware will clear the RWU bit and exits the mute mode. The RBNE bit will be set when the frame that wakes up the USART. The status bits are available in the USART_STAT0 register. If the LSB 4 bits of

an address frame differ from the ADDR[3:0] 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 address match method is selected, IDLEF can not be set and the receiver does not check the parity value of an address frame by default. If the PCM bit in USART_CHC and PCEN bit in USART_CTL0 are set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address flag.

The RWU bit can be written to as 0 or 1 when the RBNE is 0.

23.3.8. LIN mode

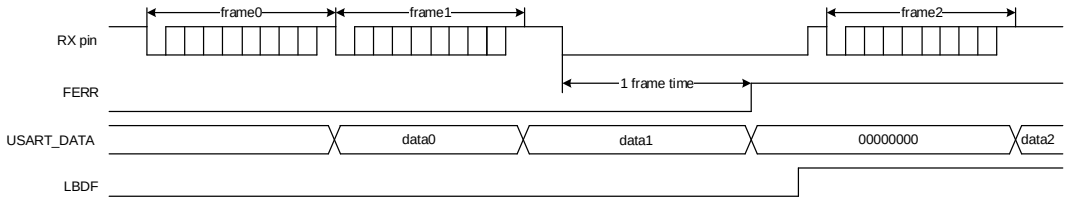
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, WL, STB[1:0] bits in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

When transmitting a normal data frame, the transmission procedure is the same as the normal USART mode. When the SBKCMD bit in USART_CTL0 is set, the USART transmits 13 '0' bits continuously, followed by 1 stop bit.

The break detection function is totally independent of the normal USART receiver. So a break frame can be detected during the idle state or during a frame. The expected length of a break frame can be selected by configuring LBLEN bit in USART_CTL1. When the RX pin is detected at low state for a time that is equal to or longer than the expected break frame length (10 bits when LBLEN=0, or 11 bits when LBLEN=1), the LBDF bit in USART_STAT0 is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

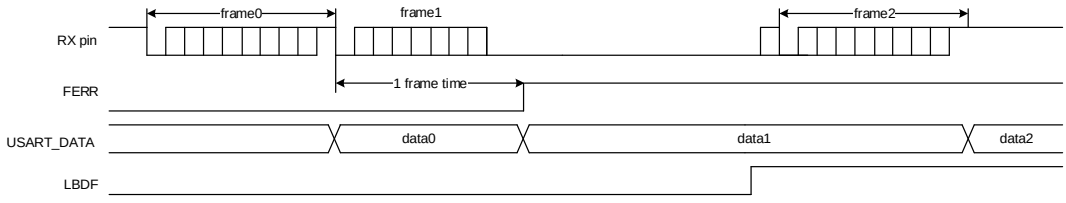
As shown in ***Figure 23-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 23-9. Break frame occurs during idle state



As shown in ***Figure 23-10. 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 23-10. Break frame occurs during a frame



23.3.9. Synchronous mode

The USART can be used for full-duplex synchronous serial communications only in master mode, by setting the CKEN bit in USART_CTL1. The LMEN bit in USART_CTL1 and SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in synchronous mode. The CK pin is the clock output of the synchronous USART transmitter, and can be only activated when the TEN bit is enabled. No clock pulse will be sent through the CK pin during the transmission of the start bit and stop bit. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the last (address flag) bit transmission. The CPH bit in USART_CTL1 can be used to determine whether data is captured on the first or the second clock edge. The CPL bit in USART_CTL1 can be used to configure the clock polarity in the USART synchronous idle state.

The CPL, CPH and CLEN bits in USART_CTL1 determine the waveform on the CK pin. Software can only change them when the USART is disabled (UEN=0).

If the REN bit in USART_CTL0 is set, the receiver works differently from the normal USART reception method. The receiver samples the data on the capture edge of the CK pin without any oversampling.

Figure 23-11. Example of USART in synchronous mode

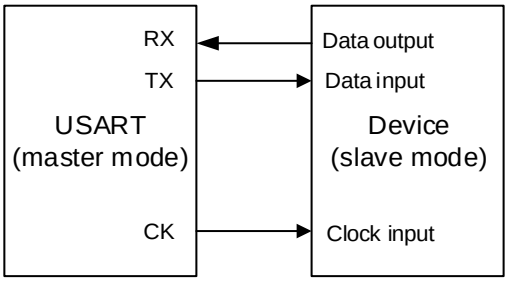
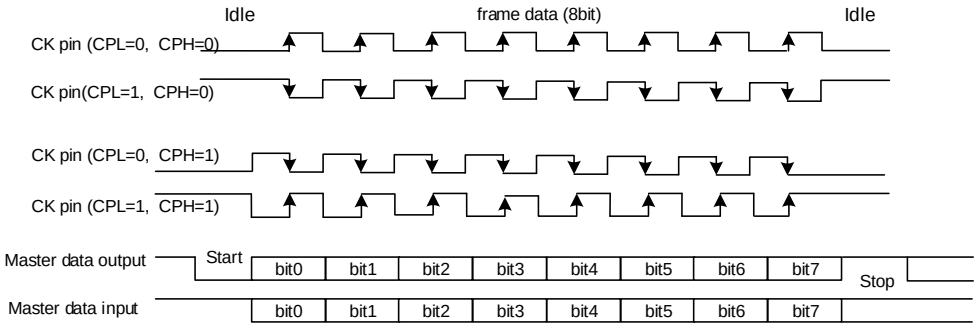


Figure 23-12. 8-bit format USART synchronous waveform (CLEN=1)

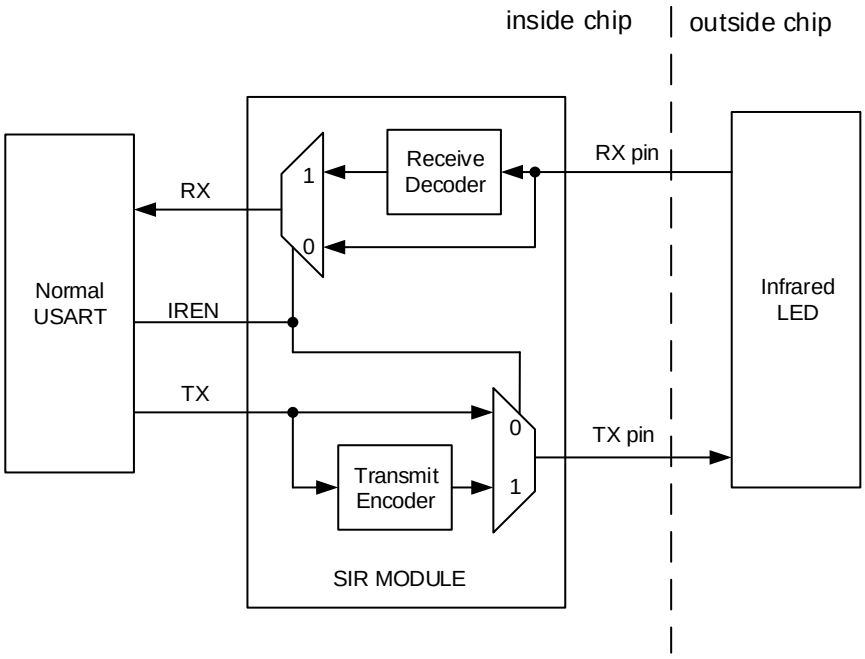


23.3.10. IrDA SIR ENDEC mode

The IrDA mode is enabled by setting the IREN bit in USART_CTL2. The LMEN, STB[1:0], CKEN bits in USART_CTL1 and HDEN, SCEN bits in USART_CTL2 should be cleared in IrDA mode.

In IrDA mode, the USART transmission data frame is modulated in the SIR transmit encoder and transmitted to the infrared LED through the TX pin. The SIR receive decoder receives the modulated signal from the infrared LED through the RX pin, and puts the demodulated data frame to the USART receiver. The baud rate should not be larger than 115200 for the encoder.

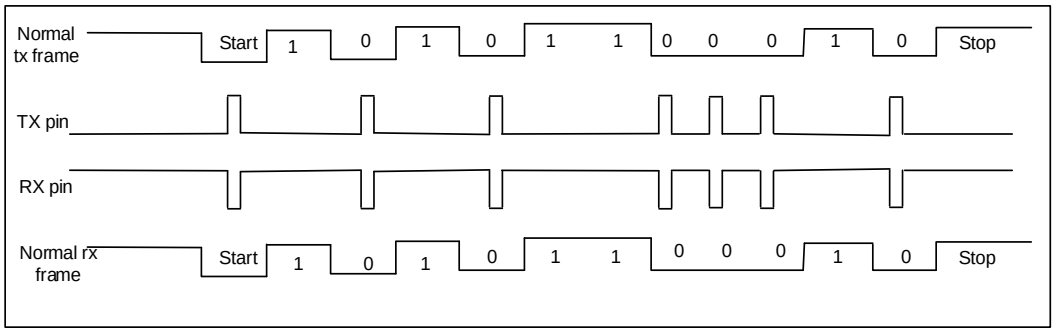
Figure 23-13. IrDA SIR ENDEC module



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 23-14. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The divide ratio is configured by the PSC[7:0] bits in USART_GP register, USART_BAUD register

needs to be configurd as 16*PSC[7:0]. The pulse width on the TX pin is 3 cycles of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

23.3.11. Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in USART_CTL2. The LMEN, CKEN bits in USART_CTL1 and SCEN, IREN bits in USART_CTL2 should be cleared in half-duplex communication mode.

Only one wire is used in half-duplex mode. The TX and RX pins are connected together internally, and the RX pin is no longer used. The TX pin should be configured as open drain mode and communication conflicts are handled by software.

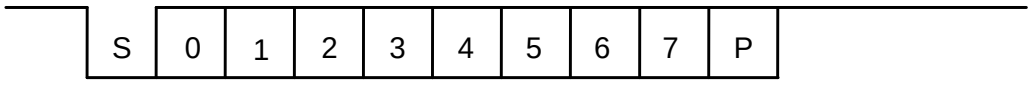
23.3.12. Smartcard (ISO7816-3) mode

The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T=0) mode and the block (T=1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be cleared in smartcard mode.

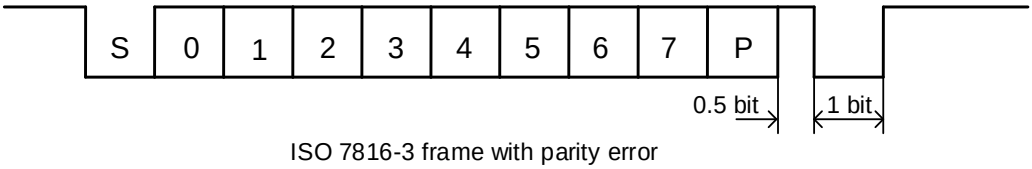
A clock is provided to the external smartcard through the CK pin after the CKEN bit is set. The clock is divided from the PCLK. The division ratio is configured by the PSC[4:0] bits in USART_GP register. The CK pin only provides a clock source to the smartcard.

The smartcard mode is a half-duplex communication protocol. When connected to a smartcard, the TX pin must be configured as open drain mode, and an external pull-up resistor will be needed, which drives a bidirectional line that is also driven by the smartcard. The data frame consists of 1 start bit, 9 data bits (1 parity bit included) and 1.5 stop bits. The 0.5 stop bit may be configured for a receiver.

Figure 23-15. ISO7816-3 frame format



ISO 7816-3 frame without parity error



ISO 7816-3 frame with parity error

Character (T=0) mode

Comparing to the timing in normal operation, the transmission time from transmit shift register to the TX pin is delayed by half baud clock, and the TC flag assertion time is delayed by a guard time that is configured by the GUAT[7:0] bits in USART_GP. In smartcard mode, the internal guard time counter starts counting up after the stop bits of the last data frame, and

the GUAT[7:0] bits should be configured as the character guard time (CGT) in ISO7816-3 protocol minus 12. The TC status is forced reset while the guard time counter is counting up. When the counter reaches the programmed value TC is asserted high.

During USART transmission, if a parity error event is detected, the smartcard may NACK the current frame by pulling down the TX pin during the last 1 bit time of the stop bits. The USART can automatically resend data according to the protocol for SCRTNUM times. An interframe gap of 2 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the framing error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE/receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the smartcard mode.

Block (T=1) mode

In block (T=1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. This timeout period is expressed in baud time units. The RTF bit in USART_STAT1 will be asserted, if no answer is received from the card before the expiration of this period. An interrupt is generated if the RTIE bit in USART_CTL3 is set. The USART generates a RBNE interrupt if the first character is received before the expiration of the RT[23:0] period. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

After the first character is received, the RT[23:0] bits should be configured to the CWT (character wait time) - 11 to enable the automatic check of the maximum interframe gap between two consecutive characters. The RTF bit in USART_STAT1 will be asserted, if the smartcard stops sending characters in the RT[23:0] period.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE=0), to count the number of received characters. The length of the block, which must be programmed to the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). The block length counter counts up from 0 to the maximum value of BL[7:0]+4. The end of the block status (EBF bit in USART_STAT1) is set after the block length counter reaches the maximum value. An interrupt is generated if the

EBIE bit in USART_CTL3 is set. The RTF bit may be set in case that an error in the block length.

If DMA is used for reception, this register field must be programmed to the minimum value (0x0) before the start of the block. With this value, the end of the block interrupt occurs after the 4th received character. The block length value can be read from the receive buffer at the third byte.

If DMA is not used for reception, the BL[7:0] bits should be firstly configured with the maximum value 0xFF to avoid generating an EBF status. The real block length value can be reconfigured to the BL[7:0] bits after the third byte is received.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

When the directed convention is selected, the LSB of the data frame is transferred first, high state on the TX pin represents logic '1', the parity check mode is even. In this case the MSBF and DINV bits in USART_CTL3 should be cleared.

When the inverse convention is selected, the MSB of the data frame is transferred first, high state on the TX pin represents logic '0', the parity check mode is even. In this case the MSBF and DINV bits in USART_CTL3 should be set.

23.3.13. USART interrupts

The USART interrupt events and flags are listed in ***Table 23-3. USART interrupt requests***.

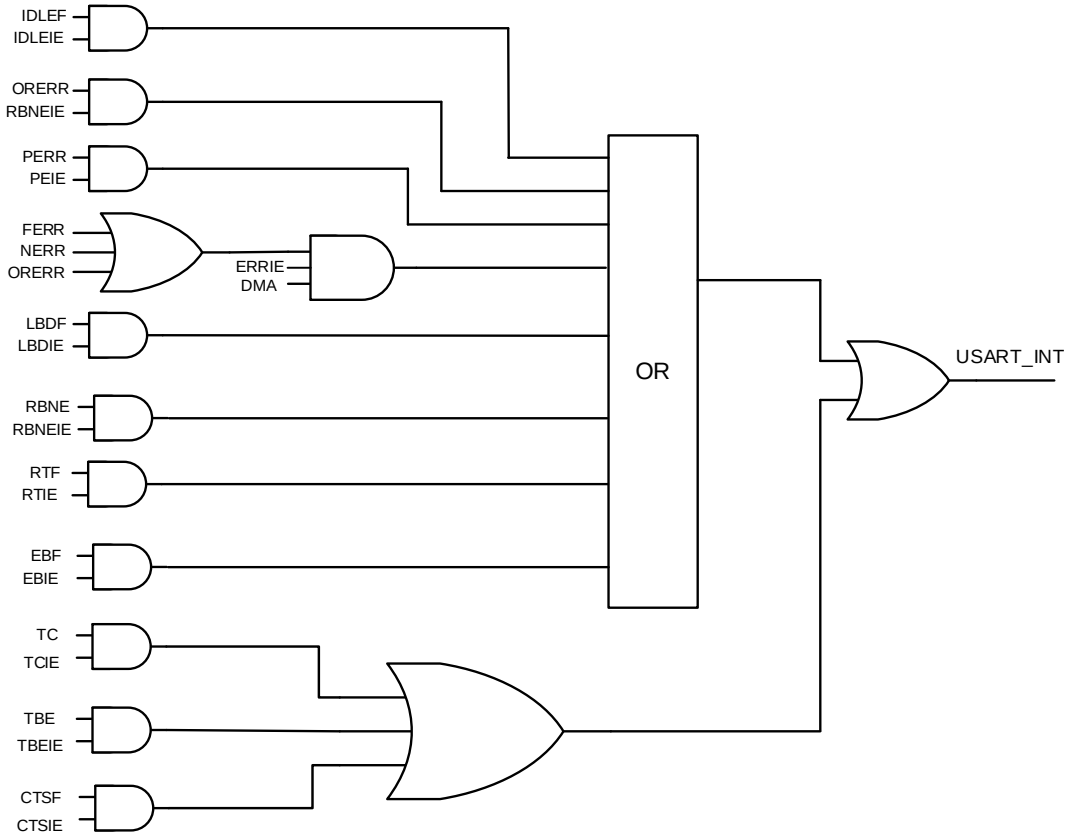
Table 23-3. USART interrupt requests

Interrupt event	Event flag	Control register	Enable Control bit
Transmit data buffer empty	TBE	USART_CTL0	TBEIE
CTS toggled flag	CTSF	USART_CTL2	CTSIE
Transmission complete	TC	USART_CTL0	TCIE
Received buff not empty	RBNE	USART_CTL0	RBNEIE
Overrun error	ORERR		
Idle frame	IDLEF	USART_CTL0	IDLEIE
Parity error	PERR	USART_CTL0	PERRIE
Break detected flag in LIN mode	LBDF	USART_CTL1	LBDIE
Receiver timeout	RTF	USART_CTL3	RTIE
End of block	EBF	USART_CTL3	EBIE
Reception errors (noise flag, overrun error, framing error) in DMA reception	NERR or ORERR or FERR	USART_CTL2	ERRIE

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 23-16. USART interrupt mapping diagram



23.4. Register definition

USART0 base address: 0x4001 3800

USART1 base address: 0x4000 4400

USART2 base address: 0x4000 4800

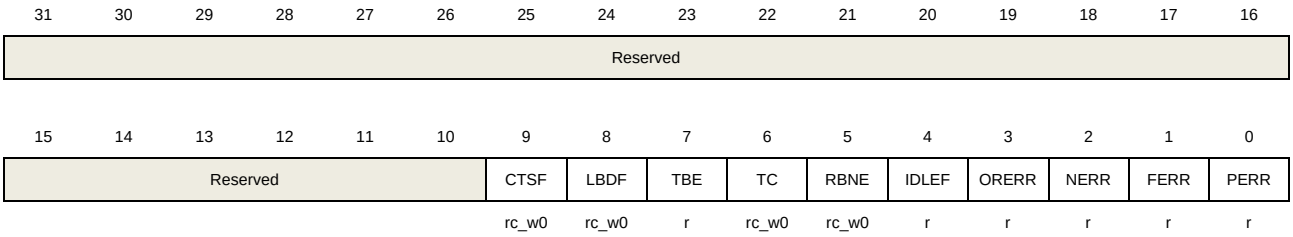
UART3 base address: 0x4000 4C00

UART4 base address: 0x4000 5000

23.4.1. Status register 0 (USART_STAT0)

Address offset: 0x00
Reset value: 0x0000 00C0

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept the reset value.
9	CTSF	CTS change flag If CTSEN bit in USART_CTL2 is set, this bit is set by hardware when the nCTS input toggles. An interrupt occurs if the CTSIE bit in USART_CTL2 is set. Software can clear this bit by writing 0 to it. 0: The status of the nCTS line does not change. 1: The status of the nCTS line has changed. This bit is not available for UART3/4.
8	LBDF	LIN break detected flag. LMEN bit in USART_CTL1 is set when LIN break is detected. An interrupt occurs if the LBDIE bit in USART_CTL1 is set. Software can clear this bit by writing 0 to it. 0: The USART does not detect a LIN break. 1: The USART has detected a LIN break.
7	TBE	Transmit data buffer empty. This bit is set after power on or when the transmit data has been transferred to the transmit shift register. An interrupt occurs if the TBEIE bit in USART_CTL0 is set. This bit is cleared when the software writes transmit data to the USART_DATA register.

0: Transmit data buffer is not empty.
1: Transmit data buffer is empty.

6	TC	Transmission complete. This bit is set after power on. If the TBE bit has been set, this bit is set when the transmission of current data is complete. An interrupt occurs if the TCIE bit in USART_CTL0 is set. Software can clear this bit by writing 0 to it. 0: Transmission of current data is not complete. 1: Transmission of current data is complete.
5	RBNE	Read data buffer not empty. This bit is set when the read data buffer is filled with a data frame, which has been received through the receive shift register. An interrupt occurs if the RBNEIE bit in USART_CTL0 is set. Software can clear this bit by writing 0 to it or by reading the USART_DATA register. 0: Read data buffer is empty. 1: Read data buffer is not empty.
4	IDLEF	IDLE frame detected flag. This bit is set when the RX pin has been detected in idle state for a frame time. An interrupt occurs if the IDLEIE bit in USART_CTL0 is set. Software can clear this bit by reading the USART_STAT0 and USART_DATA registers one by one. 0: The USART module does not detect an IDLE frame. 1: The USART module has detected an IDLE frame.
3	ORERR	Overrun error This bit is set if the RBNE is not cleared and a new data frame is received through the receive shift register. An interrupt occurs if the ERRIE bit in USART_CTL2 is set. Software can clear this bit by reading the USART_STAT0 and USART_DATA registers one by one. 0: The USART does not detect a overrun error. 1: The USART has detected a overrun error.
2	NERR	Noise error flag When the OSB bit in USART_CTL2 is reset, this bit is set if the USART detects noise on the RX pin when receiving a frame. An interrupt occurs if the ERRIE bit in USART_CTL2 is set. Software can clear this bit by reading the USART_STAT0 and USART_DATA registers one by one. 0: The USART does not detect a noise error. 1: The USART has detected a noise error.
1	FERR	Frame error flag This bit is set when the RX pin is detected low during the stop bits of a receive

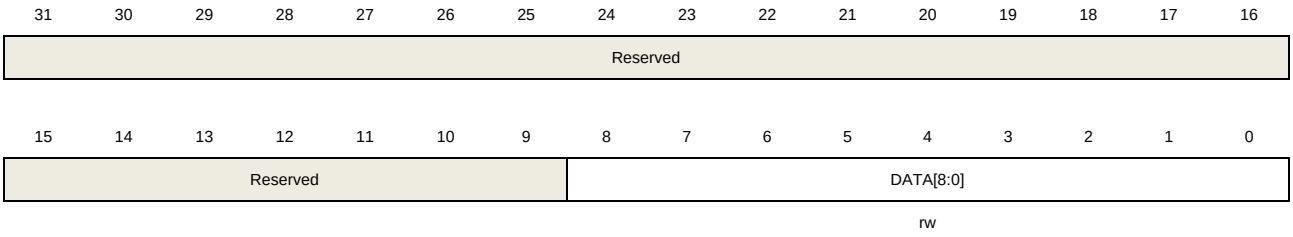
		frame. An interrupt occurs if the ERRIE bit in USART_CTL2 is set. Software can clear this bit by reading the USART_STAT0 and USART_DATA registers one by one. 0: The USART does not detect a framing error. 1: The USART has detected a framing error.
0	PERR	Parity error flag This bit is set when the parity bit of a receive frame does not match the expected parity value. An interrupt occurs if the PERRIE bit in USART_CTL0 is set. Software can clear this bit in the sequence: read the USART_STAT0 register, and then read or write the USART_DATA register. 0: The USART does not detect a parity error. 1: The USART has detected a parity error.

23.4.2. Data register (USART_DATA)

Offset: 0x04

Reset value: Undefined

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



Bits	Fields	Descriptions
31:9	Reserved	Must be kept the reset value.
8:0	DATA[8:0]	Transmit or read data value. Software can write these bits to update the transmit data or read these bits to get the receive data. If the parity check function is enabled, when transmit data is written to this register, the MSB bit (bit 7 or bit 8 depending on the WL bit in USART_CTL0) will be replaced by the parity bit.

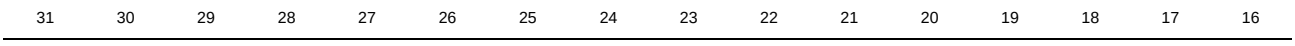
23.4.3. Baud rate register (USART_BAUD)

Address offset: 0x08

Reset value: 0x0000 0000

The software must not write this register when the USART is enabled (UEN=1).

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



Reserved	INTDIV [13:12]

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept the reset value.
17:4	INTDIV[13:0]	Integer part of baud-rate divider. **Note:** In IrDA low-power mode, the INTDIV needs to be configured as 16 times the PSC.
3:0	FRADIV[3:0]	Fraction part of baud-rate divider. Software must keep FRADIV[3] bit reset, if the oversample 8 mode is enabled.

23.4.4. Control register 0 (USART_CTL0)

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
OVSMOD	Reserved	UEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	RWU	SBKCMD
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept the reset value.
15	OVSMOD	Oversample mode 0: oversampling by 16 1: oversampling by 8 If SCEN=1, IREN=1 or LMEN=1, OVSMOD is forced to ‘0 by hardware.
14	Reserved	Must be kept the reset value.
13	UEN	USART enable 0: USART disabled. 1: USART enabled.
12	WL	Word length 0: 8 Data bits 1: 9 Data bits
11	WM	Wakeup method in mute mode

		0: wake up by idle frame. 1: wake up by address match.
10	PCEN	Parity check function enable. 0: Parity check function disabled. 1: Parity check function enabled.
9	PM	Parity mode 0: Even parity 1: Odd parity
8	PERRIE	Parity error interrupt enable. If this bit is set, an interrupt occurs when the PERR bit in USART_STAT0 is set. 0: Parity error interrupt is disabled. 1: Parity error interrupt is enabled.
7	TBEIE	Transmitter buffer empty interrupt enable. If this bit is set, an interrupt occurs when the TBE bit in USART_STAT0 is set. 0: Transmitter buffer empty interrupt is disabled. 1: Transmitter buffer empty interrupt is enabled.
6	TCIE	Transmission complete interrupt enable. If this bit is set, an interrupt occurs when the TC bit in USART_STAT0 is set. 0: Transmission complete interrupt is disabled. 1: Transmission complete interrupt is enabled.
5	RBNEIE	Read data buffer not empty interrupt and overrun error interrupt enable. If this bit is set, an interrupt occurs when the RBNE bit or the ORERR bit in USART_STAT0 is set. 0: Read data register not empty interrupt and overrun error interrupt disabled. 1: Read data register not empty interrupt and overrun error interrupt enabled.
4	IDLEIE	IDLE line detected interrupt enable. If this bit is set, an interrupt occurs when the IDLEF bit in USART_STAT0 is set. 0: IDLE line detected interrupt disabled. 1: IDLE line detected interrupt enabled.
3	TEN	Transmitter enable 0: Transmitter is disabled. 1: Transmitter is enabled.
2	REN	Receiver enable 0: Receiver is disabled. 1: Receiver is enabled.
1	RWU	Receiver wakeup from mute mode.. Software can set this bit to make the USART work in mute mode and reset this bit to wake up the USART. In wake up by idle frame mode (WM=0), this bit can be reset by hardware when an

		idle frame has been detected. In wake up by address match mode (WM=1), this bit can be reset by hardware when receiving an address match frame or set by hardware when receiving an address mismatch frame. 0: Receiver in active mode. 1: Receiver in mute mode.
0	SBKCMD	Send break command Software can set this to send a break frame. Hardware resets this bit automatically when the break frame has been transmitted. 0: Do not transmit a break frame. 1: Transmit a break frame.

23.4.5. Control register 1 (USART_CTL1)

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	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	Reserved.	LBDIE	LBLEN	Reserved	ADDR[3:0]			
	rw	rw		rw	rw	rw	rw		rw	rw		rw			

Bits	Fields	Descriptions
31:15	Reserved	Must be kept the reset value.
14	LMEN	LIN mode enable. 0: LIN mode disabled. 1: LIN mode enabled.
13:12	STB[1:0]	STOP bits length. 00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit Only 1 stop bit and 2 stop bits are available for UART3/4.
11	CKEN	CK pin enable. 0: CK pin disabled 1: CK pin enabled This bit is reserved for UART3/4.
10	CPL	CK polarity. This bit specifies the polarity of the CK pin in synchronous mode. 0: The CK pin is in low state when the USART is in idle state.

		1: The CK pin is in high state when the USART is in idle state. This bit is reserved for UART3/4.
9	CPH	CK phase. This bit specifies the phase of the CK pin in synchronous mode. 0: The capture edge of the LSB bit is the first edge of CK pin. 1: The capture edge of the LSB bit is the second edge of CK pin. This bit is reserved for UART3/4.
8	CLEN	CK length. This bit specifies the length of the CK signal in synchronous mode. 0: There are 7 CK pulses for an 8 bit frame and 8 CK pulses for a 9 bit frame 1: There are 8 CK pulses for an 8 bit frame and 9 CK pulses for a 9 bit frame This bit is reserved for UART3/4.
7	Reserved	Must be kept the reset value.
6	LBDIE	LIN break detected interrupt enable. If this bit is set, an interrupt occurs when the LBDF bit in USART_STAT0 is set. 0: LIN break detected interrupt is disabledl 1: LIN break detected interrupt is enabledl
5	LBLEN	LIN break frame length. This bit specifies the length of a LIN break frame. 0: 10 bit 1: 11 bit
4	Reserved	Must be kept the reset value.
3:0	ADDR[3:0]	Address of the USART. In wake up by address match mode (WM=1), the USART enters mute mode when the LSB 4 bits of a received frame do not equal the ADDR[3:0] bits, and wakes up when the LSB 4 bits of a received frame equal the ADDR[3:0] bits.

23.4.6. Control register 2 (USART_CTL2)

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				OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions

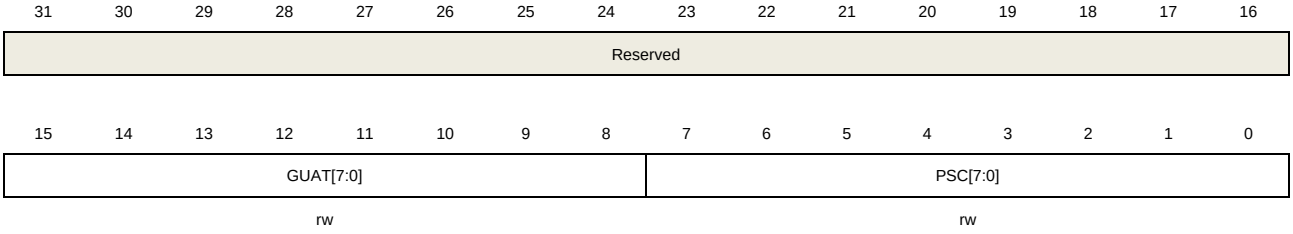
31:12	Reserved	Must be kept the reset value.
11	OSB	One sample bit method. This bit selects the sample method. When this bit is set, the USART get only one sample for a data bit instead of 3 samples per bit. The noise error flag (NERR) is disabled when the one sample bit method is selected. 0: Three sample bit method. 1: One sample bit method.
10	CTSIE	CTS interrupt enable If this bit is set, an interrupt occurs when the CTSF bit in USART_STAT0 is set. 0: CTS interrupt is disabled. 1: CTS interrupt is enabled. This bit is reserved for UART3/4.
9	CTSEN	CTS enable This bit enables the CTS hardware flow control function. 0: CTS hardware flow control disabled. 1: CTS hardware flow control enabled. This bit is reserved for UART3/4.
8	RTSEN	RTS enable This bit enables the RTS hardware flow control function. 0: RTS hardware flow control disabled. 1: RTS hardware flow control enabled. This bit is reserved for UART3/4.
7	DENT	DMA request enable for transmission. 0: DMA request is disabled for transmission. 1: DMA request is enabled for transmission.
6	DENR	DMA request enable for reception. 0: DMA request is disabled for reception. 1: DMA request is enabled for reception.
5	SCEN	Smartcard mode enable This bit enables the smartcard work mode. 0: Smartcard Mode disabled. 1: Smartcard Mode enabled. This bit is reserved for UART3/4.
4	NKEN	NACK enable in smartcard mode This bit enables the NACK transmission when parity error occurs in smartcard mode. 0: Disable NACK transmission. 1: Enable NACK transmission. This bit is reserved for UART3/4.
3	HDEN	Half-duplex enable

		This bit enables the half-duplex USART mode. 0: Half duplex mode is disabled. 1: Half duplex mode is enabled.
2	IRLP	IrDA low-power This bit selects low-power mode of IrDA mode. 0: Normal mode 1: Low-power mode
1	IREN	IrDA mode enable This bit enables the IrDA mode of USART. 0: IrDA disabled 1: IrDA enabled
0	ERRIE	Error interrupt enable When DMA request for reception is enabled (DENR=1), if this bit is set, an interrupt occurs when any one of the FERR, ORERR and NERR bits in USART_STAT0 is set. 0: Error interrupt disabled. 1: Error interrupt enabled.

23.4.7. Guard time and prescaler register (USART_GP)

Address offset: 0x18
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept the reset value.
15:8	GUAT[7:0]	Guard time value in Smartcard mode TC flag assertion time is delayed by GUAT[7:0] baud clock cycles. These bits are not available for UART3/4.
7:0	PSC[7:0]	When the USART IrDA low-power mode is enabled, these bits specify the division factor that is used to divide the peripheral clock (PCLK1/PCLK2) to generate the low-power frequency. 00000000: Reserved - never program this value. 00000001: divides by 1 00000010: divides by 2

...

11111111: divides by 255

When the USART works in IrDA normal mode, these bits must be set to 00000001.

When the USART smartcard mode is enabled, the PSC [4:0] bits specify the division factor that is used to divide the peripheral clock (APB1/APB2) to generate the smartcard clock (CK). The actual division factor is twice as the PSC [4:0] value.

00000: Reserved - never program this value.

00001: divides by 2

00010: divides by 4

...

11111: divides by 62

The PSC [7:5] bits are reserved in smartcard mode.

23.4.8. Control register 3 (USART_CTL3)

Address offset: 0x80

Reset value: 0x0000 0000

This register is not available for UART3/4.

This register 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				MSBF	DINV	TINV	RINV	Reserved		EBIE	RTIE	SCRTNUM[2:0]			RTEN
				rw	rw	rw	rw			rw	rw	rw			rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept the reset value.
11	MSBF	Most significant bit first This bit specifies the sequence of the data bits in transmission and reception. 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).
10	DINV	Data bit level inversion. This bit specifies the polarity of the data bits in transmission and reception. 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).
9	TINV	TX pin level inversion This bit specifies the polarity of the TX pin. 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).
8	RINV	RX pin level inversion This bit specifies the polarity of the RX pin. 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).
7:6	Reserved	Must be kept the reset value.
5	EBIE	Interrupt enable bit of end of block event. If this bit is set, an interrupt occurs when the EBF bit in USART_STAT1 is set. 0: End of block interrupt is disabled. 1: End of block interrupt is enabled.
4	RTIE	Interrupt enable bit of receive timeout event. If this bit is set, an interrupt occurs when the RTF bit in USART_STAT1 is set. 0: Receive timeout interrupt is disabled. 1: Receive timeout interrupt is enabled.
3:1	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 frame can be retransmitted by SCRTNUM times. If the frame is NACKed by (SCRTNUM+1) times, the FERR is set. In reception mode, a frame reception can be tried by (SCRTNUM+1) times. If the parity bit mismatch event occurs (SCRTNUM+1) times for a frame, the RBNE and PERR bits are set. When these bits are configured as 0x0, there will be no automatic retransmission in transmit mode.
0	RTEN	Receiver timeout enable. This bit enables the receive timeout counter of the USART. 0: Receiver timeout function disabled. 1: Receiver timeout function enabled.

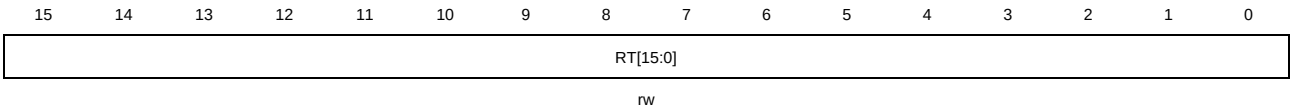
23.4.9. Receiver timeout register (USART_RT)

Address offset: 0x84
Reset value: 0x0000 0000

This register is not available for UART3/4.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BL[7:0]								RT[23:16]							
rw								rw							



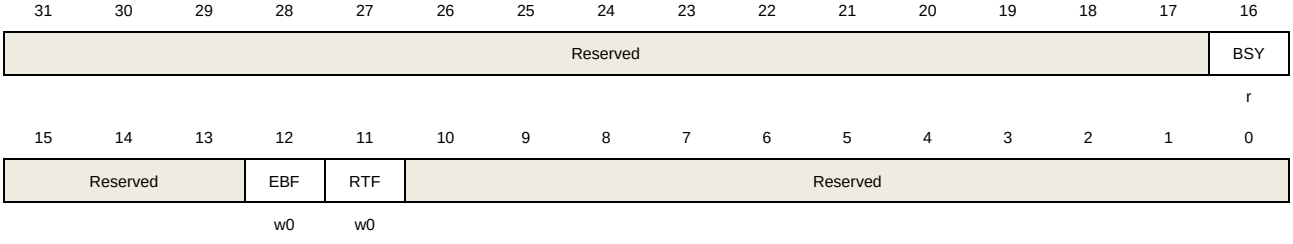
Bits	Fields	Descriptions
31:24	BL[7:0]	Block Length These bits specify the block length in Smartcard T=1 Reception. Its value equals to the number of information characters + the length of the Epilogue Field (1-LEC/2-CRC) - 1. This value, which must be programmed only once per received block, can be 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), or when the EBF bit of USART_STAT1 is written to 0, 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. If Smartcard mode is not enabled, the RTF bit of USART_STAT1 is set if no new start bit is detected longer than RT bits time after the last received character. If Smartcard mode is enabled, 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 internal timeout counter. These bits must only be programmed once per received character. **Note:** When oversampling by 16, RT[23:0] cannot be configured as 0xFFFFFF.

23.4.10. Status register 1 (USART_STAT1)

Address offset: 0x88
Reset value: 0x0000 0000

This register is not available for UART3/4.

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



Bits	Fields	Descriptions

31:17	Reserved	Must be kept the reset value.
16	BSY	Busy flag This bit is set when the USART is receiving a data frame. 0: USART reception path is idle. 1: USART reception path is working.
15:13	Reserved	Must be kept the reset value.
12	EBF	End of block flag This bit is set when the number of received bytes (from the start of the block, including the prologue) is equal or greater than BLEN + 4. An interrupt occurs if the EBIE bit in USART_CTL3 is set. Software can clear this bit by writing 0 to it. 0: End of Block event does not occur. 1: End of Block event has occurred.
11	RTF	Receiver timeout flag. This bit is set when the RX pin is in idle state for longer than RT bits time. An interrupt occurs if the RTIE bit in USART_CTL3 is set. Software can clear this bit by writing 0 to it. 0: Receiver timeout event does not occur. 1: Receiver timeout event has occurred.
10:0	Reserved	Must be kept the reset value.

23.4.11. Coherence control register (USART_CHC)

Address offset: 0xC0

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EPERR	Reserved					BCM	PCM	HCM
							rc_w0						rw	rw	rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	EPERR	Early parity error flag. This flag will be set as soon as the parity bit has been detected, which is before RBNE flag. This flag is cleared by writing 0. 0: No parity error is detected 1: Parity error is detected

7:3	Reserved	Must be kept at reset value.
2	BCM	Break frame coherence mode. 0: When the CTS flow is enabled, the transmitter does not check the nCTS input state to send a break. 1: When the CTS flow is enabled, the transmitter checks the nCTS input state to send a break. This bit is reserved for UART3/4.
1	PCM	Parity check coherence mode. 0: In case of wakeup by an address match: the MSB bit of the data is taken into account to identify an address but not the parity bit. And the receiver does not check the parity of the address data (PERR is not set in case of a parity error). 1: In case of wakeup by an address match, the receiver check the parity of the address data and PERR is set in case of a parity error.
0	HCM	Hardware flow control coherence mode. 0: nRTS signal equals to RBNE bit in USART_STAT0 register 1: nRTS signal is set when the last data bit (parity bit when PCE is set) has been sampled This bit is reserved for UART3/4.

24. Serial peripheral interface/Inter-IC sound (SPI/I2S)

24.1. Overview

The SPI/I2S module can communicate with external devices using the SPI protocol or the I2S audio protocol.

The Serial Peripheral Interface (SPI) provides a SPI protocol of data transmission and reception function in master or slave mode. Both full-duplex and simplex communication modes are supported, with hardware CRC calculation and checking. Quad-SPI master mode is also supported in SPI0.

The inter-IC sound (I2S) supports four audio standards: I2S Philips standard, MSB justified standard, LSB justified standard, and PCM standard. I2S works at either master or slave mode for transmission and reception.

24.2. Characteristics

24.2.1. SPI characteristics

- Master or slave operation with full-duplex or simplex mode.
- Separate transmit and receive buffer, 16 bits wide.
- Data frame size can be 8 or 16 bits.
- Bit order can be LSB first or MSB first.
- Software and hardware NSS management.
- Hardware CRC calculation, transmission and checking.
- Transmission and reception using DMA.
- SPI TI mode supported.
- SPI NSS pulse mode supported.
- Quad-SPI configuration available in master mode (only in SPI0).

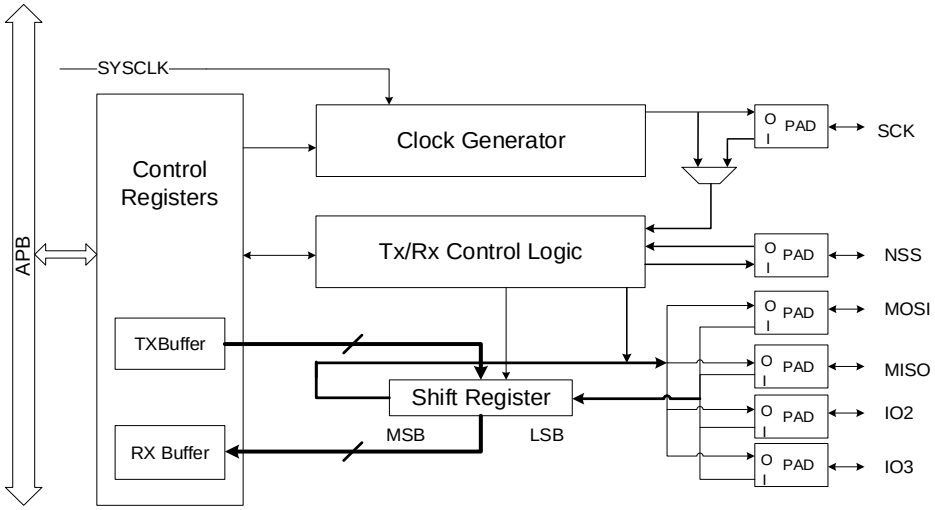
24.2.2. I2S characteristics

- Master or slave operation with transmission or reception mode.
- 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.

24.3. SPI function overview

24.3.1. SPI block diagram

Figure 24-1. Block diagram of SPI



24.3.2. SPI signal description

Normal configuration (Not Quad-SPI Mode)

Table 24-1. SPI signal description

Pin Name	Direction	Description
SCK	I / O	Master: SPI Clock Output Slave: SPI Clock Input
MISO	I / O	Master: Data reception line Slave: Data transmission line Master with Bidirectional mode: Not used Slave with Bidirectional mode: Data transmission and reception Line.
MOSI	I / O	Master: Data transmission line Slave: Data reception line Master with Bidirectional mode: Data transmission and reception Line. Slave with Bidirectional mode: Not used
NSS	I / O	Software NSS Mode: Not Used Master in hardware NSS mode: when NSSDRV=1, it is NSS output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master

Pin Name	Direction	Description
		application. Slave in Hardware NSS Mode: NSS input, as a chip select signal for slave.

Quad-SPI configuration

SPI is in single wire mode by default and enters into Quad-SPI mode after QMOD bit in SPI_QCTL register is set (only available in SPI0). Quad-SPI mode can only work at master mode.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 24-2. Quad-SPI signal description

Pin Name	Direction	Description
SCK	O	SPI Clock Output
MOSI	I / O	Transmission or Reception Data 0 line
MISO	I / O	Transmission or Reception Data 1 line
IO2	I / O	Transmission or Reception Data 2 line
IO3	I / O	Transmission or Reception Data 3 line
NSS	O	NSS output

24.3.3. SPI clock timing and data format

CKPL and CKPH bits in SPI_CTL0 register decide the timing of SPI clock and data signal. The CKPL bit decides the SCK level when idle and CKPH bit decides either first or second clock edge is a valid sampling edge. These bits take no effect in TI mode.

Figure 24-2. SPI timing diagram in normal mode

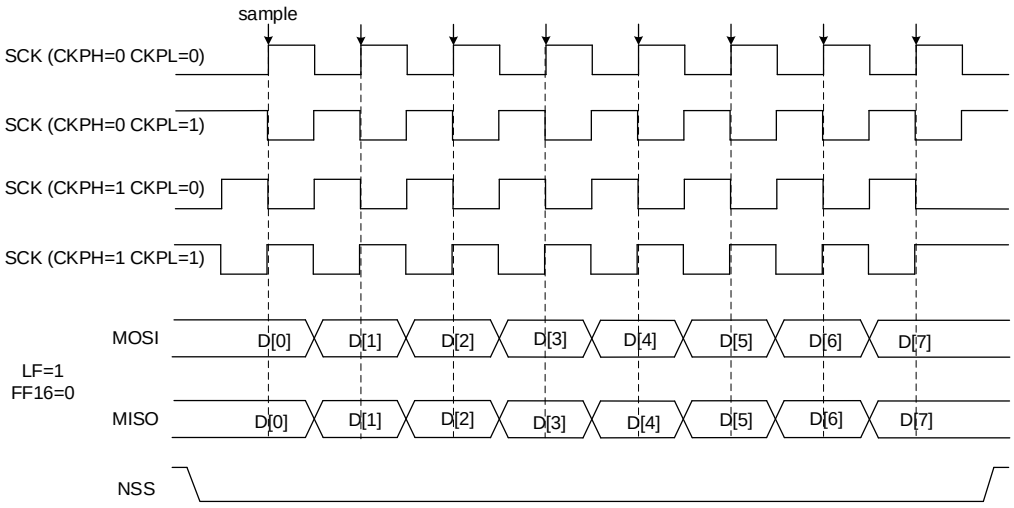
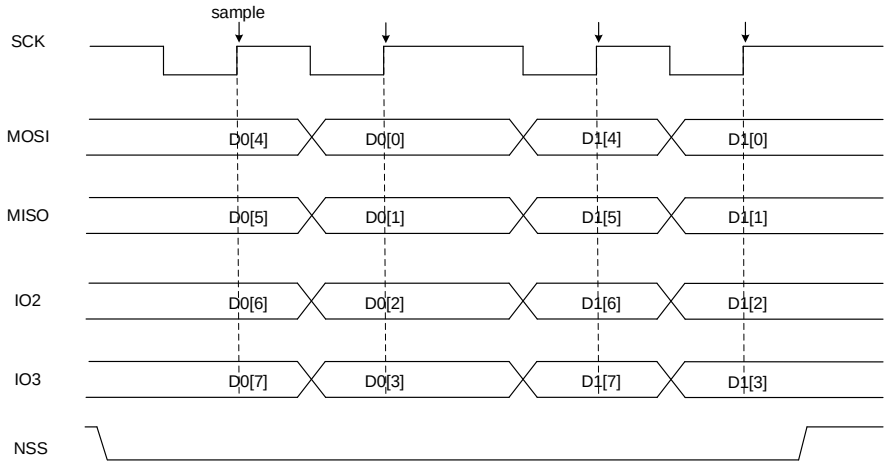


Figure 24-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



In normal mode, the length of data is configured by the FF16 bit in the SPI_CTL0 register. Data length is 16 bits if FF16=1, otherwise is 8 bits. The data frame length is fixed to 8 bits in Quad-SPI mode.

Data order is configured by LF bit in SPI_CTL0 register, and SPI will first send the LSB if LF=1, or the MSB if LF=0. The data order is fixed to MSB first in TI mode.

24.3.4. 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 24-3. NSS function in slave mode

Mode	Register configuration	Description
Slave hardware NSS mode	MSTMOD = 0 SWNSSEN = 0	SPI slave gets NSS level from NSS pin.
Slave software NSS mode	MSTMOD = 0 SWNSSEN = 1	SPI slave NSS level is determined by the SWNSS bit. SWNSS = 0: NSS level is low SWNSS = 1: NSS level is high

Master mode

In master mode (MSTMOD=1) if the application uses multi-master connection, NSS can be configured to hardware input mode (SWNSSEN=0, NSSDRV=0) or software mode (SWNSSEN=1). Then, once the NSS pin (in hardware NSS mode) or the SWNSS bit (in

software NSS mode) goes low, the SPI automatically enters to 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 24-4. NSS function in master mode

Mode	Register configuration	Description
Master hardware NSS output mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	Applicable to single-master mode. The master uses the NSS pin to control the SPI slave device. At this time, the NSS is configured as the hardware output mode.
Master hardware NSS input mode	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	Applicable to multi-master mode. At this time, NSS is configured as hardware input mode. Once the NSS pin is pulled low, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be set to 1.
Master software NSS mode	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV: Don't care	Applicable to multi-master mode. Once SWNSS = 0, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be 1.
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV: Don't care	The slave can use hardware or software NSS mode.

24.3.5. SPI operation modes

Table 24-5. 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	MSTMOD = 1	MOSI: Not used

Mode	Description	Register Configuration	Data Pin Usage
	unidirectional connection	RO = 1 BDEN = 0 BDOEN: Don't care	MISO: Reception
MTB	Master Transmission with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Transmission MISO: Not used
MRB	Master Reception with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Reception MISO: Not used
SFD	Slave Full-Duplex	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Transmission
STU	Slave Transmission with unidirectional connection	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Not used MISO: Transmission
SRU	Slave Reception with unidirectional connection	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Not used
STB	Slave Transmission with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Not used MISO: Transmission
SRB	Slave Reception with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Not used MISO: Reception

Figure 24-4. A typical Full-duplex connection

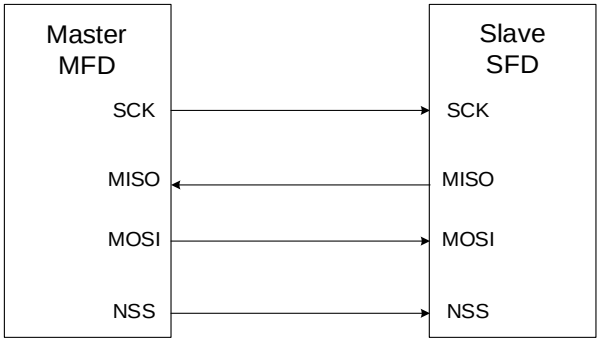


Figure 24-5. A typical simplex connection (Master: Receive, Slave: Transmit)

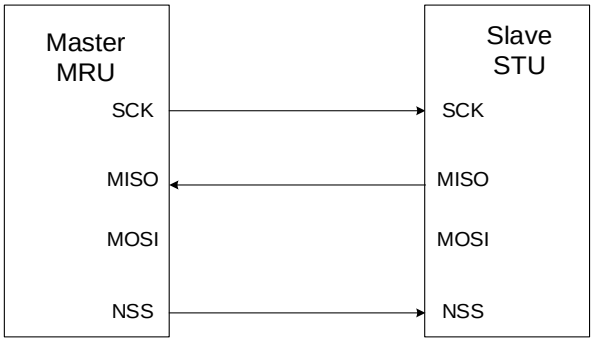


Figure 24-6. A typical simplex connection (Master: Transmit only, Slave: Receive)

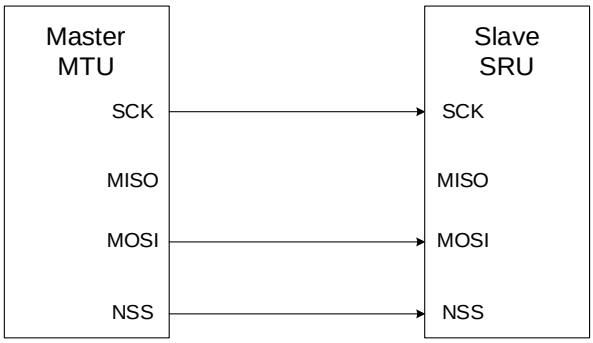
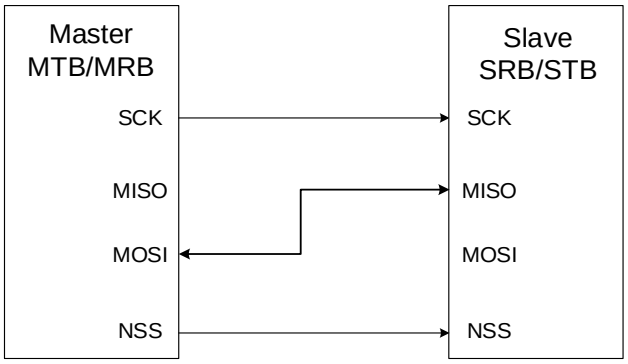


Figure 24-7. A typical bidirectional connection



SPI initialization sequence

Before transmiting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode or slave TI mode is used, program the PSC [2:0] bits in SPI_CTL0 register to generate SCK with desired baud rate or configure the Td time in TI mode, otherwise, ignore this step.
2. Program data format (FF16 bit in the SPI_CTL0 register).
3. Program the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
4. Program the frame format (LF bit in the SPI_CTL0 register).
5. Program the NSS mode (SWNSSEN and NSSDRV bits in the SPI_CTL0 register) according to the application's demand as described above in ***NSS function*** section.

6. If TI mode is used, set TMOD bit in SPI_CTL1 register, otherwise, ignore this step.
7. If NSSP mode is used, set NSSP bit in SPI_CTL1 register, otherwise, ignore this step.
8. Configure MSTMOD, RO, BDEN and BDOEN depending on the operation modes described above.
9. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.
10. Enable the SPI (set the SPIEN bit).

Note: During communication, CKPH, CKPL, MSTMOD, PSC[2:0] and LF bits should not be changed.

SPI basic transmission and reception sequence

Transmission sequence

After the initialization sequence, the SPI is enabled and stays at idle state. In master mode, the transmission starts when the application writes a data into the transmit buffer. In slave mode, the transmission starts when SCK clock signal begins to toggle at SCK pin and NSS level is low, so application should ensure that data is already written into transmit buffer before the transmission starts in slave mode.

When SPI begins to send a data frame, it first loads this data frame from the data buffer to the shift register and then begins to transmit the loaded data frame, TBE (transmit buffer empty) flag is set after the first bit of this frame is transmited. After TBE flag is set, which means the transmit buffer is empty, the application should write SPI_DATA register again if it has more data to transmit.

In master mode, software should write the next data into SPI_DATA register before the transmission of current data frame is completed if it desires to generate continuous transmission.

Reception sequence

The incoming data will be moved from shift register to the receive buffer after the last valid sample clock and also, RBNE (receive buffer not empty) will be set. The application should read SPI_DATA register to get the received data and this will clear the RBNE flag automatically. In MRU and MRB modes, hardware continuously sends clock signal to receive the next data frame, while in full-duplex master mode (MFD), hardware only receives the next data frame when the transmit buffer is not empty.

SPI operation sequence in different modes (Not Quad-SPI, TI mode or NSSP mode)

In full-duplex mode, either MFD or SFD, application should monitor the RBNE and TBE flags and follow the sequences described above.

The transmission mode (MTU, MTB, STU or STB) is similar to full-duplex mode, except that application should ignore the RBNE and OVRE flags and only perform transmission sequence

described above.

In master reception mode (MRU or MRB), the behavior is different from full-duplex mode or transmission mode. In MRU or MRB mode, the SPI continuously generates SCK just after SPI is enabled, 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 full-duplex mode, except that application should ignore the TBE flag and only perform reception sequence described above.

SPI TI mode

SPI TI mode takes NSS as a special frame header flag signal and its operation sequence is similar to normal mode described above. The modes described above (MFD, MTU, MRU, MTB, MRB, SFD, STU, SRU, STB and SRB) are still supported in TI mode. While, in TI mode the CKPL and CKPH bits in SPI_CTL0 registers take no effect and the SCK sample edge is falling edge.

Figure 24-8. Timing diagram of TI master mode with discontinuous transfer

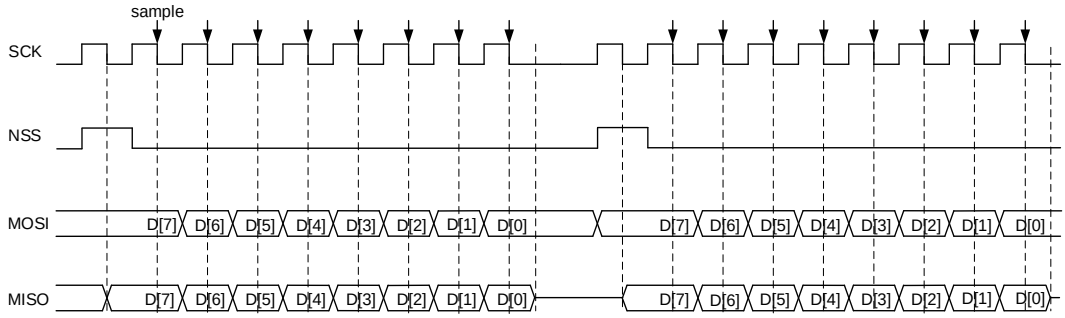
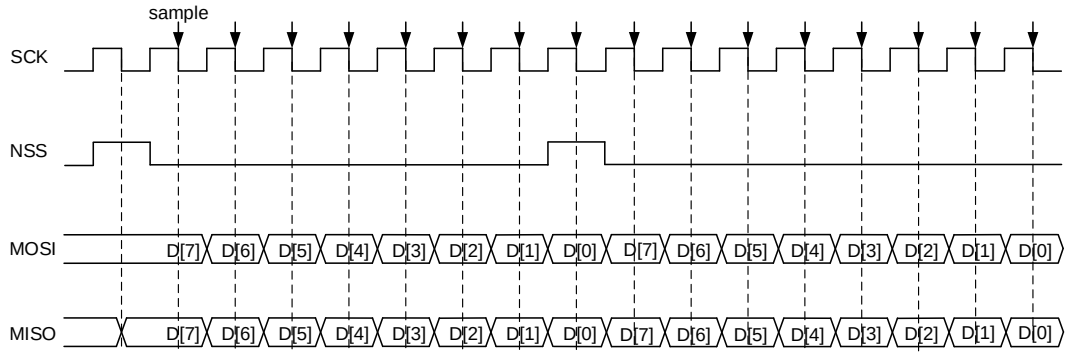
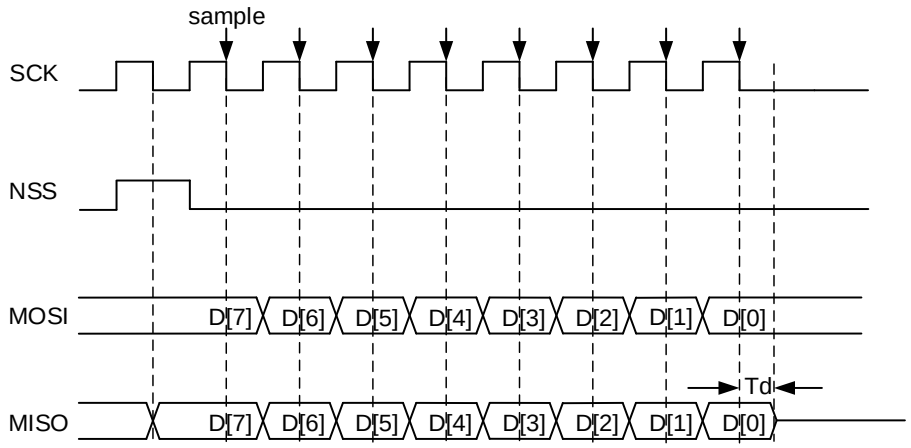


Figure 24-9. Timing diagram of TI master mode with continuous transfer



In master TI mode, SPI can perform continuous or non-continuous transfer. If the master writes SPI_DATA register fast enough, the transfer is continuous, otherwise non-continuous. In non-continuous transfer there is an extra header clock cycle before each byte. While in continuous transfer, the extra header clock cycle only exists before the first byte and the following bytes' header clock is overlaid at the last bit of pervious bytes.

Figure 24-10. Timing diagram of TI slave mode



In Slave TI mode, after the last rising edge of SCK in transfer, the slave begins to transmit the LSB bit of the last data byte, and after a half-bit time, the master begins to sample the line. To make sure that the master samples the right value, the slave should continue to drive this bit after the falling sample edge of SCK for a period of time before releasing the pin. This time is called T_d. T_d is decided by PSC [2:0] bits in SPI_CTL0 register.

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (24\text{-}1)$$

For example, if PSC[2:0] = 010, $T_d$is 9*Tpclk.

In slave mode, the slave also monitors the NSS signal and sets an error flag FERR if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

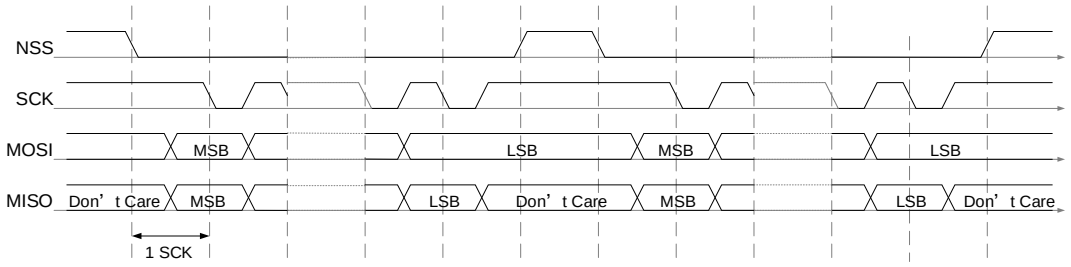
NSS pulse mode operation sequence

This function is controlled by NSSP bit in SPI_CTL1 register, for this function to fully take place, several additional conditions must be met, users must also set the device into master mode, and frame format should follow the normal SPI protocol, and set the data capture edge to first clock transition.

In summary: NSSP = 1; MSTMOD = 1; CKPH = 0;

When active, a pluse duration of least 1 SCK clock priod is inserted between successive data frames depending on internal data transmit buffer status, multiple SCK clock cycle interval is possible if the transfer buffer stays empty. This function is designed for single master-slave configuration for the slave to latch data. The following diagram depicts its timing diagram.

Figure 24-11. Timing diagram of NSS pulse with continuous transmit



Quad-SPI mode operation sequence

The Quad-SPI mode is designed to control quad SPI flash.

In order to enter Quad-SPI mode, the software should first verify that the TBE bit is set and TRANS bit is cleared, then set QMOD bit in SPI_QCTL register. In Quad-SPI mode, BDEN, BDOEN, CRCEN, CRCNT, FF16, RO and LF in SPI_CTL0 register should be kept cleared and MSTMOD should be set to ensure that SPI is in master mode. SPIEN, PSC, CKPL and CKPH should be configured as desired.

There are 2 operation modes in Quad-SPI mode: quad write and quad read, decided by QRD bit in SPI_QCTL register.

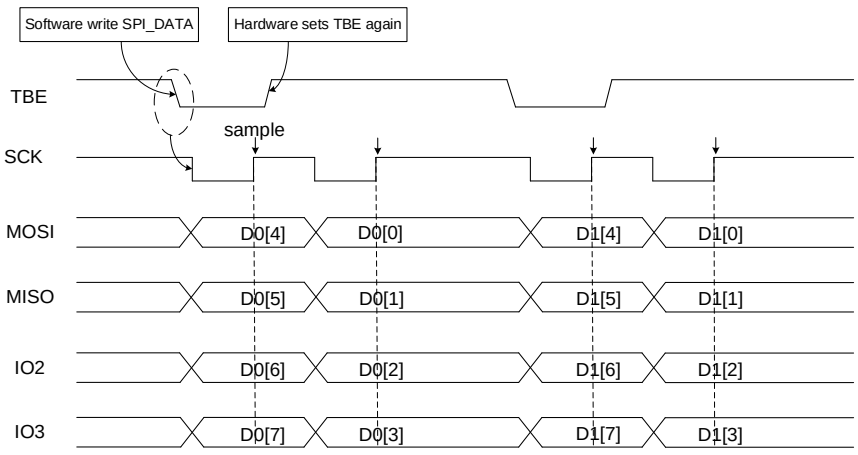
Quad write operation

SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Once SPI starts transmission, it always checks TBE status at the end of a frame and stops when condition is not met.

The operation flow for transmitting in quad mode:

1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 based on your application requirements.
2. Set QMOD bit in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0.
3. Write the byte to SPI_DATA register and the TBE will be cleared.
4. Wait until TBE is set by hardware again before writing the next byte.

Figure 24-12. Timing diagram of quad write operation in Quad-SPI mode



Quad read operation

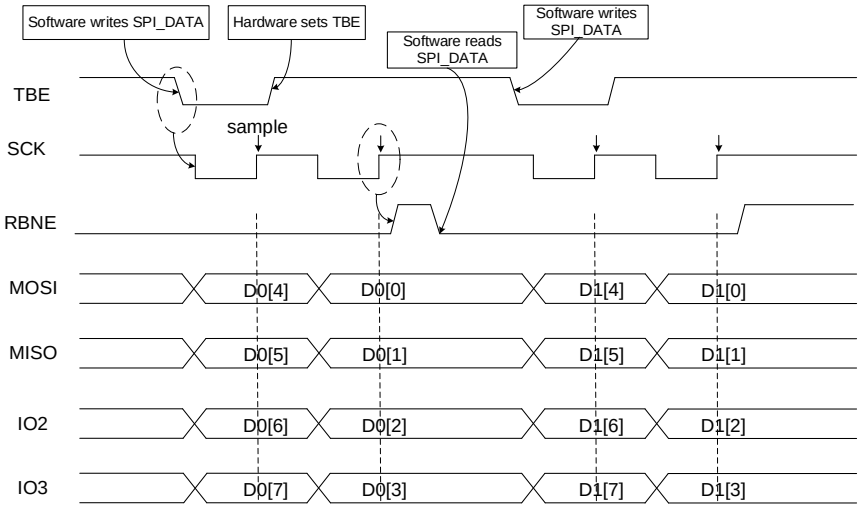
SPI works in quad read mode when QMOD and QRD are both set in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as input pins. SPI begins to generate clock on SCK line as soon as a data is written into SPI_DATA (TBE is cleared) and SPIEN is set.

Writing data into SPI_DATA is only to generate SCK clocks, so the written data can be any value. Once SPI starts transmission, it always checks SPIEN and TBE status at the end of a frame and stops when condition is not met. So, software should always write dummy data into SPI_DATA to make SPI generate SCK.

The operation flow for receiving in quad mode:

1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 register based on your application requirements.
2. Set QMOD and QRD bits in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0 register.
3. Write an arbitrary byte (for example, 0xFF) to SPI_DATA register.
4. Wait until the RBNE flag is set and read SPI_DATA to get the received byte.
5. Write an arbitrary byte (for example, 0xFF) to SPI_DATA to receive the next byte.

Figure 24-13. Timing diagram of quad read operation in Quad-SPI mode



SPI disabling sequence

Different sequences are used to disable the SPI in different operation modes:

MFD SFD

Wait for the last RBNE flag and then receive the last data. Confirm that TBE=1 and TRANS=0. At last, disable the SPI by clearing SPIEN bit.

MTU MTB STU STB

Write the last data into SPI_DATA and wait until the TBE flag is set and then wait until the TRANS flag is cleared. Disable the SPI by clearing SPIEN bit.

MRU MRB

After getting the second last RBNE flag, read out this data and delay for a SCK clock time and then, disable the SPI by clearing SPIEN bit. Wait until the last RBNE flag is set and read

out the last data.

SRU SRB

Application can disable the SPI when it doesn't want to receive data, and then wait until the TRANS=0 to ensure the on-going transfer completes.

TI mode

The disabling sequence of TI mode is the same as the sequences described above.

NSS pulse mode

The disabling sequence of NSS pulse mode is the same as the sequences described above.

Quad-SPI mode

Before leaving quad wire mode or disabling SPI, software should first check that, TBE bit is set and TRANS bit is cleared, then the QMOD bit in SPI_QCTL register and SPIEN bit in SPI_CTL0 register are cleared.

24.3.6. DMA function

The DMA function frees the application from data writing and reading process during transfer, thus improve the system efficiency.

DMA function in SPI is enabled by setting DMATEN and DMAREN bits in SPI_CTL1 register. To use DMA function, application should first correctly configure DMA modules, then configure SPI module according to the initialization sequence, at last enable SPI.

After being enabled, If DMATEN is set, SPI will generate a DMA request each time 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 RBNE=1, then DMA will acknowledge to this request and read data from the SPI_DATA register automatically.

24.3.7. CRC function

There are two CRC calculators in SPI: one for transmission and the other for reception. The CRC calculation uses the polynomial in SPI_CRCPOLY register.

Application can switch on the CRC function by setting CRCEN bit in SPI_CTL0 register. The CRC calculators continuously calculate CRC for each bit transmitted and received on lines, and the calculated CRC values can be read from SPI_TCRC and SPI_RCRC register.

To transmit the calculated CRC value, application should set the CRCNT bit in SPI_CTL0 register after the last data is written to the transmit buffer. In full-duplex mode (MFD or SFD) the SPI treats the incoming data as a CRC value when it transmits a CRC and will check the received CRC value. In reception mode (MRB, MRU, SRU and SRB), the application should set the CRCNT bit after the second-last data frame is received. When CRC checking fails,

the CRCERR flag will be set.

If DMA function is enabled, application doesn't need to operate CRCNT bit and hardware will automatically process the CRC transmitting and checking.

Note: When SPI is in slave mode and CRC function is enable, the CRC calculator is sensitive to input SCK clock whether SPI is enable or not. The software must enable CRC only when the clock is stable to avoid wrong CRC calculation. And when SPI works as a slave, the NSS internal signal needs to be kept low between the data phase and CRC phase.

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).

24.3.8. SPI interrupts

Status flags

- **Transmit buffer empty flag (TBE)**

This bit is set when the transmit buffer is empty, the software can write the next data to the transmit buffer by writing the SPI_DATA register.

- **Receive buffer not empty flag (RBNE)**

This bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer, and software can read the data by reading the SPI_DATA register.

- **SPI Transmitting On-Going flag (TRANS)**

TRANS is a status flag to indicate whether the transfer is on-going or not. It is set and cleared by internal 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 receive buffer contents won't be covered with the newly incoming data, so the newly incoming data is lost.

- **Format Error (FERR)**

In slave TI mode, the slave also monitors the NSS signal and set an error flag if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

- **CRC Error (CRCERR)**

When the CRCEN bit is set, the CRC value received in the SPI_RCRC register will be compared with the CRC value received immediately after the last frame of data. The CRCERR will set when they are different.

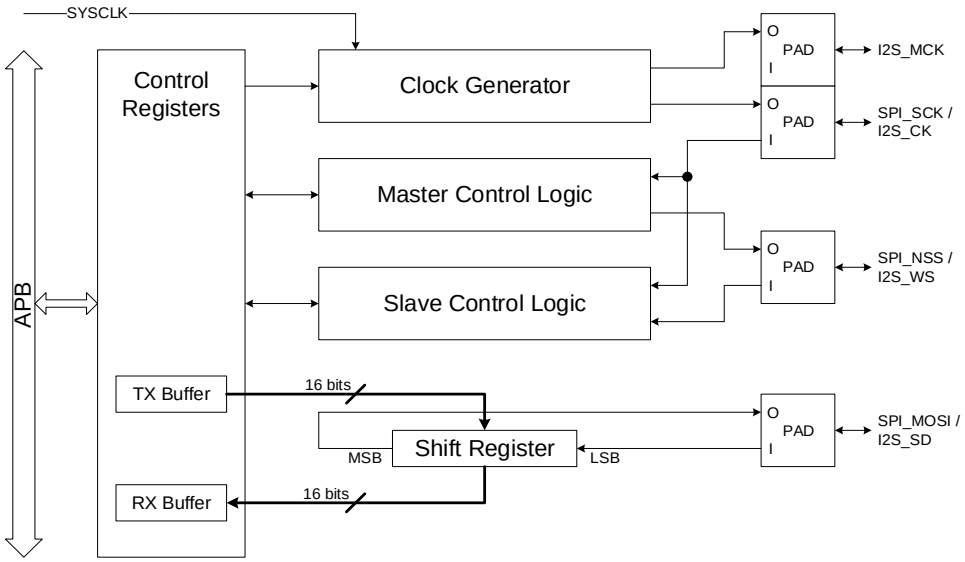
Table 24-6. SPI interrupt requests

Flag	Description	Clear Method	Interrupt Enable bit
TBE	Transmit buffer empty	Write SPI_DATA register.	TBEIE
RBNE	Receive buffer not empty	Read SPI_DATA register	RBNEIE
CONFERR	Configuration fault error	Read or write SPI_STAT register, then write SPI_CTL0 register.	ERRIE
RXORERR	Rx overrun error	Read SPI_DATA register, then read SPI_STAT register.	
CRCERR	CRC error	Write 0 to CRCERR bit	
FERR	TI mode format error	Write 0 to FERR bit	

24.4. I2S function overview

24.4.1. I2S block diagram

Figure 24-14. Block diagram of I2S



There are five sub modules to support I2S function, including control registers, clock generator, master control logic, slave control logic and shift register. All the user configuration registers are implemented in the control registers module, including the TX buffer and RX buffer. The clock generator is used to produce I2S communication clock in master mode. The master control logic is implemented to generate the I2S_WS signal and control the communication in master mode. The slave control logic is implemented to control the communication in slave mode according to the received I2S_CK and I2S_WS. The shift register handles the serial data transmission and reception on I2S_SD.

Note: The configuration value of the I2S serial clock must be set to less than 1/6 of the corresponding PCLK clock (excluding 1/6).

24.4.2. 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.

24.4.3. 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 bits and CHLEN bit in the SPI_I2SCTL register. Since the channel length must be greater than or equal to the data length, four packet types are available. They are 16-bit data packed in 16-bit frame, 16-bit data packed in 32-bit frame, 24-bit data packed in 32-bit frame, and 32-bit data packed in 32-bit frame. The data buffer for transmission and reception is 16-bit wide. In the case that the data length is 24 bits or 32 bits, two write or read operations to or from the SPI_DATA register are needed to complete a frame. In the case that the data length is 16 bits, only one write or read operation to or from the SPI_DATA register is needed to complete 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 24-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

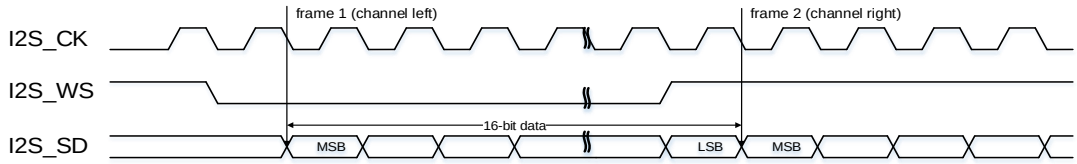
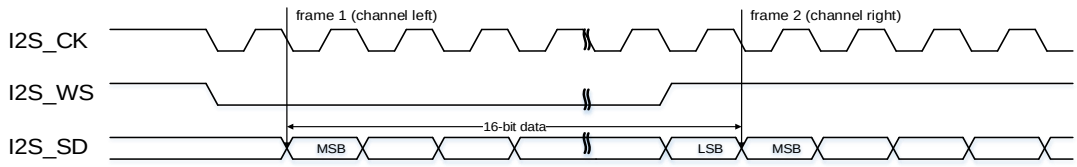


Figure 24-16. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)



When the packet type is 16-bit data packed in 16-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete a frame.

Figure 24-17. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

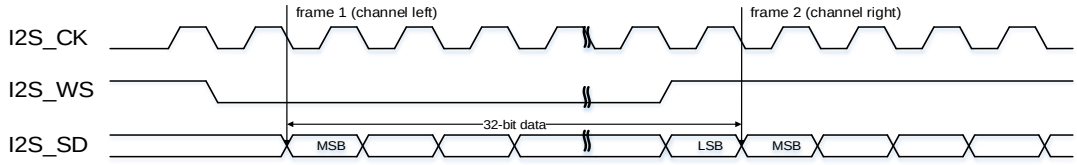
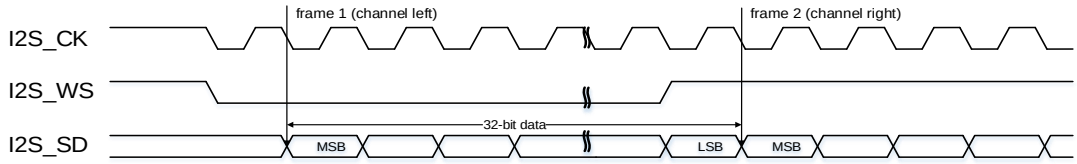


Figure 24-18. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)



When the packet type is 32-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete 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 24-19. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

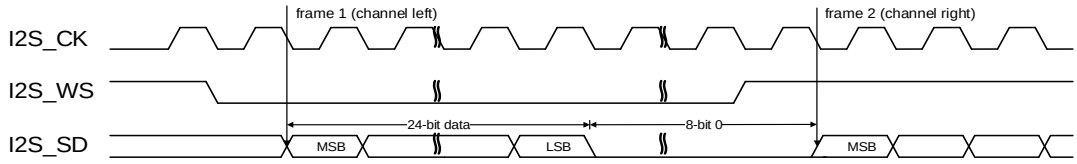
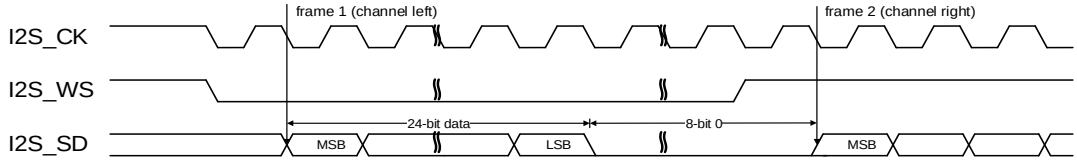


Figure 24-20. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete 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 24-21. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

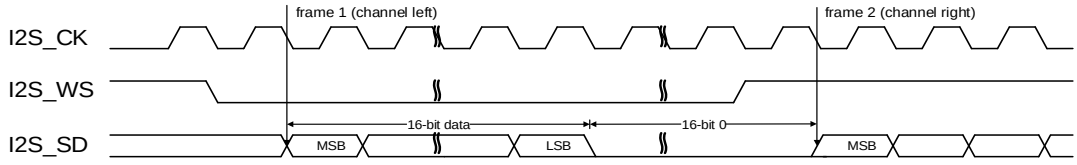
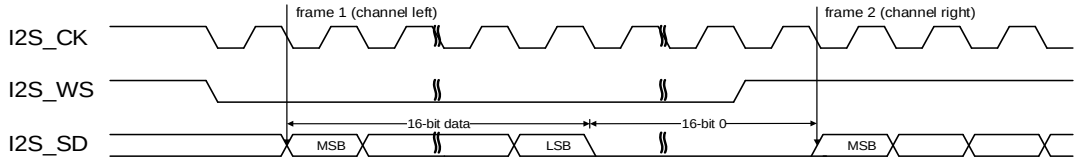


Figure 24-22. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete a frame. The 16 remaining 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 24-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

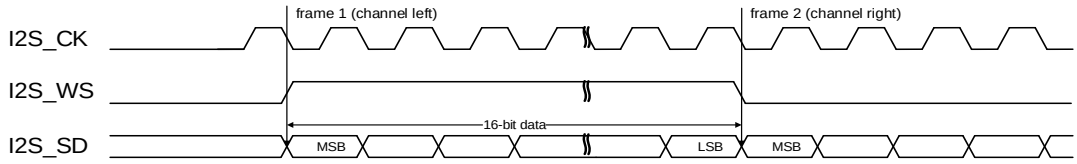


Figure 24-24. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

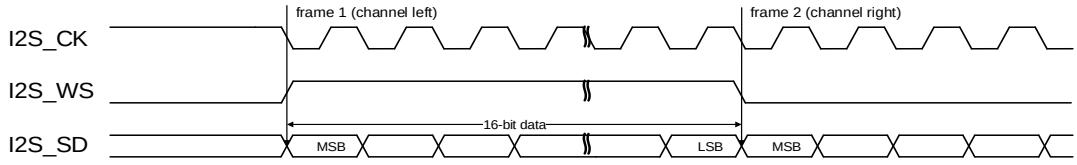


Figure 24-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

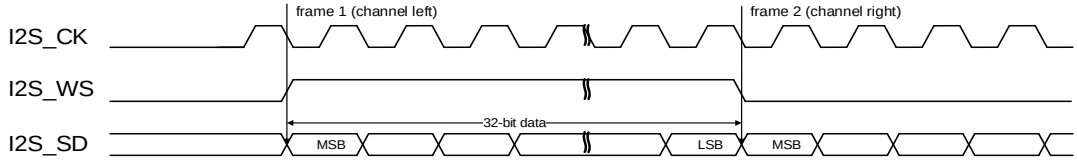


Figure 24-26. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

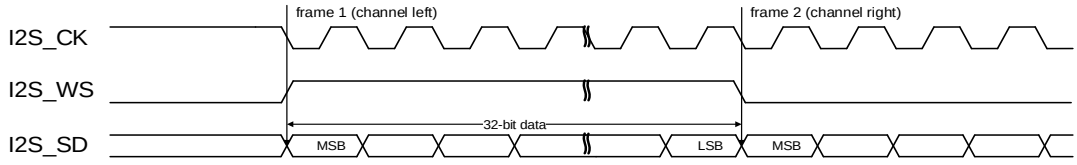


Figure 24-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

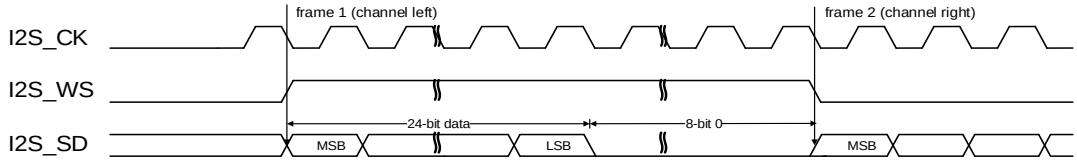


Figure 24-28. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

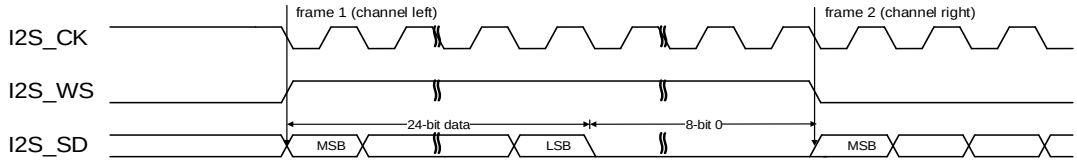


Figure 24-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

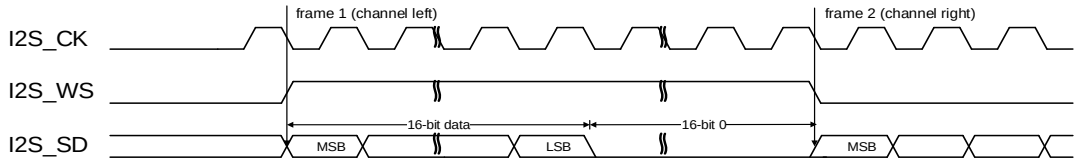
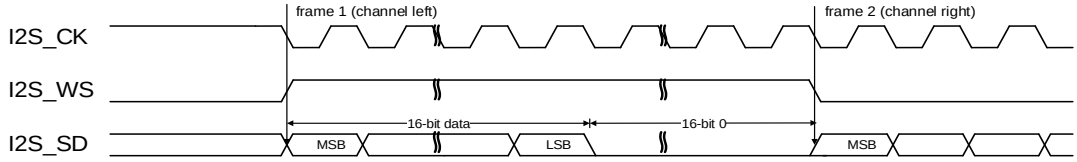


Figure 24-30. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



LSB justified standard

For LSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. In the case that the channel length is equal to the data length, LSB justified standard and MSB justified standard are exactly the same. In the case that the channel length is greater than the data length, the valid data is aligned to LSB for LSB justified standard while the valid data is aligned to MSB for MSB justified standard. The timing diagrams for the cases that the channel length is greater than the data length are shown below.

Figure 24-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

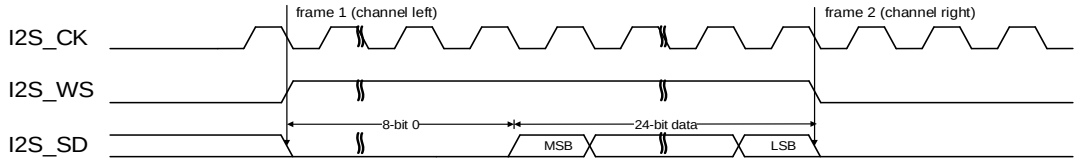
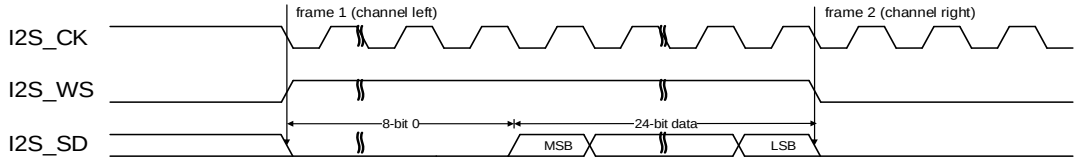


Figure 24-32. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete 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 24-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

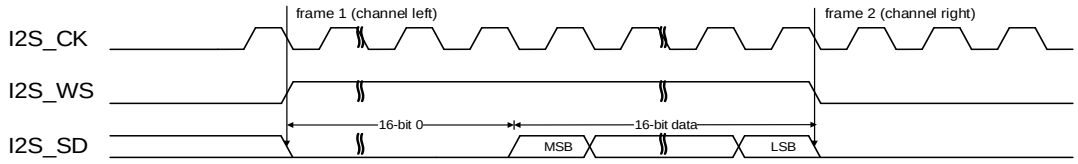
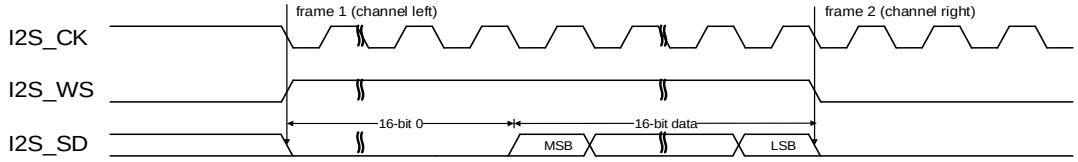


Figure 24-34. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete a frame. The 16 remaining 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 24-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

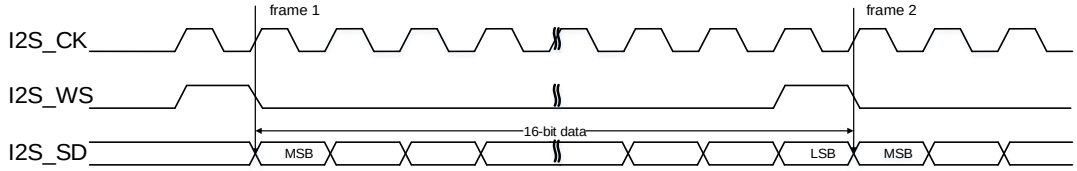


Figure 24-36. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

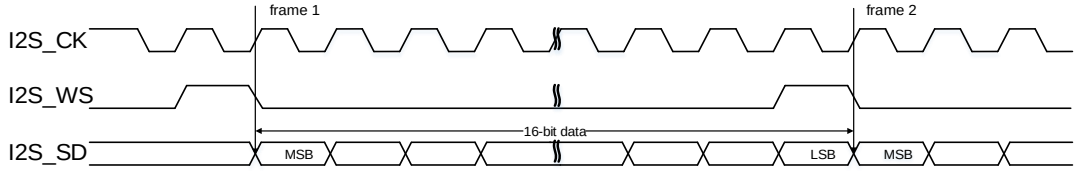


Figure 24-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

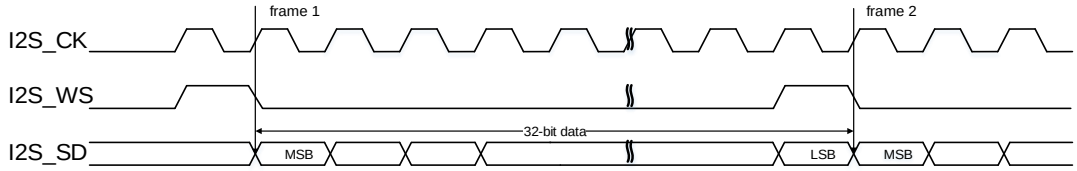


Figure 24-38. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

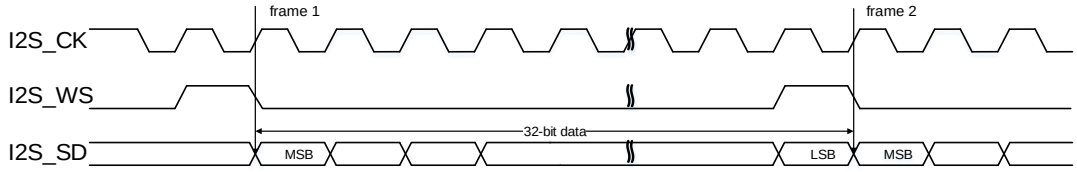


Figure 24-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

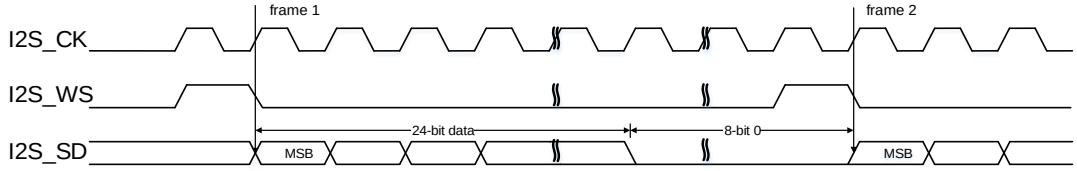


Figure24-40. PCM standard short frame synchronization mode timing diagram

(DTLEN=01, CHLEN=1, CKPL=1)

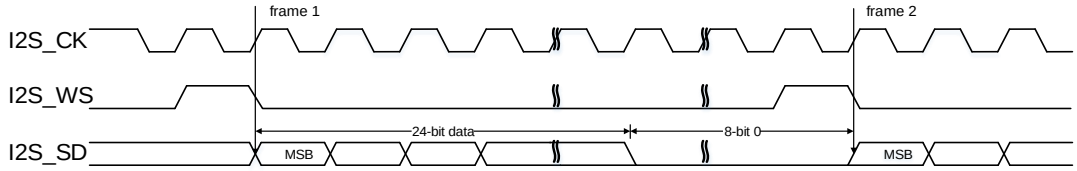


Figure 24-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

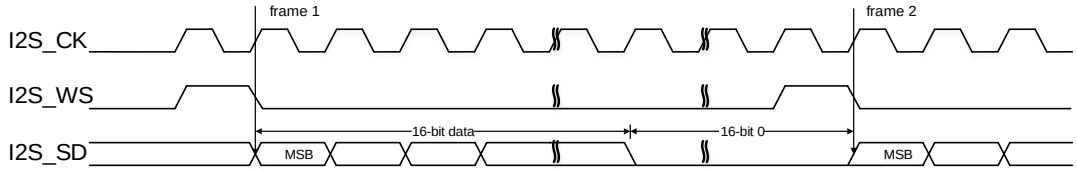
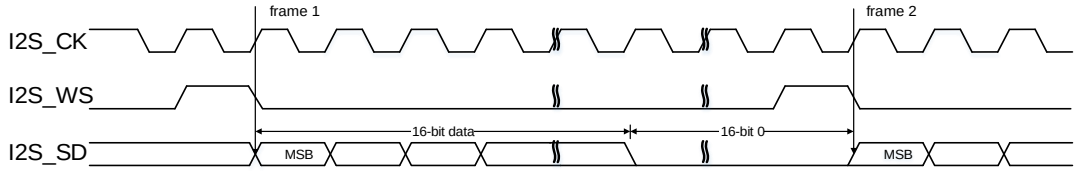


Figure 24-42. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 24-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

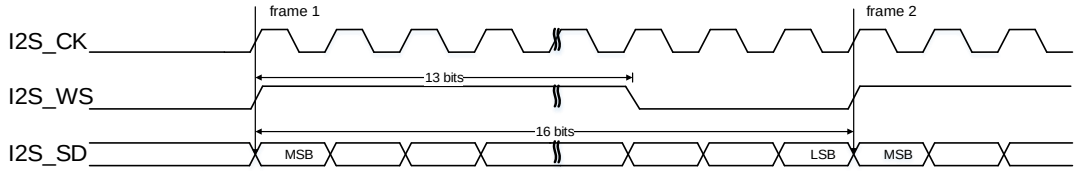


Figure24-44. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

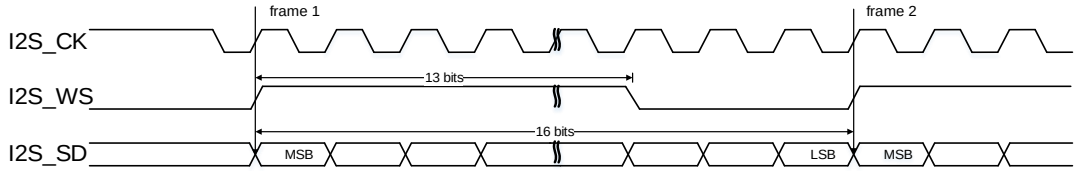


Figure 24-45. PCM standard long frame synchronization mode timing diagram

(DTLEN=10, CHLEN=1, CKPL=0)

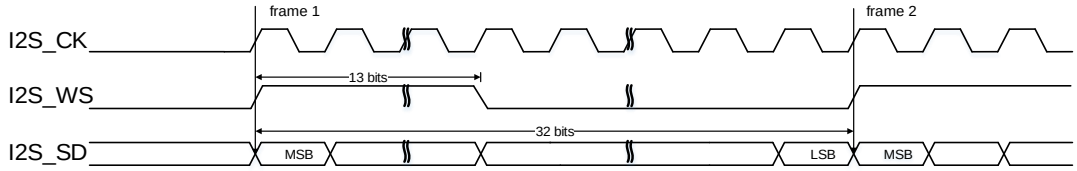


Figure 24-46. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

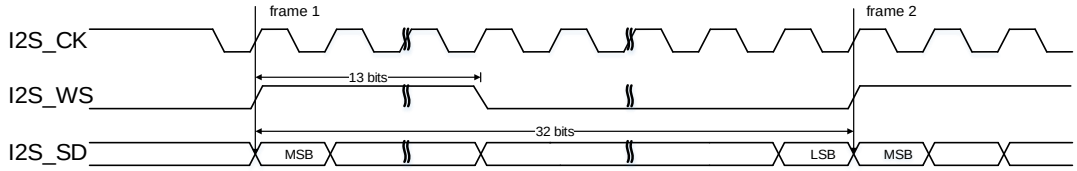


Figure 24-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

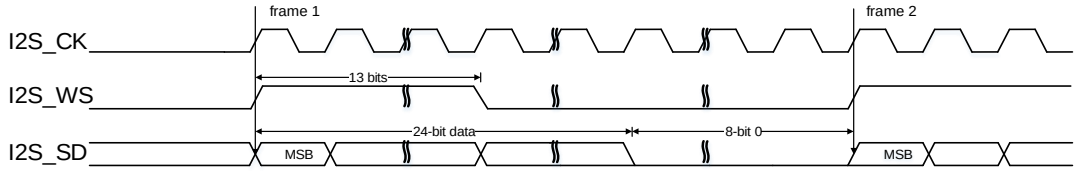


Figure 24-48. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

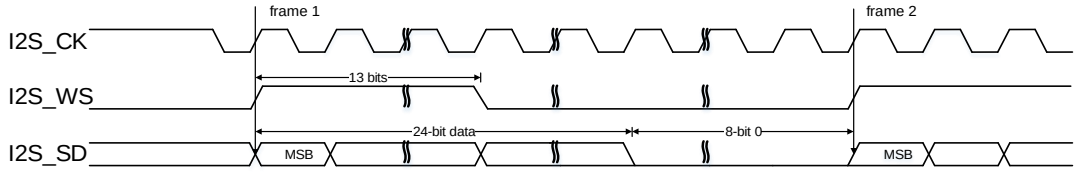


Figure 24-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

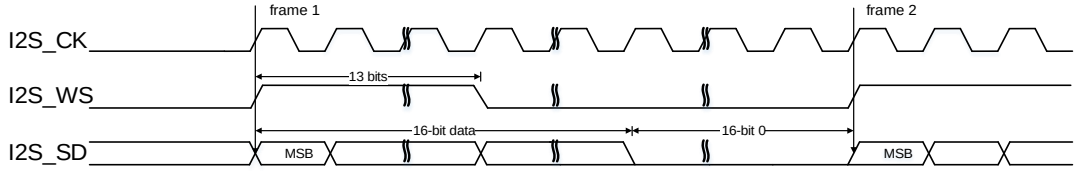
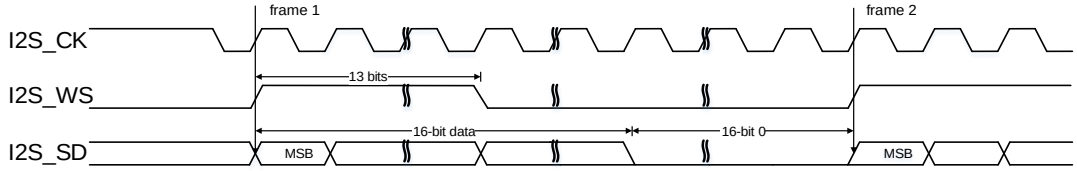
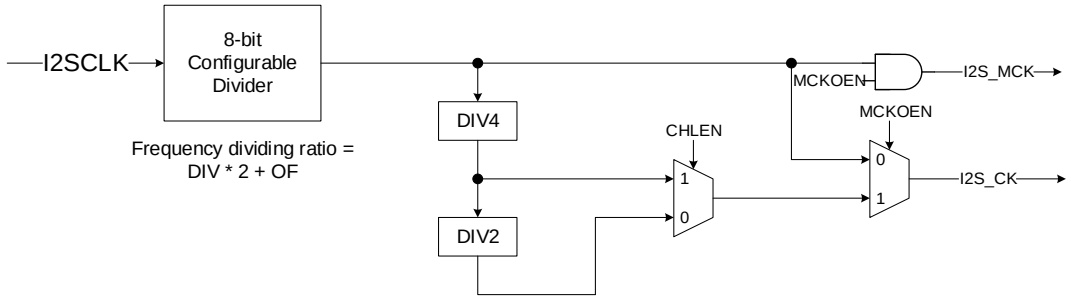


Figure 24-50. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



24.4.4. I2S clock

Figure 24-51. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as ***Figure 24-51. Block diagram of I2S clock generator***. The I2S interface clocks are configured by the DIV bits, the OF bit, the MCKOEN bit in the SPI_I2SPSC register and the CHLEN bit in the SPI_I2SCTL register. The source clock is the system clock or CK_PLL1 (CK_SYS/ CK_PLL1). It is recommended to configure the I2S clock source to be more than 6 times higher than the I2S serial clock. The I2S bitrate can be calculated by the formulas shown in ***Table 24-7. I2S bitrate calculation formulas***.

Table 24-7. I2S bitrate calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

The relationship between audio sampling frequency (Fs) and I2S bitrate is defined by the following formula:

Fs = I2S bitrate / (number of bits per channel * number of channels)

So, in order to get the desired audio sampling frequency, the clock generator needs to be configured according to the formulas listed in ***Table 24-8. Audio sampling frequency calculation formulas***.

Table 24-8. Audio sampling frequency calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

24.4.5. Operation

Operation modes

The operation mode is selected by the I2SOPMOD bits in the SPI_I2SCTL register. There are four available operation modes, including master transmission mode, master reception mode, slave transmission mode, and slave reception mode. The direction of I2S interface signals for each operation mode is shown in the *Table 24-9. Direction of I2S interface signals for each operation mode*.

Table 24-9. Direction of I2S interface signals for each operation mode

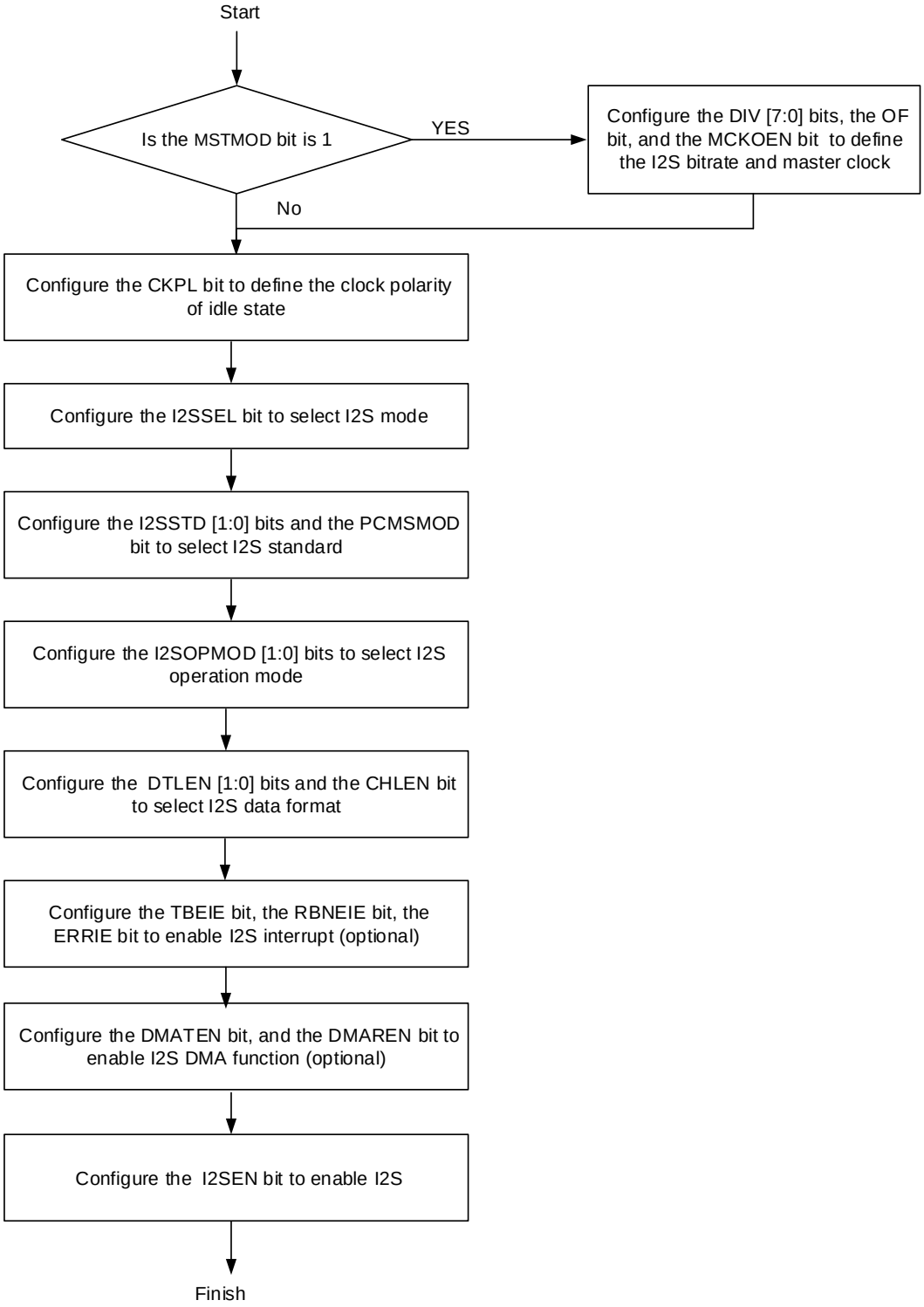
Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
Master transmission	output or NU(1)	output	output	output
Master reception	output or NU(1)	output	output	input
Slave transmission	input or NU(1)	input	input	output
Slave reception	input or NU(1)	input	input	input

1. NU means the pin is not used by I2S and can be used by other functions.

I2S initialization sequence

I2S initialization sequence is shown as below *Figure 24-52. I2S initialization sequence*.

Figure 24-52. I2S initialization sequence



I2S master transmission sequence

The TBE flag is used to control the transmission sequence. As is mentioned before, the TBE flag indicates that the transmit buffer is empty, and an interrupt will be generated if the TBEIE bit in the SPI_CTL1 register is set. At the beginning, the transmit buffer is empty (TBE is high) and no transmission sequence is processing in the shift register. When a half word is written

to the SPI_DATA register (TBE goes low), the data is transferred from the transmit buffer to the shift register (TBE goes high) immediately. At the moment, the transmission sequence begins.

The data is parallel loaded into the 16-bit shift register, and shifted out serially to the I2S_SD pin, MSB first. The next data should be written to the SPI_DATA register, when the TBE flag is high. After a write operation to the SPI_DATA register, the TBE flag goes low. When the current transmission finishes, the data in the transmit buffer is loaded into the shift register, and the TBE flag goes back high. Software should write the next audio data into SPI_DATA register before the current data finishes, otherwise, the audio data transmission is not continuous.

For all standards except PCM, the I2SCH flag is used to distinguish the channel side to which the data to transfer belongs. 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 switch off I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

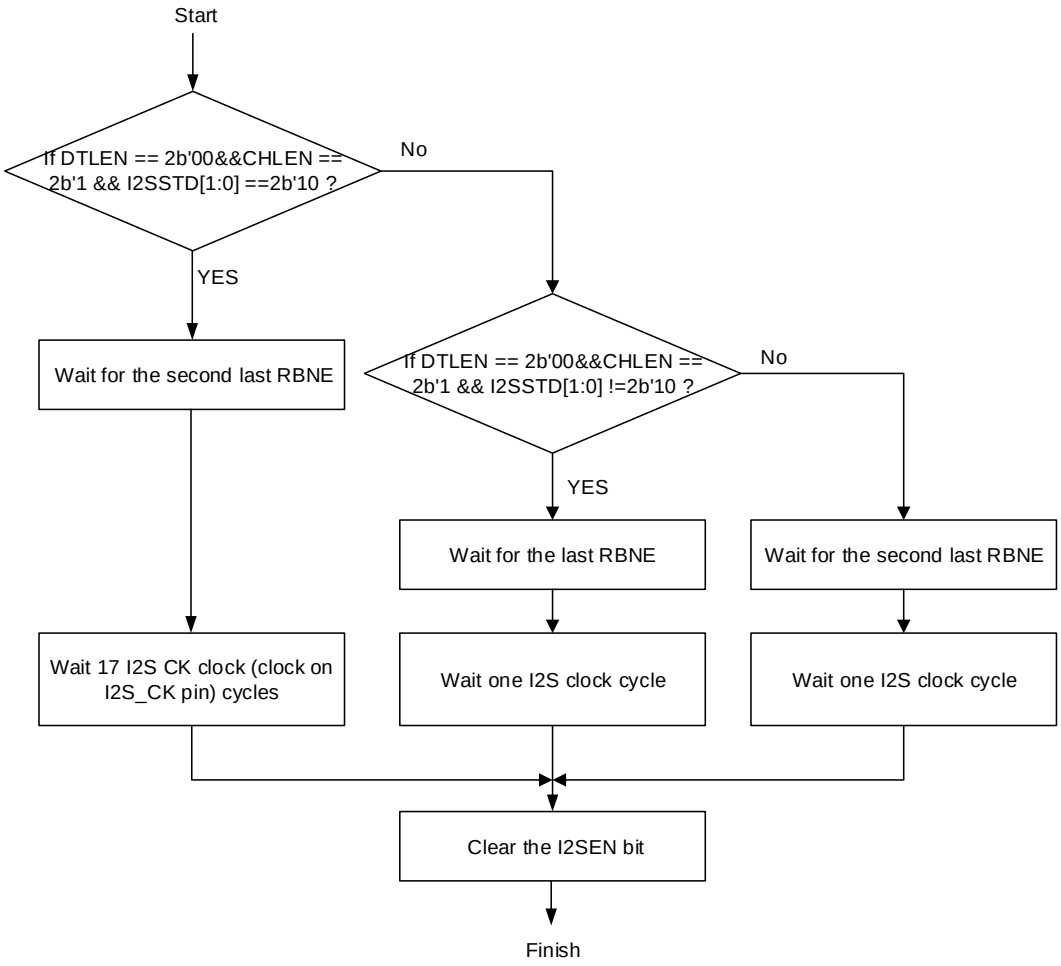
I2S master reception sequence

The RBNE flag is used to control the reception sequence. As is mentioned before, the RBNE flag indicates the receive buffer is not empty, and an interrupt will be generated if the RBNEIE bit in the SPI_CTL1 register is set. The reception sequence begins immediately when the I2SEN bit in the SPI_I2SCTL register is set. At the beginning, the receive buffer is empty (RBNE is low). When a reception sequence finishes, the received data in the shift register is loaded into the receive buffer (RBNE goes high). The data should be read from the SPI_DATA register, when the RBNE flag is high. After a read operation to the SPI_DATA register, the RBNE flag goes low. It is mandatory to read the SPI_DATA register before the end of the next reception. Otherwise, reception overrun error occurs. The RXORERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is necessary to switch off and then switch on I2S before resuming the communication.

For all standards except PCM, the I2SCH flag is used to distinguish the channel side to which the received data belongs. 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 24-53. I2S master reception disabling sequence***.

Figure 24-53. I2S master reception disabling sequence



I2S slave transmission sequence

The transmission sequence in slave mode is similar to that in master mode. The difference between them is 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 finishe. 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 switch off and switch on I2S to resume the communication. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to switch off 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 difference between them is 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 switch off I2S, it is mandatory to clear the I2SEN bit immediately after receiving the last RBNE.

24.4.6. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

24.4.7. I2S interrupts

Status flags

There are four status flags implemented in the SPI_STAT register, including TBE, RBNE, TRANS and I2SCH. The user can use them to fully monitor the state of the I2S bus.

- Transmit buffer empty flag (TBE)

This bit is set when the transmit buffer is empty, the software can write the next data to the transmit buffer by writing the SPI_DATA register.

- Receive buffer not empty flag (RBNE)

This bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer, and software can read the data by reading the SPI_DATA register.

- I2S Transmitting On-Going flag (TRANS)

TRANS is a status flag to indicate whether the transfer is on-going or not. It is set and cleared by internal hardware and not controlled by software. This flag doesn't generate any interrupt.

- I2S channel side flag (I2SCH)

This flag indicates the channel side information of the current transfer and has no meaning in PCM mode. It is updated when TBE rises in transmission mode or RBNE rises in reception mode. This flag doesn't generate any interrupt.

Error conditions

There are three error conditions:

- **Transmission Underrun Error Flag (TXURERR)**

In the slave transmit mode, when the valid SCK signal starts transmitting, if the transmit buffer is empty, TXURERR will be set.

- **Reception Overrun Error Flag (RXORERR)**

This condition occurs when the receive buffer is full and a newly incoming data has been completely received. When overrun occurs, the data in receive buffer is not updated and the newly incoming data is lost.

- **Format Error (FERR)**

In slave I2S mode, the I2S monitors the I2S_WS signal and an error flag will be set if I2S_WS toggles at an unexpected position.

I2S interrupt events and corresponding enabled bits are summed up in the ***Table 24-10. I2S interrupt***.

Table 24-10. I2S interrupt

Flag Name	Description	Clear Method	Interrupt Enable bit
TBE	Transmit buffer empty	Write SPI_DATA register	TBEIE
RBNE	Receive buffer not empty	Read SPI_DATA register	RBNEIE
TXURERR	Transmission underrun error	Read SPI_STAT register	ERRIE
RXORERR	Reception overrun error	Read SPI_DATA register and then read SPI_STAT register.	
FERR	I2S Format Error	Read SPI_STAT register	

24.5. Register definition

SPI0 base address: 0x4001 3000

SPI1/I2S1 base address: 0x4000 3800

SPI2/I2S2 base address: 0x4000 3C00

24.5.1. Control register 0 (SPI_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by 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	SWNSSEN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The information transfers between the MOSI pin in master and the MISO pin in slave.
14	BDOEN	Bidirectional transmit output enable When BDEN is set, this bit determines the direction of transfer. 0: Work in receive-only mode 1: Work in transmit-only mode
13	CRCEN	CRC calculation enable 0: CRC calculation is disabled 1: CRC calculation is enabled.
12	CRCNT	CRC next transfer 0: Next transfer is Data 1: Next transfer is CRC value (TCR) 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 1: Receive-only
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. This bit has no meaning in SPI TI mode.
8	SWNSS	NSS pin selection in NSS software mode 0: NSS pin is pulled low 1: NSS pin is pulled high This bit has an effect only when the SWNSSEN bit is set. This bit has no meaning in SPI TI mode.
7	LF	LSB first mode 0: Transmit MSB first 1: Transmit LSB first This bit has no meaning in SPI TI mode.
6	SPIEN	SPI enable 0: SPI peripheral is disabled 1: SPI peripheral is enabled
5:3	PSC[2:0]	Master clock prescaler selection 000: PCLK/2 100: PCLK/32 001: PCLK/4 101: PCLK/64 010: PCLK/8 110: PCLK/128 011: PCLK/16 111: PCLK/256 PCLK means PCLK2 when using SPI0 or PCLK1 when using SPI1 and SPI2.
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

24.5.2. Control register 1 (SPI_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								TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TBEIE	Transmit buffer empty interrupt enable 0: TBE interrupt is disabled 1: TBE interrupt is enabled. An interrupt is generated when the TBE bit is set.
6	RBNEIE	Receive buffer not empty interrupt enable 0: RBNE interrupt is disabled 1: RBNE interrupt is enabled. An interrupt is generated when the RBNE bit is set.
5	ERRIE	Errors interrupt enable. 0: Error interrupt is disabled. 1: Error interrupt is enabled. An interrupt is generated when the CRCERR bit or the CONFERR bit or the RXORERR bit or the TXURERR bit is set.
4	TMOD	SPI TI mode enable 0: SPI TI Mode Disabled 1: SPI TI Mode Enabled
3	NSSP	SPI NSS pulse mode enable 0: SPI NSS Pulse Mode Disable 1: SPI NSS Pulse Mode Enable
2	NSSDRV	Drive NSS output 0: NSS output is disabled 1: NSS output is enabled If the NSS pin is configured as output, the NSS pin is pulled low in master mode when SPI is enabled. If the NSS pin is configured as input, the NSS pin should be pulled high in master mode, and this bit has no effect.

1	DMATEN	Transmit buffer DMA enable 0: Transmit buffer DMA is disabled 1: Transmit buffer DMA is enabled. When the TBE bit in SPI_STAT is set, it will generate a DMA request at corresponding DMA channel.
0	DMAREN	Receive buffer DMA enable 0: Receive buffer DMA is disabled 1: Receive buffer DMA is enabled. When the RBNE bit in SPI_STAT is set, it will generate a DMA request at corresponding DMA channel.

24.5.3. Status register (SPI_STAT)

Address offset: 0x08
Reset value: 0x0002

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							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 bit SPI TI Mode: 0: No TI Mode format error 1: TI Mode format error occurs. I2S Mode: 0: No I2S format error 1: I2S format error occurs. This bit is set by hardware and is able to be cleared by writing 0.
7	TRANS	Transmitting on-going 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 bit 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

24.5.4. Data register (SPI_DATA)

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
SPI_DATA[15:0]															
rw															

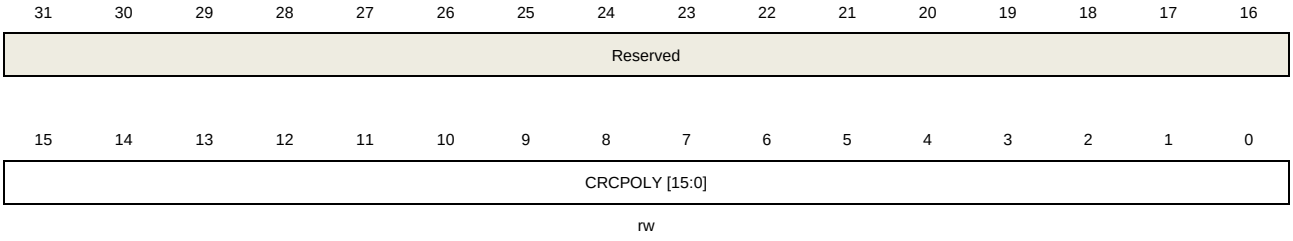
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	Data transfer register The hardware has two buffers, including transmit buffer and receive buffer. Write data to SPI_DATA will save the data to transmit buffer and read data from SPI_DATA will get the data from receive buffer. When the data frame format is set to 8-bit data, the SPI_DATA [15:8] is forced to 0 and the SPI_DATA [7:0] is used for transmission and reception, transmit buffer and receive buffer are 8-bits. If the Data frame format is set to 16-bit data, the SPI_DATA [15:0] is used for transmission and reception, transmit buffer and receive buffer are 16-bit.

24.5.5. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

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



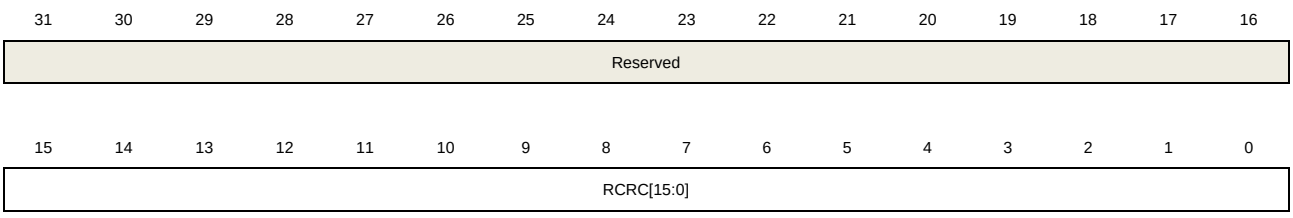
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRCPOLY[15:0]	CRC polynomial register This register contains the CRC polynomial and it is used for CRC calculation. The default value is 0007h.

24.5.6. RX CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000 0000

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



r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCRC[15:0]	RX CRC register When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the received bytes and saves them in RCRC register. If the Data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in RCRC [7:0]. When the Data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in RCRC[15:0]. The hardware computes the CRC value after each received bit, when the TRANS is set, a read to this register could return an intermediate value. This register is reset when the CRCEN bit or the SPIEN bit in SPI_CTL0 register is cleared.

24.5.7. TX CRC register (SPI_TCRC)

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															

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

r

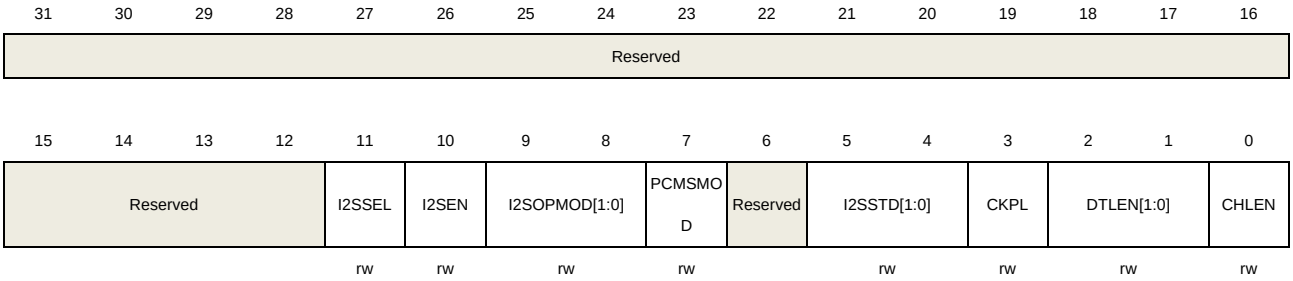
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	TX CRC register When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the transmitted bytes and saves them in TCRC register. If the Data frame format is set to 8-bit data, CRC calculation is based 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 or the SPIEN bit in SPI_CTL0 register is cleared.

24.5.8. I2S control register (SPI_I2SCTL)

Address offset: 0x1C

Reset value: 0x0000 0000

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



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: I2S is disabled 1: I2S is enabled This bit is not used in SPI mode.
9:8	I2SOPMOD[1:0]	I2S operation mode 00: Slave transmission mode 01: Slave reception mode 10: Master transmission mode 11: Master reception mode This bit should be configured when I2S 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

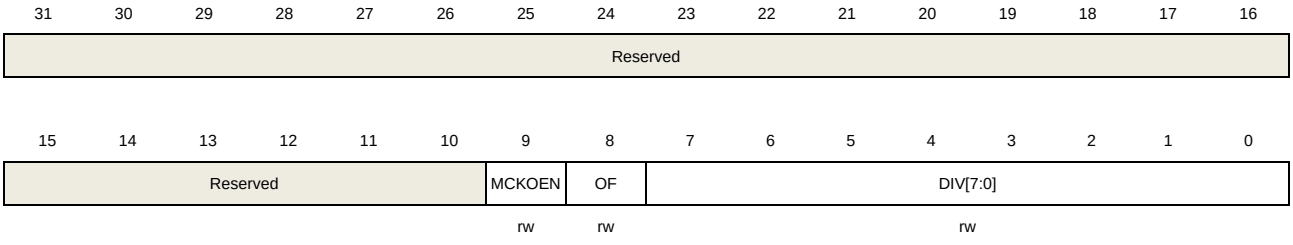
		11: PCM standard These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
3	CKPL	Idle state clock polarity 0: The idle state of I2S_CK is low level 1: The idle state of I2S_CK is high level This bit should be configured when I2S mode is disabled. This bit is not 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.

24.5.9. I2S clock prescaler register (SPI_I2SPSC)

Address offset: 0x20

Reset value: 0x0000 0002

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	MCKOEN	I2S_MCK output enable 0: I2S_MCK output is disabled 1: I2S_MCK output is enabled This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.

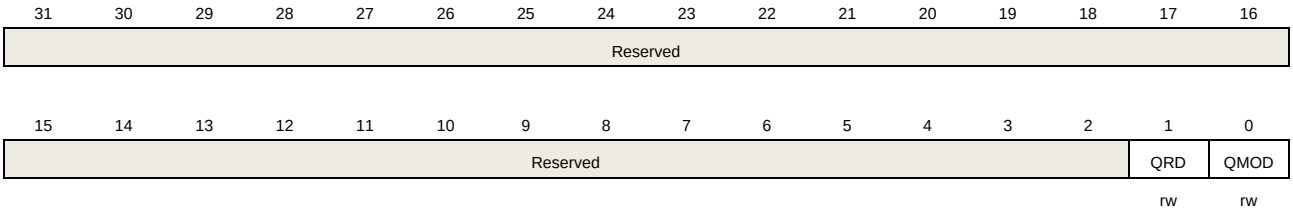
8	OF	Odd factor for the prescaler 0: Real divider value is DIV * 2 1: Real divider value is DIV * 2 + 1 This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
7:0	DIV[7:0]	Dividing factor for the prescaler Real divider value is DIV * 2 + OF. DIV must not be 0. These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.

24.5.10. Quad-SPI mode control register (SPI_QCTL) of SPI0

Address offset: 0x80

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	QRD	Quad-SPI mode read select. 0: SPI is in quad wire write mode 1: SPI is in quad wire read mode This bit should be only be configured when SPI is not busy (TRANS bit cleared) This bit is only available in SPI0.
0	QMOD	Quad-SPI mode enable 0: SPI is in single wire mode 1: SPI is in Quad-SPI mode This bit should only be configured when SPI is not busy (TRANS bit cleared). This bit is only available in SPI0.

25. Inter-integrated circuit interface (I2C)

25.1. Overview

The I2C (inter-integrated circuit) module provides an I2C interface which is an industry standard two-line serial interface for MCU to communicate with external I2C interface. I2C bus uses two serial lines: a serial data line (SDA) and a serial clock line (SCL).

The I2C interface implements standard I2C protocol with standard mode, fast mode and fast mode plus as well as CRC calculation and checking, SMBus (system management bus), and PMBus (power management bus). It also supports multi-master I2C bus. The I2C interface provides DMA mode for users to reduce CPU overload.

25.2. Characteristics

- Parallel-bus to I2C-bus protocol converter and interface.
- Both master and slave functions with the same interface.
- Bi-directional data transfer between master and slave.
- Supports 7-bit and 10-bit addressing and general call addressing.
- Multiple 7-bit slave addresses (2 address, 2 with configurable mask).
- Programmable setup time and hold time.
- Multi-master capability.
- Supports standard mode (up to 100 kHz) and fast mode (up to 400 kHz) and fast mode plus (up to 1MHz, must enable Fm+ in SYSCFG_CFG1 register).
- Configurable SCL stretching in slave mode.
- Supports DMA mode.
- SMBus 3.0 and PMBus 1.3 compatible.
- Optional PEC (packet error checking) generation and check.
- Programmable analog and digital noise filters.

25.3. Function overview

Figure 25-1. I2C module block diagram below provides details on the internal configuration of the I2C interface.

Figure 25-1. I2C module block diagram

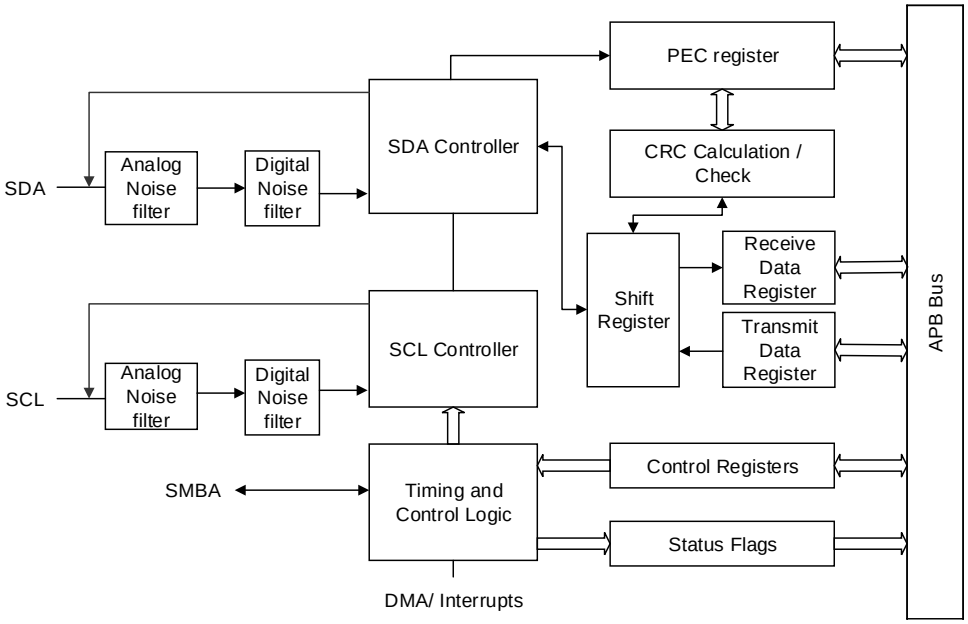


Table 25-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

Term	Description
Transmitter	the device which sends data to the bus
Receiver	the device which receives data from the bus
Master	the device which initiates a transfer, generates clock signals and terminates a transfer
Slave	the device addressed by a master
Multi-master	more than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	procedure to ensure that, if more than one master tries to control the bus simultaneously, only one is allowed to do so and the winning master's message is not corrupted

25.3.1. Clock requirements

The I2CCLK period t_{I2CCLK} must match the conditions as follows:

- $t_{I2CCLK} < (t_{LOW} - t_{filters})/4$
- $t_{I2CCLK} < t_{HIGH}$

with:

t_{LOW}: SCL low time

t_{HIGH}: SCL high time

$t_{filters}$: When the filters are enabled, represent the delays by the analog filter and digital filter.

Analog filter delay is maximum 126ns. Digital filter delay is DNF[3:0]×t_{I2CCLK}.

The period of PCLK clock t_{PCLK} match the conditions as follows:

- t_{PCLK}<4/3*t_{SCL}

with:

t_{SCL}: the period of SCL

Note: When the I2C kernel is provided by PCLK, this clock must match the conditions for t_{I2CCLK}.

25.3.2. I2C communication flow

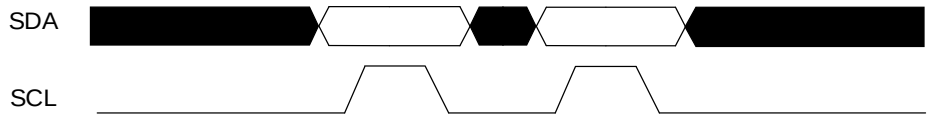
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

Data validation

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (see ***Figure 25-2. Data validation***). One clock pulse is generated for each data bit transferred.

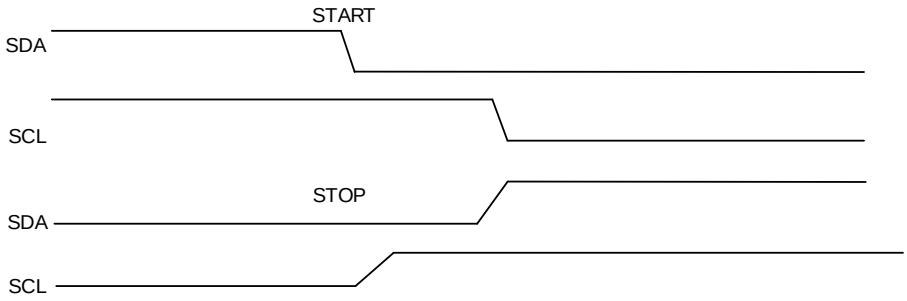
Figure 25-2. Data validation



START and STOP signal

All transactions begin with a START and are terminated by a STOP (see ***Figure 25-3. 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 25-3. START and STOP signal



Each I2C device is recognized by a unique address (whether it is a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. It operates in slave mode by default. When it generates a START signal, the interface automatically switches from slave to master. If an arbitration loss or a STOP generation occurs, then the interface switches from master to slave, allowing multimaster capability.

An I2C slave will continue to detect addresses after a START signal on I2C bus and compare the detected address with its slave address which is programmable by software. Once the two addresses match, the I2C slave will send an ACK to the I2C bus and responses to the following command on I2C bus: transmitting or receiving the desired data. Additionally, if General Call is enabled by software, the I2C slave always responses to a General Call Address (0x00). The I2C block support both 7-bit and 10-bit address modes.

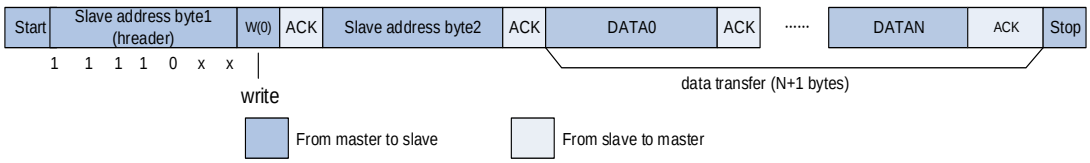
Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the START signal contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

A 9th clock pulse follows the 8 clock cycles of byte transmission, during which the receiver must send an acknowledge bit to the transmitter. Acknowledge can be enabled or disabled by software.

An I2C master always initiates or end a transfer using START or STOP signal and it's also responsible for SCL clock generation.

In master mode, if AUTOEND=1, the STOP signal is generated automatically by hardware. If AUTOEND=0, the STOP signal generated by software, or the master can generate a RESTART signal to start a new transfer.

Figure 25-4. I2C communication flow with 10-bit address (Master Transmit)



Note: In the 10-bit address master transmission scenario, after a data packet is sent, the master must first send a stop signal and then send a start signal to initiate the transmission of a new data packet.

Figure 25-5. I2C communication flow with 7-bit address (Master Transmit)

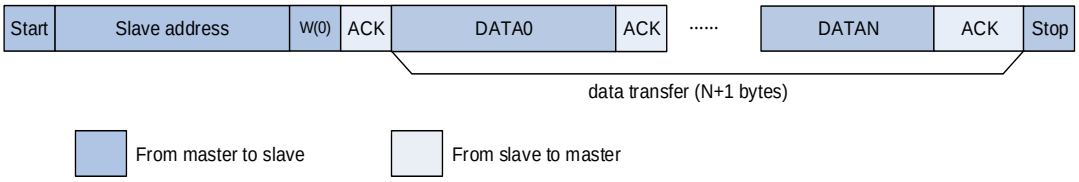
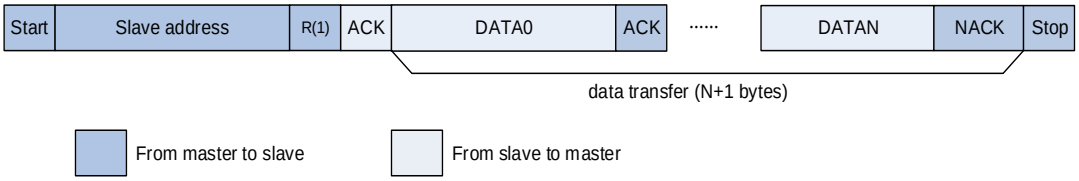


Figure 25-6. I2C communication flow with 7-bit address (Master Receive)



In 10-bit addressing mode, the HEAD10R bit can configured to decide whether the complete address sequence must be executed, or only the header to be sent. When HEAD10R=0, the complete 10 bit address read sequence must be executed with START + header of 10-bit address in write direction + slave address byte 2 + RESTART + header of 10-bit address in read direction, as is shown in ***Figure 25-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)***.

In 10-bit addressing mode, when HEAD10R=1, the I2C transmission sequence is limited to a master transmit immediately followed by a master receive. The sequence is shown in ***Figure 25-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)***.

Figure 25-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

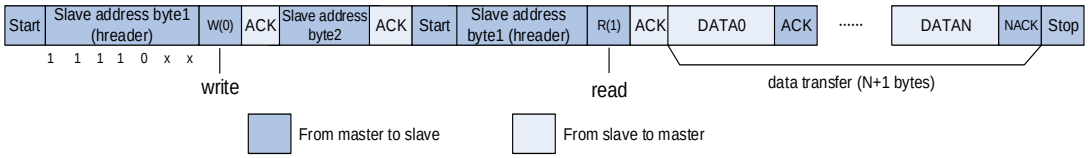
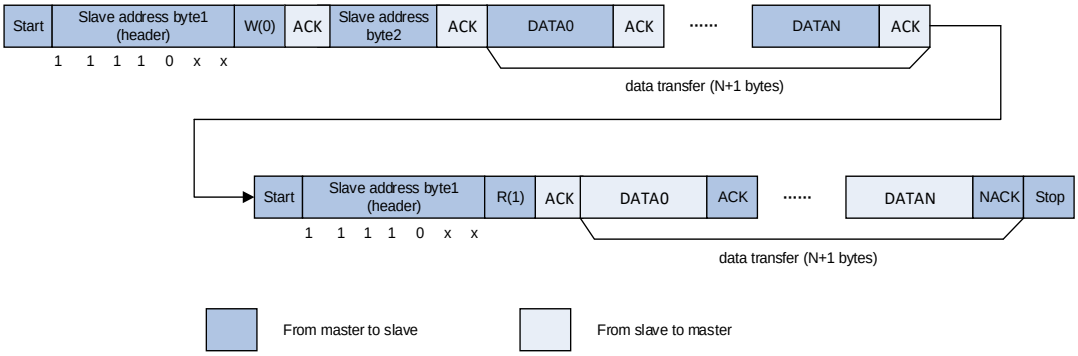


Figure 25-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



25.3.3. Noise filter

Analog noise filter and digital noise filter are integrated in I2C peripherals, the noise filters can

be configured before the I2C peripheral is enabled according to the actual requirements.

The analog noise filter is disabled by setting the ANOFF bit in I2C_CTL0 register and enabled when ANOFF is 0. It can suppress noise with a pulse width of less than 50ns in fast mode and fast mode plus.

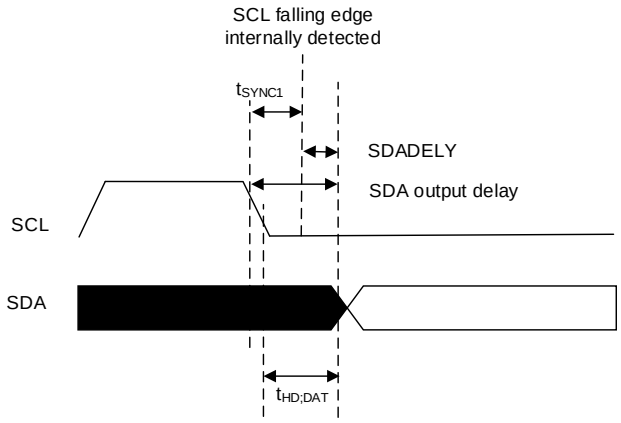
The digital noise filter can be used by configuring the DNF [3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will be changed if the level is stable for more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF [3:0].

25.3.4. I2C timings configuration

The PSC [3:0], SCLDELY [3:0] and SDADELY [3:0] bits in the I2C_TIMING register must be configured in order to guarantee a correct data hold and setup time used in I2C communication.

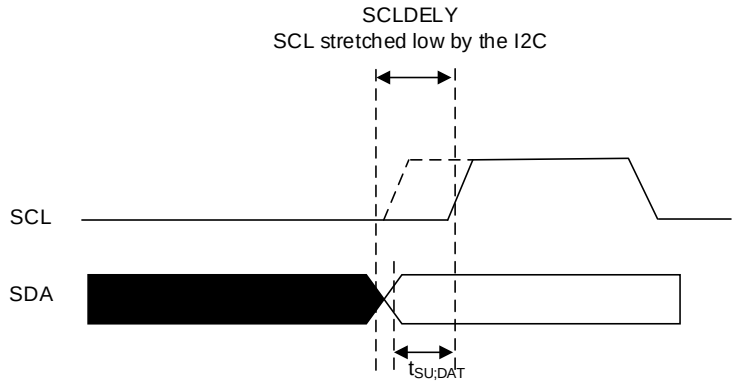
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 25-9. Data hold time***.

Figure 25-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in ***Figure 25-10. Data setup time***.

Figure 25-10. Data setup time



When the SCL falling edge is internally detected, a delay is inserted before sending SDA output. This delay is $t_{SDADELY}$ = SDADELY * t_{PSC} + t_{I2CCLK} where t_{PSC} = (PSC+1) * t_{I2CCLK}. $t_{SDADELY}$ effects $t_{HD;DAT}$. The total delay of SDA output is t_{SYNC1} + {[SDADELY * (PSC+1) + 1] * t_{I2CCLK}}. t_{SYNC1} depends on SCL falling slope, the delay of analog filter, the delay of digital filter and delay of SCL synchronization to I2CCLK clock. The delay of SCL synchronization to I2CCLK clock is 2 to 3 t_{I2CCLK}.

Note: When SDADELY = 0, $t_{SDADELY}$ = t_{PSC}.

SDADELY must match condition as follows:

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

Note: t_{AF} is the delay of analog filter. The $t_{HD;DAT}$ should be less than the maximum of $t_{VD;DAT}$.

When SS = 0, after $t_{SDADELY}$ delay, the slave had to stretch the clock before the data writing to I2C_TDATA register, SCL is low during the data setup time. The setup time is $t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}. $t_{SCLDELY}$ effects $t_{SU;DAT}$.

SCLDELY must match condition as follows:

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

In master mode, the SCL clock high and low levels must be configured by programming the PSC [3:0], SCLH [7:0] and SCLL[7:0] bits in the I2C_TIMING register.

When the SCL falling edge is internally detected, a delay is inserted before releasing the SCL output. This delay is t_{SCLL}=(SCLL+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLL} impacts the SCL low time t_{LOW}.

When the SCL rising edge is internally detected, a delay is inserted before forcing the SCL output to low level. This delay is t_{SCLH}=(SCLH+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLH} impacts the SCL high time t_{HIGH}.

Note: When the I2C is enabled, the timing configuration and SS mode must not be changed.

Table 25-2. Data setup time and data hold time

Symbol	Parameter	Standard mode		Fast mode		Fast mode plus		SMBus		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{HD;DAT}$	Data hold time	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	Data valid time	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	Data setup time	250	-	100	-	50	-	250	-	ns
t_r	Rising time of SCL and SDA	-	1000	-	300	-	120	-	1000	
t_f	falling time of SCL and SDA	-	300	-	300	-	120	-	300	

25.3.5. I2C reset

A software reset can be performed by clearing the I2CEN bit in the I2C_CTL0 register. When a software reset is generated, the SCL and SDA are released. The communication control bits and status bits come back to the reset value. Software reset have no effect on configuration registers. The impacted register bits are START, STOP, NACKEN in I2C_CTL1 register, I2CBSY, TBE, TI, RBNE, ADDSEND, NACK, TCR, TC, STPDET, BERR, LOSTARB and OUERR in I2C_STAT register. Additionally, when the SMBus is supported, PECTRANS in I2C_CTL1 register, PECERR, TIMEOUT and SMBALT in I2C_STAT are also impacted.

In order to perform the software reset, I2CEN must be kept low during at least 3 APB clock cycles. This is ensured by writing software sequence as follows:

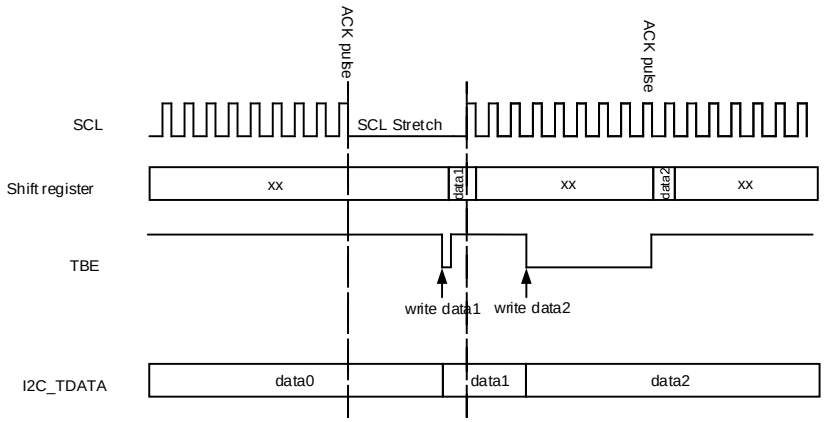
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

25.3.6. Data transfer

Data Transmission

When transmitting data, if TBE is 0, it indicates that the I2C_TDATA register is not empty, the data in I2C_TDATA register is moved to the shift register after the 9th SCL pulse. Then the data will be transmitted through the SDA line from the shift register. If TBE is 1, it indicates that the I2C_TDATA register is empty, the SCL line is stretched low until I2C_TDATA is not empty. The stretch begins after the 9th SCL pulse.

Figure 25-11. Data transmission

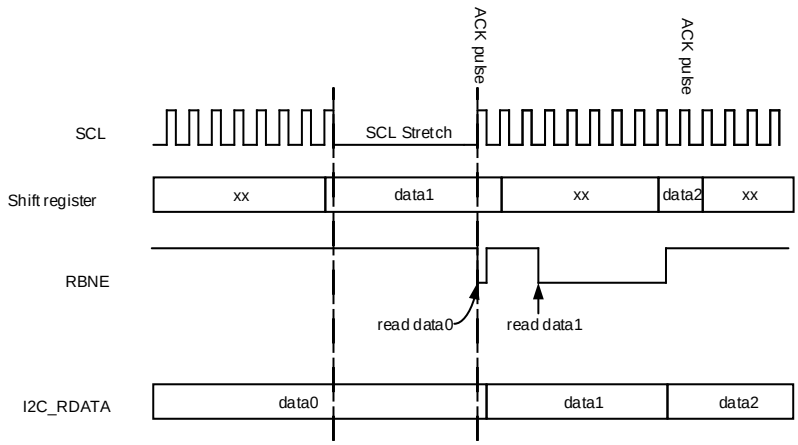


Data Reception

When receiving data, the data will be received in the shift register first. If RBNE is 0, the data in the shift register will move into I2C_RDATA register. If RBNE is 1, the SCL line will be stretched until the previous received data in I2C_RDATA is read. The stretch is inserted before

the acknowledge pulse.

Figure 25-12. Data reception



Reload and automatic end mode

In order to manage byte transfer and to shut down the communication in modes as is shown in ***Table 25-3. Communication modes to be shut down***, the I2C embedded a byte counter in the hardware.

Table 25-3. Communication modes to be shut down

Working mode	Action
Master mode	NACK, STOP and RESTART generation
Slave receiver mode	ACK control
SMBus mode	PEC generation/checking

The number of bytes to be transferred is configured by BYTENUM [7:0] in I2C_CTL1 register. If BYTENUM is greater than 255, or in slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In reload mode, when BYTENUM counts to 0, the TCR bit will be set, and an interrupt will be generated if TCIE is set. Once the TCR flag is set, SCL is stretched. The TCR bit is cleared by writing a non-zero number in BYTENUM.

Note: The reload mode must be disabled after the last reloading of BYTENUM [7:0].

The reload mode must be disabled when the automatic end mode is enabled. In automatic end mode, the master will send a STOP signal automatically when the BYTENUM [7:0] counts to 0.

When reload mode and automatic end mode are disabled, the I2C communication process needs to be terminated by software. If the number of bytes BYTENUM [7:0] has been transferred, the STOP bit should be set by software to generate a STOP signal, and then TC flag must be cleared.

25.3.7. I2C slave mode

Initialization

When works in slave mode, at least one slave address should be enabled. Slave address 1 can be programmed in I2C_SADDR0 register and slave address 2 can be programmed in I2C_SADDR1 register. ADDRESSEN in I2C_SADDR0 register and ADDRESS2EN in I2C_SADDR1 register should be set when the corresponding address is used. 7-bit address or 10-bit address can be programmed in ADDRESS [9:0] in I2C_SADDR0 register by configuring the ADDFORMAT bit in 7-bit address or 10-bit address.

The ADDM[6:0] in I2C_CTL2 register defines which bits of ADDRESS [7:1] are compared with an incoming address byte, and which bits are ignored.

The ADDMSK2[2:0] is used to mask ADDRESS2[7:1] in I2C_SADDR1 register. For details, refer to the description of ADDMSK2[2:0] in I2C_SADDR1 register.

When the I2C received address matches one of its enabled addresses, the ADDSEND will be set, and an interrupt is generated if the ADDMIE bit is set. The READDR[6:0] bits in I2C_STAT register will store the received address. And TR bit in I2C_STAT register updates after the ADDSEND is set. The bit will let the slave to know whether to act as a transmitter or receiver.

SCL line stretching

The clock stretching is used in slave mode by default (SS=0), the SCL line can be stretched low if necessary. The SCL will be stretched in following cases.

- The SCL is stretched when the ADDSEND bit is set, and released when the ADDSEND bit is cleared.
- In slave transmitting mode, after the ADDSEND bit is cleared, the SCL will be stretched before the first data byte writing to the I2C_TDATA register. Or the SCL will be stretched before the new data is written to the I2C_TDATA register after the previous data transmission is completed.
- In slave receiving mode, a new reception is completed but the data in I2C_RDATA register has not been read.
- When SBCTL=1 and RELOAD=1, after the transfer of the last byte, TCR is set. Before the TCR is cleared, the SCL will be stretched.
- The I2C stretches SCL low during $[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}$ after detecting the SCL falling edge.

The clock stretching can be disabled by setting the SS bit in I2C_CTL0 register (SS=1). The SCL will not be stretched in following cases.

- When the ADDSEND is set, the SCL will be not stretched.
- In slave transmitting mode, before the first SCL pulse, the data must be written in the I2C_TDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, an interrupt will be generated. When the STPDET bit is set and the first

data transmission starts, OUERR bit in the I2C_STAT register will also be set.

- In slave receiving mode, before the 9th SCL pulse (ACK pulse) occurred by the next data byte, the data must be read out from the I2C_RDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, and an interrupt will be generated.

Slave byte control mode

In slave receiving mode, the slave byte control mode can be enabled by setting the SBCTL bit in the I2C_CTL0 register to allow byte ACK control. When SS=1, the slave byte control mode is not allowed.

When using slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In slave byte control mode, BYTENUM [7:0] in I2C_CTL1 register must be configured as 1 in the ADDSEND interrupt service routine and reloaded to 1 after each byte received. The TCR bit in I2C_STAT register will be set when a byte is received, the SCL will be stretched low by slave between the 8th and 9th clock pulses. Then the data can be read from the I2C_RDATA register, and the slave determines to send an ACK or a NACK by configuring the NACKEN bit in the I2C_CTL1 register. When the BYTENUM [7:0] is written a non-zero value, the slave will release the stretch.

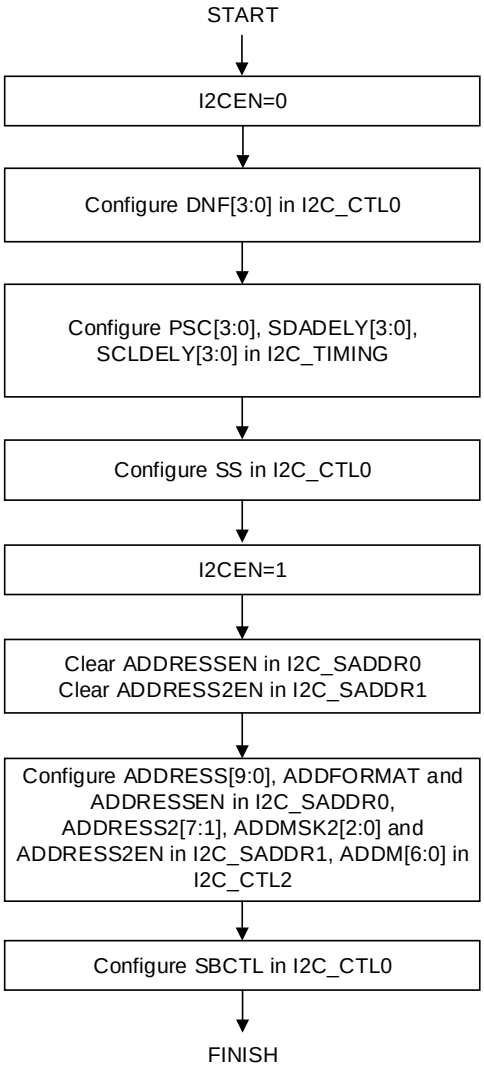
When the BYTENUM [7:0] is greater than 0x1, there is no stretch between the reception of two data bytes.

Note: The SBCTL bit can be configured in following cases:

- I2CEN=0.
- The slave has not been addressed.
- ADDSEND=1.

Only when the ADDSEND=1, or TCR=1, the RELOAD bit can be modified.

Figure 25-13. I2C initialization in slave mode



Programming model in slave transmitting mode

When the I2C_TDATA register is empty, the TI bit in I2C_STAT register will be set. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The NACK bit in I2C_STAT register will be set when a NACK is received. And an interrupt is generated if the NACKIE bit is set in the I2C_CTL0 register. The TI bit in I2C_STAT register will not be set when a NACK is received.

The STPDET bit in I2C_STAT register will be set when a STOP is received. If the STPDETIE in I2C_CTL0 register is set, an interrupt will be generated.

When SBCTL is 0, if ADDSEND=1, and the TBE bit in I2C_STAT register is 0, the data in I2C_TDATA register can be chosen to be transmitted or flushed. The data is flushed by setting the TBE bit.

When SBCTL=1, the slave works in slave byte control mode, the BYTENUM [7:0] must be configured in the ADDSEND interrupt service routine. And the number of TI events is equal

to the value of BYTENUM [7:0].

When SS=1, the SCL will not be stretched when ADDSEND bit in I2C_STAT register is set. In this case, the data in I2C_TDATA register cannot be flushed in ADDSEND interrupt service routine. So the first byte to be sent must be programmed in the I2C_TDATA register previously.

- This data can be the one which is written in the last TI event of the last transfer.
- Setting the TBE bit can flush the data if it is not the one to be sent, then a new byte can be written in I2C_TDATA register. The STPDET must be 0 when the data transmission begins. Or else the OUERR bit in I2C_STAT register will be set and an underrun error occurs.
- When interrupt or DMA is used in slave transmitter, if a TI event is needed, in order to generate a TI event both the TI bit and the TBE bit must be set.

Figure 25-14. Programming model for slave transmitting when SS=0

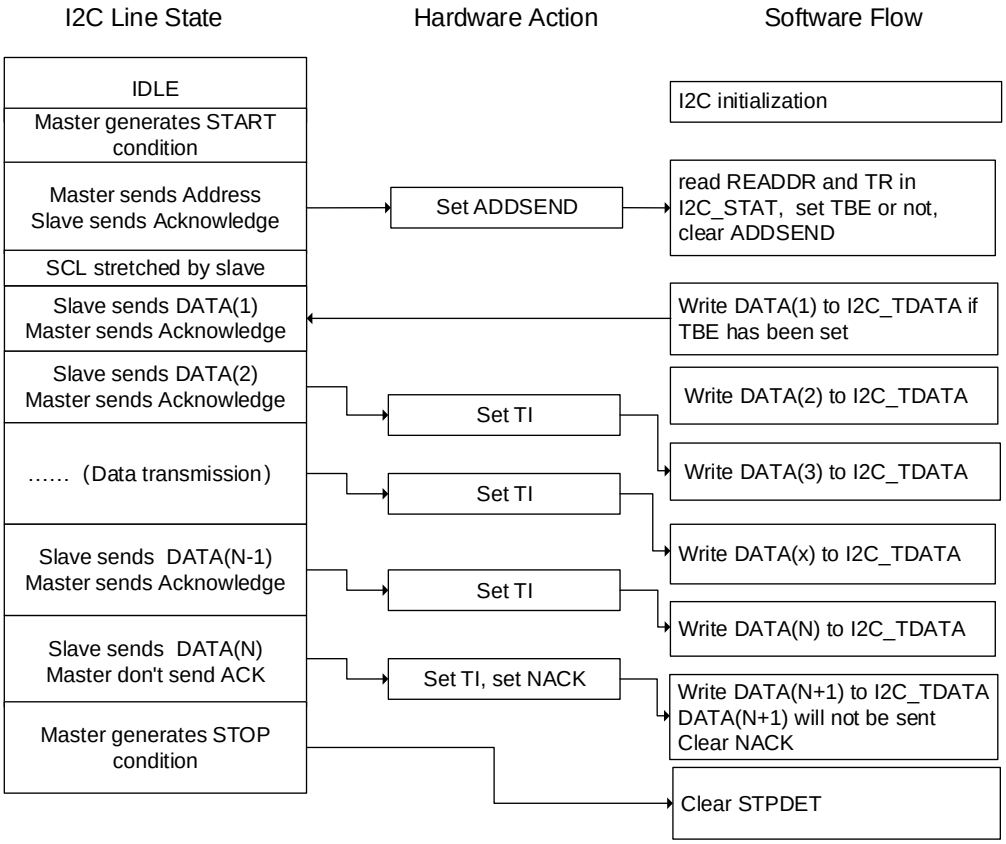
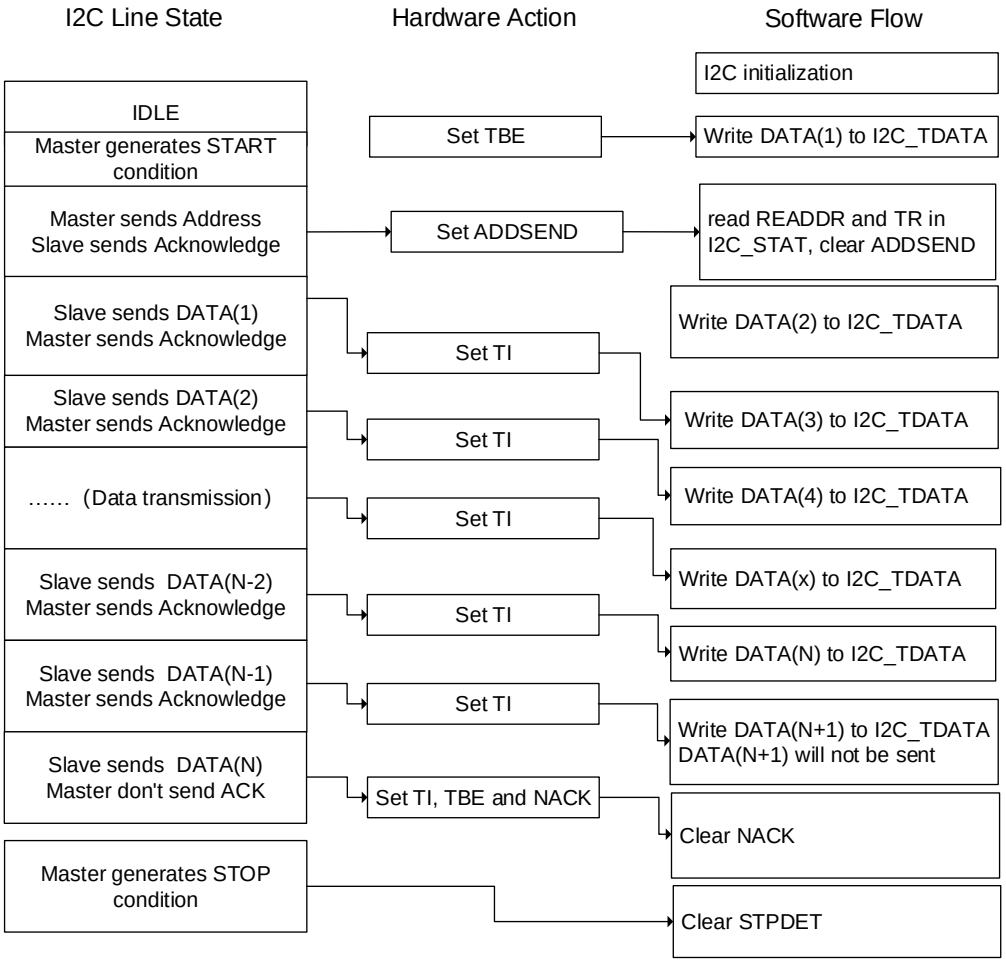


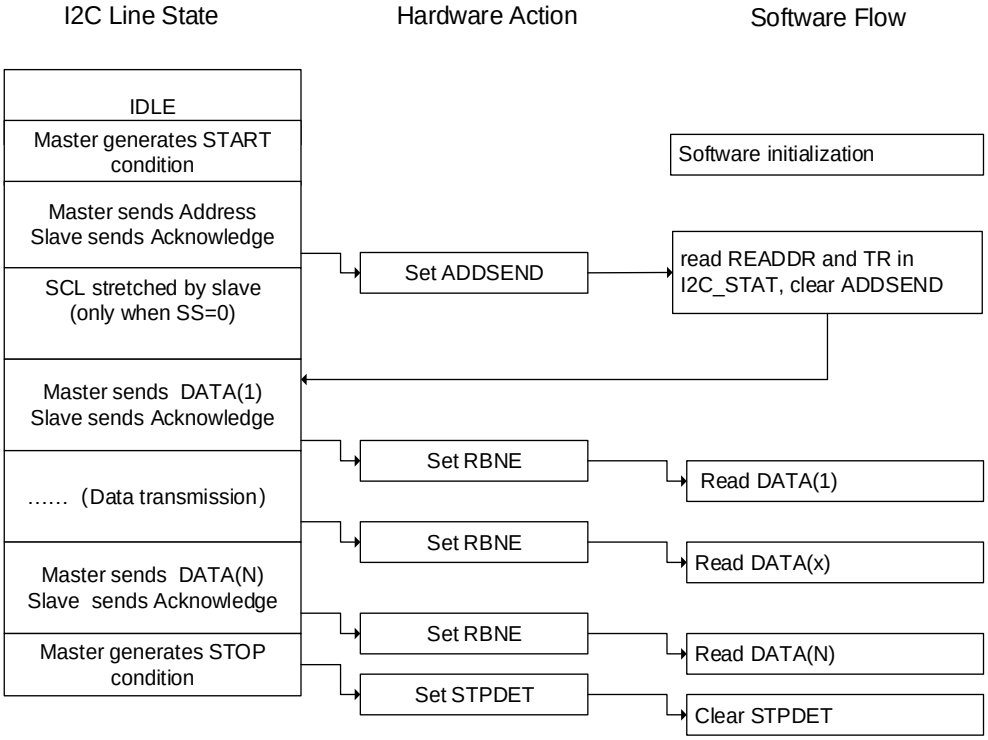
Figure 25-15. Programming model for slave transmitting when SS=1



Programming model in slave receiving mode

When the I2C_RDATA is not empty, the RBNE bit in I2C_STAT register is set, and if the RBNEIE bit in I2C_CTL0 register is set, an interrupt will be generated. When a STOP is received, STPDET will be set in I2C_STAT register. If the STPDETIE bit in I2C_CTL0 register is set, and an interrupt will be generated.

Figure 25-16. Programming model for slave receiving



25.3.8. I2C master mode

Initialization

The SCLH [7:0] and SCLL [7:0] in I2C_TIMING register should be configured when I2CEN is 0. In order to support multi-master communication and slave clock stretching, a clock synchronization mechanism is implemented.

The SCLL [7:0] and SCLH [7:0] are used for the low level counting and high level counting respectively. After a t_{SYNC1} delay, when the SCL low level is detected, the SCLL[7:0] starts counting, if the SCLL[7:0] in I2C_TIMING register is reached by SCLL[7:0] counter, the I2C will release the SCL clock. After a t_{SYNC2} delay, when the SCL high level is detected, the SCLH[7:0] starts counting, if the SCLH[7:0] in I2C_TIMING register is reached by SCLH[7:0] counter, the I2C will stretch the SCL clock.

So the master clock period is:

$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$.

The t_{SYNC1} depends on the SCL falling slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The delay by digital noise filter is DNF[3:0]*t_{I2CCLK}.

When works in master mode, the ADD10EN bit, SADDRESS [9:0] bits, TRDIR bit should be configured in I2C_CTL1 register. When the addressing mode is 10-bit in master receiving

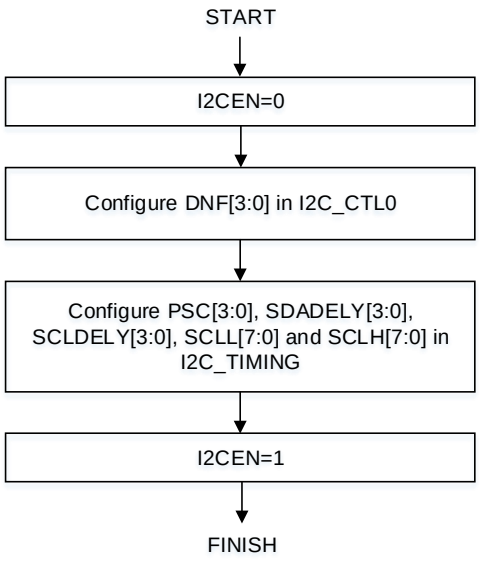
mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to be sent. The number of bytes to be transferred should be configured in BYTENUM [7:0] in I2C_CTL1 register. If the number of bytes to be transferred is equal to or greater than 255, BYTENUM [7:0] should be configured as 0xFF. Then the master sends the START signal. All the bits above should be configured before the START is set. The slave address will be sent after the START signal when the I2CBSY bit in I2C_STAT register is detected as 0. When the arbitration is lost, the master changes to slave mode and the START bit will be cleared by hardware. When the slave address has been sent, the START bit will be cleared by hardware.

Note: BYTENUM[7:0] can not be configured as 0.

In 10-bit addressing mode, if the master receives a NACK after the transmission of 10-bit header, the master will resend it until ACK is received. The ADDSENDC bit must be set to stop sending the slave address.

If the START bit is set, meanwhile the ADDSEND is set by addressing as a slave, the master changes to slave mode. The ADDSENDC bit must be set to clear the START bit.

Figure 25-17. I2C initialization in master mode



Programming model in master transmitting mode

In master transmitting mode, the TI bit is set after the ACK is received of each byte transmission. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The bytes to be transferred is programmed in BYTENUM [7:0] in I2C_CTL1 register. If the bytes to be transferred is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM [7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches until BYTENUM [7:0] is modified with a non-zero value.

When a NACK is received, the TI bit will not set.

- If data of BYTENUM [7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.
- If a NACK is received, a STOP signal is automatically generated, the NACK is set in I2C_STAT register, if the NACKIE bit is set, an interrupt will be generated.

Note: When the RELOAD bit is 1, the AUTOEND has no effect.

Figure 25-18. Programming model for master transmitting (N<=255)

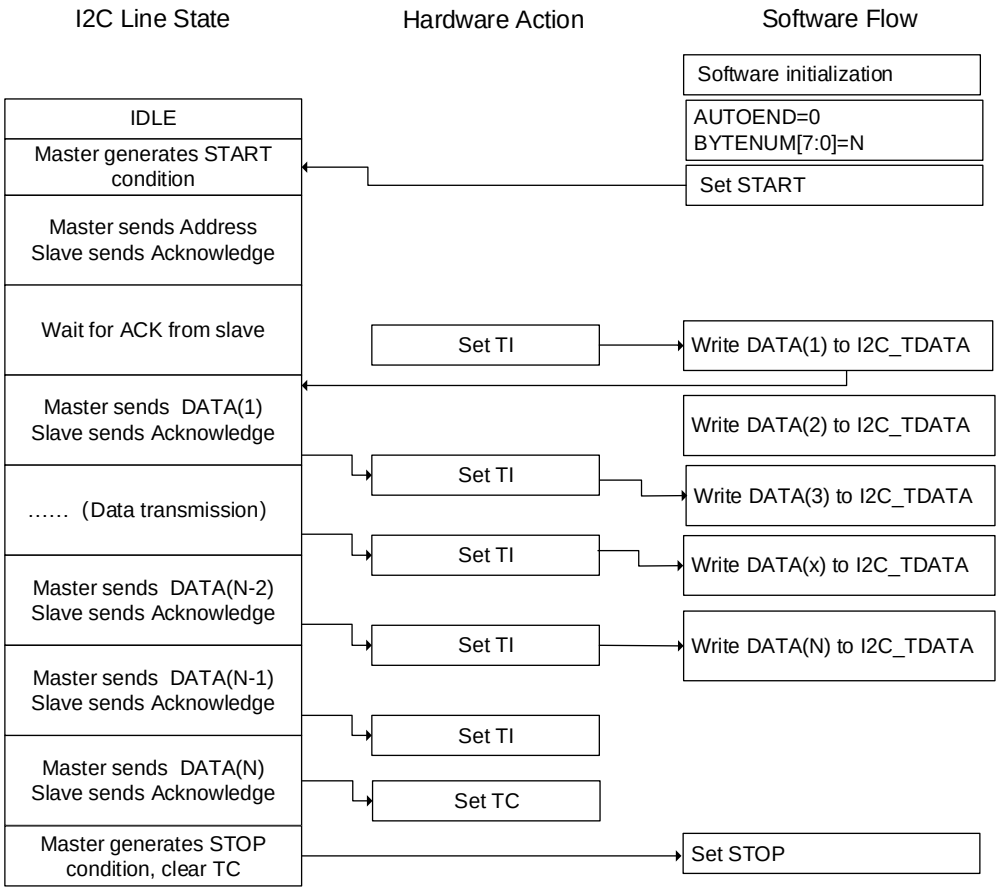
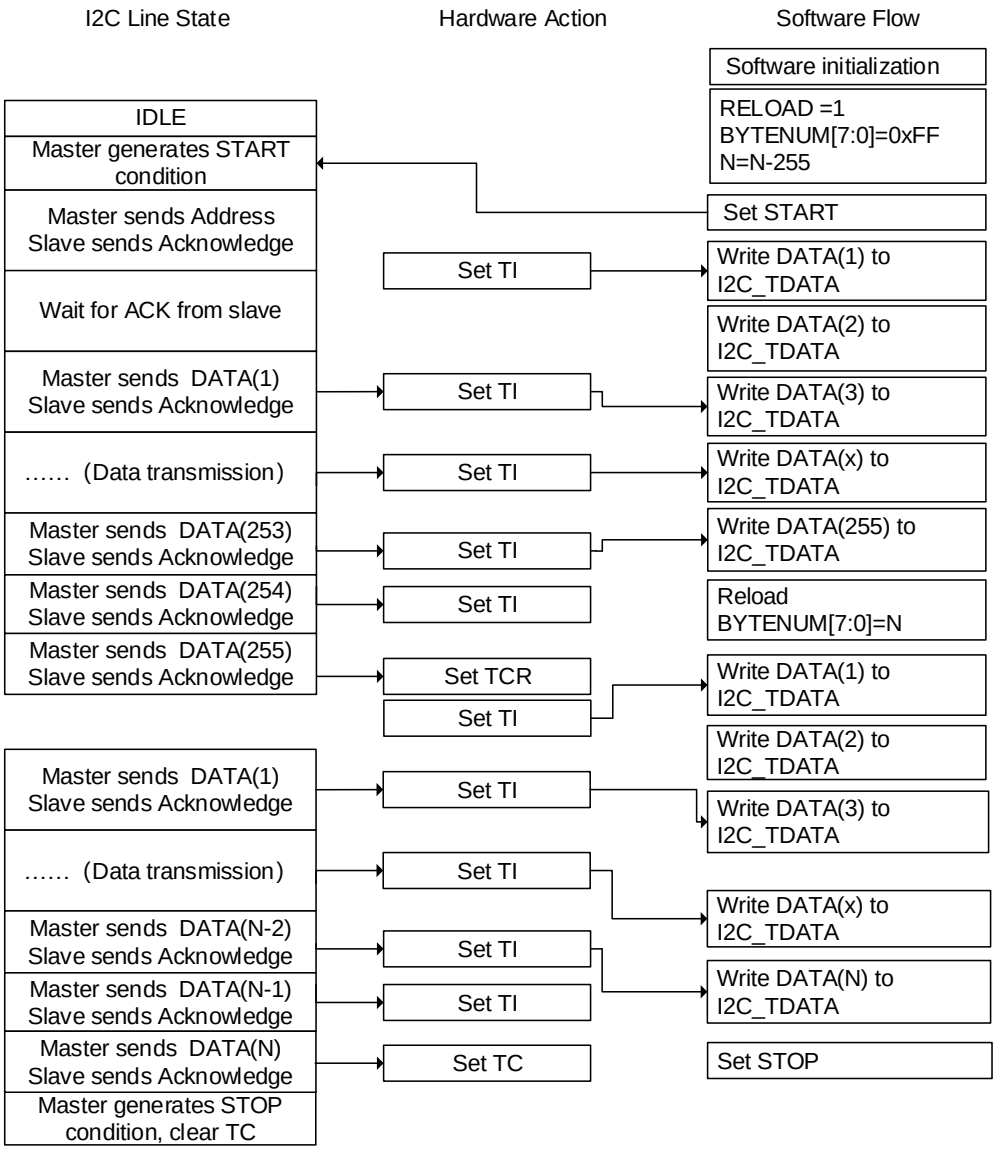


Figure 25-19. Programming model for master transmitting (N>255)



Programming model in master receiving mode

In master receiving mode, the RBNE bit in I2C_STAT register will be set when a byte is received. If the RBNEIE bit is set in I2C_CTL0 register, an interrupt will be generated. If the number of bytes to be received is greater than 255, RELOAD bit in I2C_CTL1 register must be set to enable the reload mode. In reload mode, when data of BYTENUM [7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches until BYTENUM [7:0] is modified with a non-zero value.

If data of BYTENUM [7:0] bytes have been transferred and RELOAD=0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START/STOP bit is set.

Figure 25-20. Programming model for master receiving (N<=255)

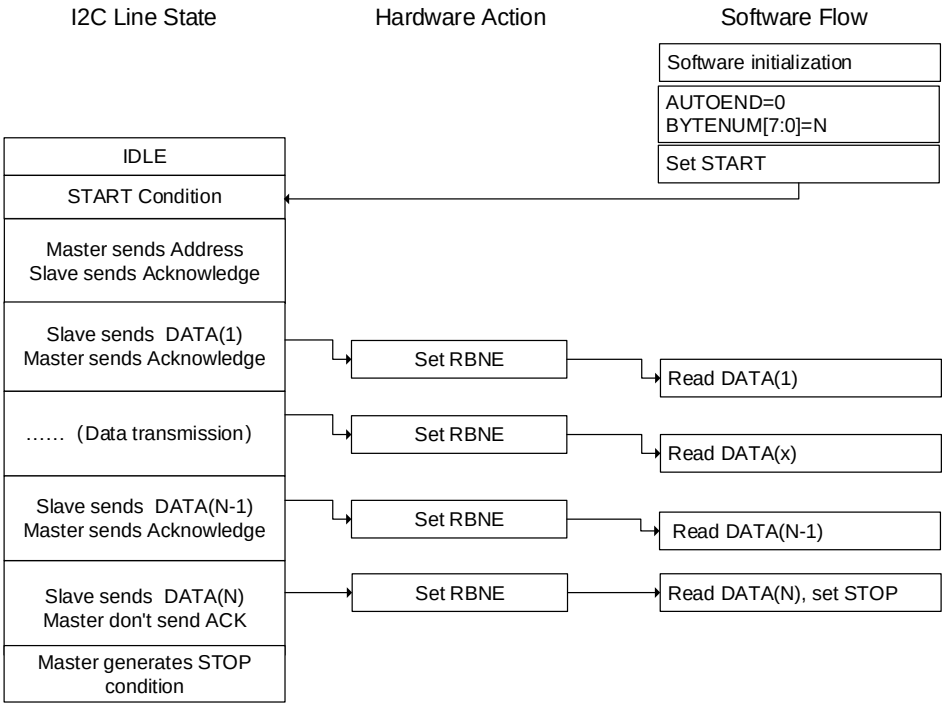
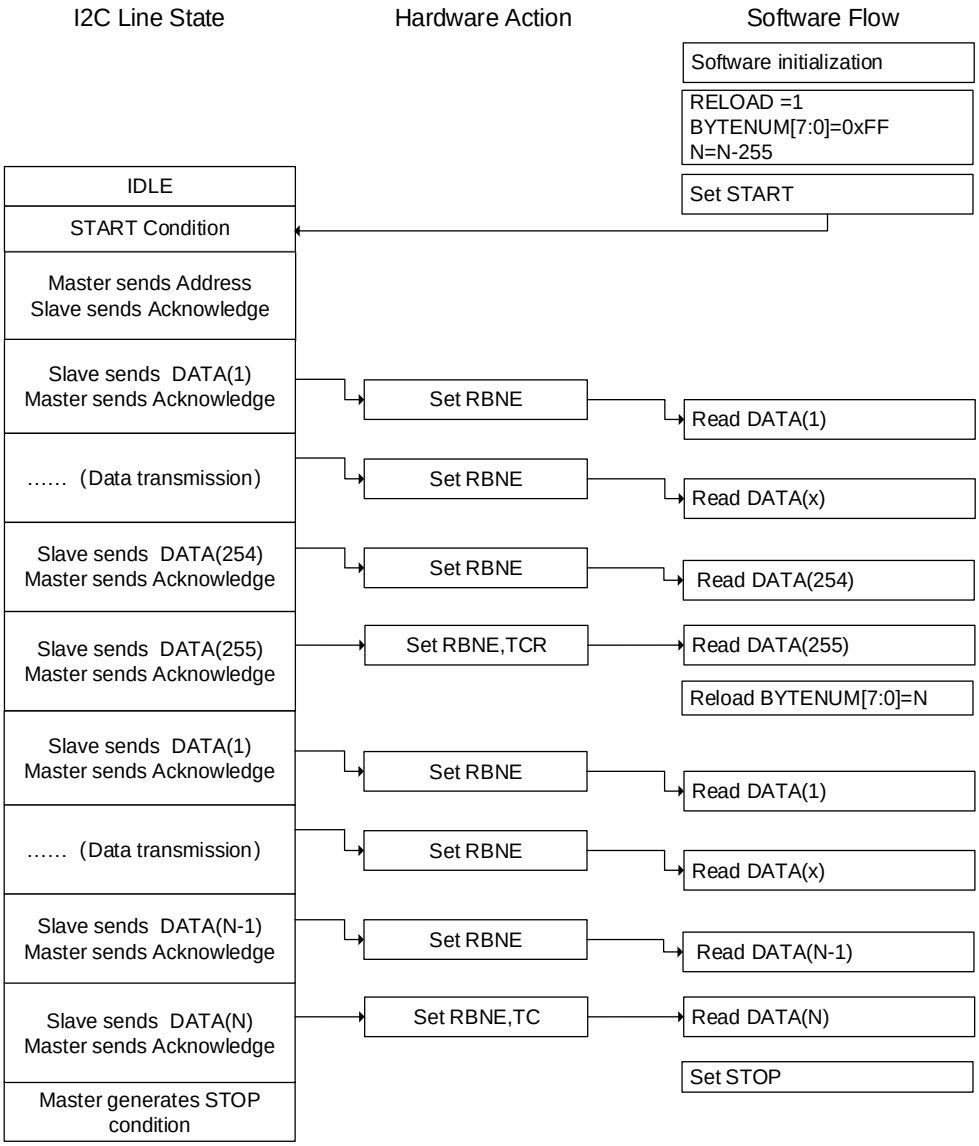


Figure 25-21. Programming model for master receiving (N>255)



25.3.9. SMBus support

The System Management Bus (abbreviated to SMBus or SMB) is a single-ended simple two-wire bus for the purpose of lightweight communication. Most commonly it is found in computer motherboards for communication with power source for ON/OFF instructions. It is derived from I2C for communication with low-bandwidth devices on a motherboard, especially power related chips such as a laptop's rechargeable battery subsystem (see Smart Battery Data).

SMBus protocol

Each message transaction on SMBus follows the format of one of the defined SMBus protocols. The SMBus protocols are a subset of the data transfer formats defined in the I2C specifications. I2C devices that can be accessed through one of the SMBus protocols are compatible with the SMBus specifications. I2C devices that do not adhere to these protocols cannot be accessed by standard methods as defined in the SMBus and Advanced

Configuration and Power Management Interface (abbreviated to ACPI) specifications.

Address resolution protocol

The SMBus uses I2C hardware and I2C hardware addressing, but adds second-level software for building special systems. Additionally, its specifications include an Address Resolution Protocol that can make dynamic address allocations. Dynamic reconfiguration of the hardware and software allow bus devices to be 'hot-plugged' and used immediately, without restarting the system. The devices are recognized automatically and assigned unique addresses. This advantage results in a plug-and-play user interface. In this protocol there is a very useful distinction between a system host and all the other devices in the system, that is the host provides address assignment function.

SMBus slave byte control

The slave byte control of SMBus receiver is the same as I2C. It allows the ACK control of each byte. The Slave Byte Control mode must be enabled by setting SBCTL bit in I2C_CTL0 register.

Host Notify protocol

When the SMBHAEN bit in the I2C_CTL0 register is set, the SMBus supports the host notify protocol. In this protocol, the device acts as a master and the host as a slave, and the host will acknowledge the SMBus host address.

Time-out feature

SMBus has a time-out feature which resets devices if a communication takes too long. This explains the minimum clock frequency of 10 kHz to prevent locking up the bus. I2C can be a 'DC' bus, meaning that a slave device stretches the master clock when performing some routine while the master is accessing it. This will notify to the master that the slave is busy but does not want to lose the communication. The slave device will allow continuation after its task is completed. There is no limit in the I2C bus protocol as to how long this delay can be, whereas for a SMBus system, it would be limited to 25~35ms. SMBus protocol just assumes that if something takes too long, then it means that there is a problem on the bus and that all devices must reset in order to clear this mode. Slave devices are not allowed to hold the clock low too long.

The timeout detection can be enabled by setting TOEN and EXTOEN bits in the I2C_TIMEOUT register. The timer must be configured to guarantee that the timeout detected before the maximum time given in the SMBus specification.

The value programmed in BUSTOA[11:0] is used to check the $t_{TIMEOUT}$ parameter. To detect the SCL low level timeout, the TOIDLE bit must be 0. And the timer can be enabled by setting the TOEN bit in the I2C_TIMEOUT register, after the TOEN bit is set, the BUSTOA [11:0] and the TOIDLE bit cannot be changed. If the low level time of SCL is greater than (BUSTOA+1)*2048*t_{I2CCLK}, the TIMEOUT flag will be set in I2C_STAT register.

The BUSTOB[11:0] is used to check the $t_{LOW:SEXT}$ of the slave and the $t_{LOW:MEXT}$ of the master. The timer can be enabled by setting the EXTOEN bit in the I2C_TIMEOUT register, after the EXTOEN bit is set, the BUSTOB [11:0] cannot be changed. If the SCL stretching time of the SMBus peripheral is greater than (BUSTOB+1)*2048*t_{I2CCLK} and within the timeout interval described in the bus idle detection section, the TIMEOUT bit in the I2C_STAT register will be set.

Packet error checking

There is a CRC-8 calculator in I2C block to perform Packet Error Checking for I2C data. A PEC (packet error code) byte is appended at the end of each transfer. The byte is calculated as CRC-8 checksum, calculated over the entire message including the address and read/write bit. The polynomial used is x^8+x^2+x+1 (the CRC-8-ATM HEC algorithm, initialized to zero).

When I2C is disabled, the PEC can be enabled by setting the PECEN bit in I2C_CTL0 register. Since the PEC transmission is managed by BYTENUM [7:0] in I2C_CTL1 register, SBCTL bit must be set when act as a slave. When PECTRANS is set and the RELOAD bit is cleared, PEC is transmitted after the BYTENUM [7:0]-1 data byte. The PECTRANS has no effect if RELOAD is set.

Table 25-4. SMBus with PEC configuration

Mode	SBCTL bit	RELOAD bit	AUTOEND bit	PECTRANS bit
Master Tx/Rx BYTENUM + PEC+ STOP	x	0	1	1
Master Tx/Rx BYTENUM + PEC + RESTART	x	0	0	1
Slave Tx/Rx with PEC	1	0	x	1

SMBus alert

The SMBus has an extra optional shared interrupt signal called SMBALERT# which can be used by slaves to tell the host to ask its slaves about events of interest. The host processes the interrupt and accesses all SMBALERT# devices through the Alert Response Address at the same time. If the SMBALERT# is pulled low by the devices, the devices will acknowledge the Alert Response Address. When SMBHAEN is 0, it is configured as a slave device, the SMBA pin will be pulled low by setting the SMBALTEN bit in the I2C_CTL0 register. Meanwhile the Alert Response Address is enabled. When SMBHAEN is 1, it is configured as a host, and the SMBALTEN is 1, as soon as a falling edge is detected on the SMBA pin, the SMBALT flag will be set in the I2C_STAT register. If the ERRIE bit is set in the I2C_CTL0 register, an interrupt will be generated. When SMBALTEN is 0, the level of ALERT line is considered high even if the SMBA pin is low. The SMBA pin can be used as a standard GPIO if SMBALTEN is 0.

Bus idle detection

If the master detects that the high level duration of the clock and data signals is greater than $t_{HIGH,MAX}$, the bus can be considered idle.

This timing parameter includes the case of a master that has been dynamically added to the bus and may not have detected a state transition on a SMBCLK or SMBDAT lines. In this case, in order to ensure that there is no ongoing transmission, the master must wait long enough.

The BUSTOA[11:0] bits must be programmed with the timer reload value to enable the t_{IDLE} check in order to obtain the t_{IDLE} parameter. To detect SCL and SDA high level timeouts, the TOIDLE bit must be set. Then setting the TOEN bit in the I2C_TIMEOUT register to enable the timer, after the TOEN bit is set, the BUSTOA [11:0] bit and the TOIDLE bit cannot be changed. If the high level time of both SCL and SDA is greater than (BUSTOA+1)*4*t_{I2CCLK}, the TIMEOUT flag will be set in the I2C_STAT register.

SMBus slave mode

The SMBus receiver must be able to NACK each command or data it receives. For ACK control in slave mode, slave byte control mode can be enabled by setting SBCTL bit in I2C_CTL0 register.

SMBus-specific addresses should be enabled when needed. The SMBus Device Default address (0b1100 001) is enabled by setting the SMBDAEN bit in the I2C_CTL0 register. The SMBus Host address (0b0001 000) is enabled by setting the SMBHAEN bit in the I2C_CTL0 register. The Alert Response Address (0b0001 100) is enabled by setting the SMBALTEN bit in the I2C_CTL0 register.

25.3.10. SMBus mode

SMBus Master Transmitter and Slave Receiver

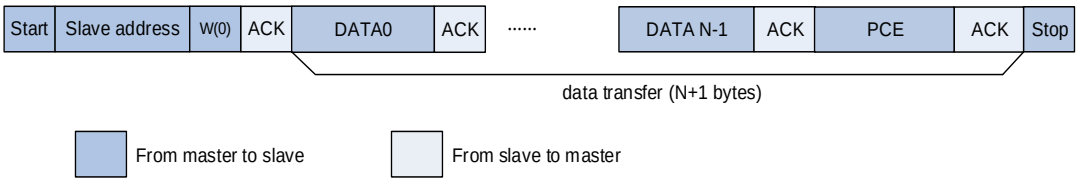
The PEC in SMBus master mode can be transmitted by setting the PECTRANS bit before setting the START bit, and the number of bytes in the BYTENUM [7:0] field must be configured. In this case, the total number of transmissions when TI interrupt occur is BYTENUM-1. So if BYTENUM=0x1 and PECTRANS bit is set at the same time, the contents of the I2C_PEC register are automatically transferred. If the automatic end mode is selected (AUTOEND=1), the SMBus master automatically sends the STOP signal after the PEC byte. If the automatic end mode is not selected (AUTOEND=0), the SMBus master can send a RESTART signal after the PEC. The I2C_PEC register content will be sent after BYTENUM-1 bytes, and the TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART must be set in the TC interrupt routine.

When used as slave receiver, in order to allow PEC checking at the end of the number of bytes transmitted, SBCTL must be set. To configure ack control for each byte, the RELOAD must be set. In order to check the PEC byte, it is necessary to clear the RELOAD bit and set PECTRANS bit. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register. If the PEC values does not match, the NACK is automatically generated. If the PEC values matches, the ACK is automatically generated, regardless of the NACKEN bit value. When PEC byte is received, it is also copied into the

I2C_RDATA register, and RBNE flag will be set. If the ERRIE bit in I2C_CTL0 register is 1, when PEC value does not match, the PECERR flag will be set and the interrupt will be generated. If ACK control is not required, then PECTRANS can be set to 1 and BYTENUM can be programmed according to the number of bytes to be received.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 25-22. SMBus Master Transmitter and Slave Receiver communication flow



SMBus Master Receiver and Slave Transmitter

If the SMBus master is required to receive PEC at the end of bytes transfer, automatic end mode can be enabled. Before sending a START signal on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. A NACK is respond to the PEC byte before STOP signal.

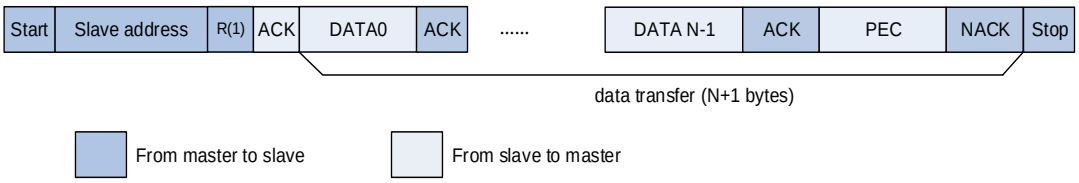
If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, the software mode (AUTOEND = 0) must be selected. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the contents of the I2C_PEC register automatically. The TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART can be set in the TC interrupt routine.

When used as slave transmitter, in order to allow PEC transmission at the end of BYTENUM [7:0] bytes, SBCTL must be set. If PECTRANS bit is set, the number of bytes in BYTENUM [7:0] contains PEC byte. In this case, if the number of bytes requested by the master is greater than BYTENUM-1, the total number of TI interrupts will be BYTENUM-1, and the contents of the I2C_PEC register will be transmitted automatically.

Note:

1. After the RELOAD bit is set, the PECTRANS cannot be changed.
2. When SMBus acts as a slave transmitter and uses the PEC function, BYTENUM[7:0] must be configured before detecting the start signal.

Figure 25-23. SMBus Master Receiver and Slave Transmitter communication flow



25.3.11. Use DMA for data transfer

As is shown in I2C slave mode and I2C master mode, each time TI or RBNE is asserted, software should write or read a byte, this may cause CPU's high overload. The DMA controller can be used to process TI and RBNE flag: each time TI or RBNE is asserted, DMA controller does a read or write operation automatically.

The DMA transmission request is enabled by setting the DENT bit in I2C_CTL0 register. The DMA reception request is enabled by setting the DENR bit in I2C_CTL0 register. In master mode, the slave address, transmission direction, number of bytes and START bit are programmed by software. The DMA must be initialized before setting the START bit. The number of bytes to be transferred is configured in the BYTENUM [7:0] in I2C_CTL1 register. In slave mode, the DMA must be initialized before the address match event or in the ADDSEND interrupt routine, before clearing the ADDSEND flag.

25.3.12. I2C error and interrupts

The I2C error flags are listed in ***Table 25-5. I2C error flags***.

Table 25-5. I2C error flags

I2C Error Name	Description
BERR	Bus error
LOSTARB	Arbitration lost
OUERR	Overrun/Underrun flag
PECERR	CRC value doesn't match
TIMEOUT	Bus timeout in SMBus mode
SMBALT	SMBus Alert

The I2C interrupt events and flags are listed in ***Table 25-6. I2C interrupt events***.

Table 25-6. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP signal detected in slave mode	STPDET	STPDETIE
Transfer complete reload	TCR	TCIE
Transfer complete	TC	
Address match	ADDSEND	ADDMIE
Not acknowledge received	NACK	NACKIE
Bus error	BERR	ERRIE
Arbitration Lost	LOSTARB	
Overrun/Underrun error	OUERR	
PEC error	PECERR	
Timeout error	TIMEOUT	
SMBus Alert	SMBALT	

25.3.13. I2C debug mode

When the microcontroller enters the debug mode (Cortex®-M33 core halted), the SMBus timeout either continues to work normally or stops, depending on the I2Cx_HOLD configuration bits in the DBG module.

25.4. Register definition

I2C0 base address: 0x4000 5400

I2C1 base address: 0x4000 5800

25.4.1. Control register 0 (I2C_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PECEN	SMBALTEN	SMBDAEN	SMBHAEN	GCEN	Reserved	SS	SBCTL
								rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	Reserved	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETIE	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	PECEN	PEC Calculation Switch 0: PEC Calculation off 1: PEC Calculation on
22	SMBALTEN	SMBus Alert enable 0: SMBA pin is not pulled down (device mode) or SMBus Alert pin SMBA is disabled (host mode) 1: SMBA pin is pulled down (device mode) or SMBus Alert pin SMBA is enabled (host mode)
21	SMBDAEN	SMBus device default address enable 0: Device default address is disabled, the default address 0b1100001x will be not acknowledged. 1: Device default address is enabled, the default address 0b1100001x will be acknowledged.
20	SMBHAEN	SMBus host address enable 0: Host address is disabled, address 0b0001000x will be not acknowledged. 1: Host address is enabled, address 0b0001000x will be acknowledged.
19	GCEN	Whether or not to response to a General Call (0x00) 0: Slave won't response to a General Call 1: Slave will response to a General Call

18	Reserved	Must be kept at reset value.
17	SS	Whether to stretch SCL low when data is not ready in slave mode. This bit is set and cleared by software. 0: SCL Stretching is enabled 1: SCL Stretching is disabled **Note:** When in master mode, this bit must be 0. This bit can be modified when I2CEN = 0.
16	SBCTL	Slave byte control This bit is used to enable hardware byte control in slave mode. 0: Slave byte control is disabled 1: Slave byte control is enabled
15	DENR	DMA enable for reception 0: DMA is disabled for reception 1: DMA is enabled for reception
14	DENT	DMA enable for transmission 0: DMA is disabled for transmission 1: DMA is enabled for transmission
13	Reserved	Must be kept at reset value.
12	ANOFF	Analog noise filter disable 0: Analog noise filter is enabled 1: Analog noise filter is disabled **Note:** This bit can only be programmed when the I2C is disabled (I2CEN = 0).
11:8	DNF[3:0]	Digital noise filter These bits are used to configure the digital noise filter on SDA and SCL input. The digital filter will filter spikes with a length of up to DNF[3:0]*t_{I2CCLK} 0000: Digital filter is disabled 0001: Digital filter is enabled and filter spikes with a length of up to 1 t_{I2CCLK} ... 1111: Digital filter is enabled and filter spikes with a length of up to15 t_{I2CCLK} These bits can only be modified when the I2C is disabled (I2CEN = 0).
7	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled. When BERR, LOSTARB, OUERR, PECERR, TIMEOUT or SMBALT bit is set, an interrupt will be generated.
6	TCIE	Transfer complete interrupt enable 0: Transfer complete interrupt is disabled 1: Transfer complete interrupt is enabled
5	STPDETIE	Stop detection interrupt enable 0: Stop detection (STPDET) interrupt is disabled

		1: Stop detection (STPDET) interrupt is enabled
4	NACKIE	Not acknowledge received interrupt enable 0: NACK received interrupt is disabled 1: NACK received interrupt is enabled
3	ADDMIE	Address match interrupt enable in slave mode 0: Address match interrupt is disabled 1: Address match interrupt is enabled
2	RBNEIE	Receive interrupt enable 0: Receive (RBNE) interrupt is disabled 1: Receive (RBNE) interrupt is enabled
1	TIE	Transmit interrupt enable 0: Transmit (TI) interrupt is disabled 1: Transmit (TI) interrupt is enabled
0	I2CEN	I2C peripheral enable 0: I2C is disabled 1: I2C is enabled

25.4.2. Control register 1 (I2C_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PECTRANS	PEC Transfer Set by software. Cleared by hardware in the following cases: When PEC byte is transferred or ADDSEND bit is set or STOP signal is detected or I2CEN=0. 0: Don’t transfer PEC value 1: Transfer PEC

		Note: This bit has no effect when RELOAD=1, or SBCTL=0 in slave mode.
25	AUTOEND	Automatic end mode in master mode 0: TC bit is set when the transfer of BYTENUM [7:0] bytes is completed. 1: a STOP signal is sent automatically when the transfer of BYTENUM [7:0] bytes is completed. **Note:** This bit works only when RELOAD=0. This bit is set and cleared by software.
24	RELOAD	Reload mode 0: After the data of BYTENUM [7:0] bytes transfer, the transfer is completed. 1: After data of BYTENUM [7:0] bytes transfer, the transfer is not completed and the new BYTENUM [7:0] will be reloaded. Every time when the BYTENUM [7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set. This bit is set and cleared by software.
23:16	BYTENUM[7:0]	Number of bytes to be transferred These bits are programmed with the number of bytes to be transferred. When SBCTL=0, these bits have no effect. **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
15	NACKEN	Generate NACK in slave mode 0: an ACK is sent after receiving a new byte. 1: a NACK is sent after receiving a new byte. **Note:** The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP signal is detected or ADDSEND is set, or when I2CEN=0. When PEC is enabled, whether to send an ACK or a NACK is not depend on the NACKEN bit. When SS=1, and the OUERR bit is set, the value of NACKEN is ignored and a NACK will be sent.
14	STOP	Generate a STOP signal on I2C bus This bit is set by software and cleared by hardware when I2CEN=0 or STOP signal is detected. 0: STOP will not be sent 1: STOP will be sent
13	START	Generate a START signal on I2C bus This bit is set by software and cleared by hardware after the address is sent. When the arbitration is lost, or a timeout error occurred, or I2CEN=0, this bit can also be cleared by hardware. It can be cleared by software by setting the ADDSENDC bit in I2C_STATC register. 0: START will not be sent 1: START will be sent
12	HEAD10R	10-bit address header executes read direction only in master receive mode 0: The 10-bit master receive address sequence is START + header of 10-bit address (write) + slave address byte 2 + RESTART + header of 10-bit address (read). 1: The 10-bit master receive address sequence is RESTART + header of 10-bit

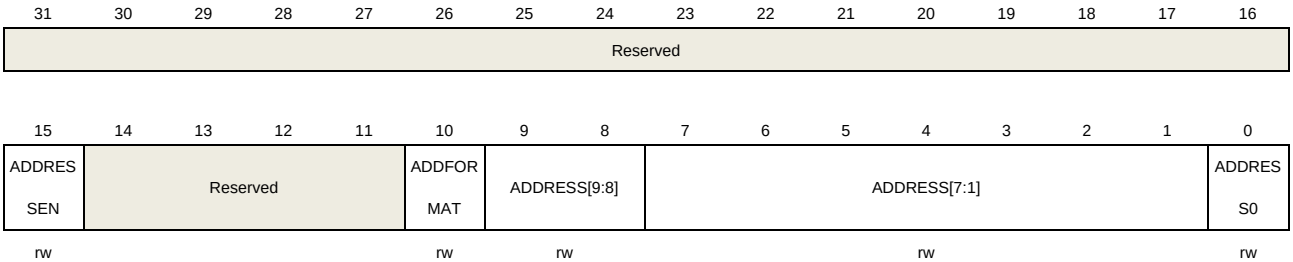
		address (read). **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
11	ADD10EN	10-bit addressing mode enable in master mode 0: 7-bit addressing in master mode 1: 10-bit addressing in master mode **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.
9:0	SADDRESS[9:0]	Slave address to be sent SADDRESS[9:8]: Slave address bit 9:8 If ADD10EN = 0, these bits have no effect. If ADD10EN = 1, these bits should be written with bits 9:8 of the slave address to be sent. SADDRESS[7:1]: Slave address bit 7:1 If ADD10EN = 0, these bits should be written with the 7-bit slave address to be sent. If ADD10EN = 1, these bits should be written with bits 7:1 of the slave address to be sent. SADDRESS0: Slave address bit 0 If ADD10EN = 0, this bit has no effect. If ADD10EN = 1, this bit should be written with bit 0 of the slave address to be sent **Note:** When the START bit is set to 1, this bit cannot be modified, and the software cannot change this bit during transmission.

25.4.3. Slave address register 0 (I2C_SADDR0)

Address offset: 0x08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15	ADDRESSEN	I2C address enable 0: I2C address disable. 1: I2C address enable.
14:11	Reserved	Must be kept at reset value.
10	ADDFORMAT	Address mode for the I2C slave 0: 7-bit address 1: 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.
9:8	ADDRESS[9:8]	Highest two bits of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.
7:1	ADDRESS[7:1]	7-bit address or bits 7:1 of a 10-bit address **Note:** 1. When ADDRESSEN is set, this bit should not be written. 2. When the slave address is set to the reserved address 0b'1111000, the slave will not respond if an address match occurs.
0	ADDRESS0	Bit 0 of a 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.

25.4.4. Slave address register 1 (I2C_SADDR1)

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
ADDRESS2EN	Reserved				ADDMSK2[2:0]			ADDRESS2[7:1]							Reserved
rw					rw			rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESS2EN	Second I2C address enable 0: Second I2C address disable. 1: Second I2C address enable.
14:11	Reserved	Must be kept at reset value.
10:8	ADDMSK2[2:0]	ADDRESS2[7:1] mask

		Defines which bits of ADDRESS2[7:1] are compared with an incoming address byte, and which bits are masked (don't care). 000: No mask, all the bits must be compared. N (001~110) : ADDRESS2[n:1] is masked. Only ADDRESS2[7:n+1] are compared. 111: ADDRESS2[7:1] are masked. All 7-bit received addresses are acknowledged except the reserved address (0b0000xxx and 0b1111xxx). **Note:** When ADDRESS2EN is set, these bits should not be written. If ADDMSK2 is not equal to 0, the reserved I2C addresses (0b0000xxx and 0b1111xxx) are not acknowledged even if all the bits are matched.
7:1	ADDRESS2[7:1]	Second I2C address for the slave **Note:** 1. When ADDRESS2EN is set, these bits should not be written. 2. When the slave address is set to the reserved address 0b'1111000, the slave will not respond if an address match occurs.
0	Reserved	Must be kept at reset value.

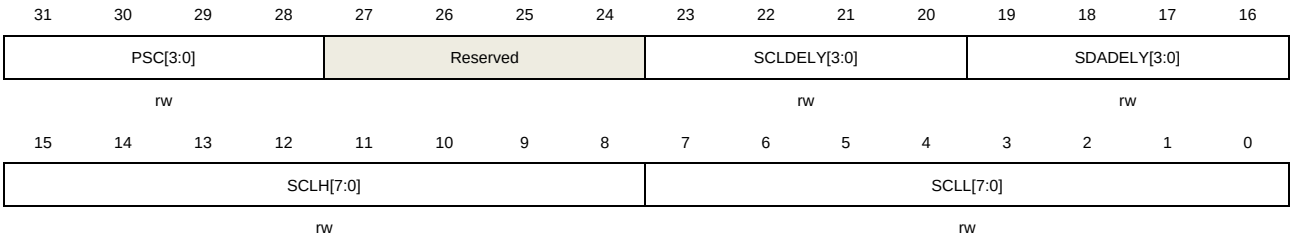
25.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

This register cannot be modified when the I2C is enabled (I2CEN=1)

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



Bits	Fields	Descriptions
31:28	PSC[3:0]	Timing prescaler In order to generate the clock period t_{PSC} used for data setup and data hold counters, these bits are used to configure the prescaler for I2CCLK. The t_{PSC} is also used for SCL high and low level counters. $t_{PSC}=(PSC+1)*t_{I2CCLK}$
27:24	Reserved	Must be kept at reset value.
23:20	SCLDELY[3:0]	Data setup time A delay $t_{SCLDELY}$ between SDA edge and SCL rising edge can be generated by configuring these bits. And during $t_{SCLDELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0.

$t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$

19:16	SDADELY[3:0]	Data hold time A delay $t_{SDADELY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. 0x0: $t_{SDADELY}$ = 1*t_{PSC} 0x1-0xF: $t_{SDADELY}$=SDADELY*t_{PSC}
15:8	SCLH[7:0]	SCL high period SCL high period can be generated by configuring these bits. $t_{SCLH}=(SCLH+1)*t_{PSC}$ **Note:** These bits can only be used in master mode.
7:0	SCLL[7:0]	SCL low period SCL low period can be generated by configuring these bits. $t_{SCLL}=(SCLL+1)*t_{PSC}$ **Note:** These bits can only be used in master mode.

25.4.6. Timeout register (I2C_TIMEOUT)

Address offset: 0x14

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTOEN	Reserved			BUSTOB[11:0]											
rw				rw											

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

Bits	Fields	Descriptions
31	EXTOEN	Extended clock timeout detection enable When a cumulative SCL stretch time is greater than $t_{LOW:EXT}$, a timeout error will be occurred. $t_{LOW:EXT}=(BUSTOB+1)*2048*t_{I2CCLK}$. 0: Extended clock timeout detection is disabled. 1: Extended clock timeout detection is enabled.
30:28	Reserved	Must be kept at reset value.
27:16	BUSTOB	Bus timeout B Configure the cumulative clock extension timeout. In master mode, the master cumulative clock low extend time $t_{LOW:MEXT}$ is detected. In slave mode, the slave cumulative clock low extend time $t_{LOW:SEXT}$ is detected. $t_{LOW:EXT}=(BUSTOB+1)*2048*t_{I2CCLK}$.

Note: These bits can be modified only when EXTOEN =0.

15	TOEN	Clock timeout detection enable If the SCL stretch time greater than $t_{TIMEOUT}$ when TOIDLE =0 or high for more than t_{IDLE} when TOIDLE =1, a timeout error is detected. 0: SCL timeout detection is disabled 1: SCL timeout detection is enabled
14:13	Reserved	Must be kept at reset value.
12	TOIDLE	Idle clock timeout detection 0: BUSTOA is used to detect SCL low timeout 1: BUSTOA is used to detect both SCL and SDA high timeout when the bus is idle **Note:** This bit can be written only when TOEN =0.
11:0	BUSTOA	Bus timeout A When TOIDLE=0, $t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK} When TOIDLE=1, t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK} **Note:** These bits can be written only when TOEN =0.

25.4.7. Status register (I2C_STAT)

Address offset: 0x18

Reset value: 0x0000 0001

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								READDR[6:0]							TR
											r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	Reserved	SMBALT	TIMEOUT	PECERR	OUERR	LOSTARB	BERR	TCR	TC	STPDET	NACK	ADDSEND	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:17	READDR[6:0]	Received match address in slave mode When the ADDSEND bit is set, these bits store the matched address. In the case of a 10-bit address, READDR[6:0] stores the header of the 10-bit address followed by the 2 MSBs of the address.
16	TR	Whether the I2C is a transmitter or a receiver in slave mode This bit is updated when the ADDSEND bit is set. 0: Receiver 1: Transmitter
15	I2CBSY	Busy flag

		This bit is set by hardware when a START signal is detected and cleared by hardware after a STOP signal. When I2CEN=0, this bit is also cleared by hardware. 0: No I2C communication. 1: I2C communication active.
14	Reserved	Must be kept at reset value.
13	SMBALT	SMBus Alert When SMBHAEN=1, SMBALTEN=1, and a SMBALERT event (falling edge) is detected on SMBA pin, this bit will be set by hardware. It is cleared by software by setting the SMBALTC bit. This bit is cleared by hardware when I2CEN=0. 0: SMBALERT event is not detected on SMBA pin 1: SMBALERT event is detected on SMBA pin
12	TIMEOUT	TIMEOUT flag. When a timeout or extended clock timeout occurred, this bit will be set. It is cleared by software by setting the TIMEOUTC bit and cleared by hardware when I2CEN=0. 0: no timeout or extended clock timeout occur 1: a timeout or extended clock timeout occur
11	PECERR	PEC error This flag is set by hardware when the received PEC does not match with the content of I2C_PEC register. Then a NACK is automatically sent. It is cleared by software by setting the PECERRC bit and cleared by hardware when I2CEN=0. 0: Received PEC and content of I2C_PEC match 1: Received PEC and content of I2C_PEC don't match, I2C will send NACK regardless of NACKEN bit.
10	OUERR	Overrun/Underrun error in slave mode In slave mode with SS=1, when an overrun/underrun error occurs, this bit will be set by hardware. It is cleared by software by setting the OUERRC bit and cleared by hardware when I2CEN=0. 0: No overrun or underrun occurs 1: Overrun or underrun occurs
9	LOSTARB	Arbitration Lost It is cleared by software by setting the LOSTARBC bit and cleared by hardware when I2CEN=0. 0: No arbitration lost. 1: Arbitration lost occurs and the I2C block changes back to slave mode.
8	BERR	Bus error When an unexpected START or STOP signal on I2C bus is detected, a bus error occurs and this bit will be set. It is cleared by software by setting BERRC bit and cleared by hardware when I2CEN=0. 0: No bus error 1: A bus error detected

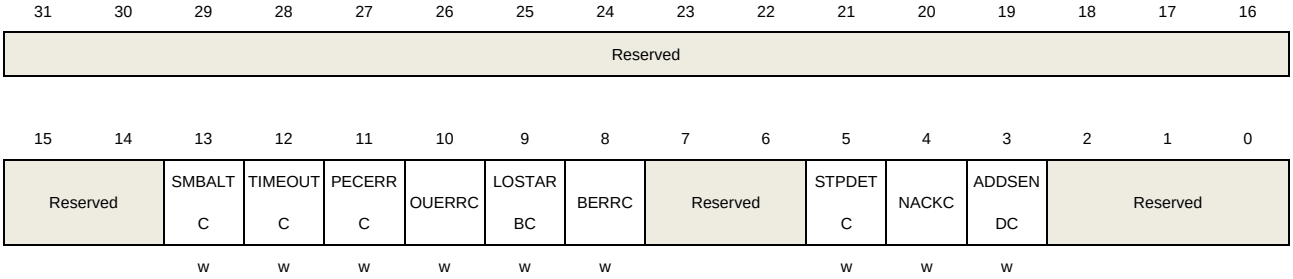
7	TCR	Transfer complete reload This bit is set by hardware when RELOAD=1 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when BYTENUM[7:0] is written to a non-zero value. 0: When RELOAD=1, transfer of BYTENUM[7:0] bytes is not completed 1: When RELOAD=1, transfer of BYTENUM[7:0] bytes is completed
6	TC	Transfer complete in master mode This bit is set by hardware when RELOAD=0, AUTOEND=0 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when START bit or STOP bit is set. 0: Transfer of BYTENUM[7:0] bytes is not completed 1: Transfer of BYTENUM[7:0] bytes is completed
5	STPDET	STOP signal detected in slave mode This flag is set by hardware when a STOP signal is detected on the bus. It is cleared by software by setting STPDETC bit and cleared by hardware when I2CEN=0. 0: STOP signal is not detected. 1: STOP signal is detected.
4	NACK	Not Acknowledge flag This flag is set by hardware when a NACK is received. It is cleared by software by setting NACKC bit and cleared by hardware when I2CEN=0. 0: ACK is received. 1: NACK is received.
3	ADDSEND	Address received matches in slave mode. This bit is set by hardware when the received slave address matched with one of the enabled slave addresses. It is cleared by software by setting ADDSENDC bit and cleared by hardware when I2CEN=0. 0: Received address not matched 1: Received address matched
2	RBNE	I2C_RDATA is not empty during receiving This bit is set by hardware when the received data is shift into the I2C_RDATA register. It is cleared when I2C_RDATA is read. 0: I2C_RDATA is empty 1: I2C_RDATA is not empty, software can read
1	TI	Transmit interrupt This bit is set by hardware when the I2C_TDATA register is empty and the I2C is ready to transmit data. It is cleared when the next data to be sent is written in the I2C_TDATA register. When SS=1, this bit can be set by software, in order to generate a TI event (interrupt if TIE=1 or DMA request if DENT =1). 0: I2C_TDATA is not empty or the I2C is not ready to transmit data 1: I2C_TDATA is empty and the I2C is ready to transmit data
0	TBE	I2C_TDATA is empty during transmitting

This bit is set by hardware when the I2C_TDATA register is empty. It is cleared when the next data to be sent is written in the I2C_TDATA register. This bit can be set by software in order to empty the I2C_TDATA register.
0: I2C_TDATA is not empty
1: I2C_TDATA is empty

25.4.8. Status clear register (I2C_STATC)

Address offset: 0x1C
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit
10	OUERRC	Overrun/Underrun flag clear. Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit
9	LOSTARBC	Arbitration Lost flag clear. Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit
7:6	Reserved	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by writing 1 to this bit
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by writing 1 to this bit

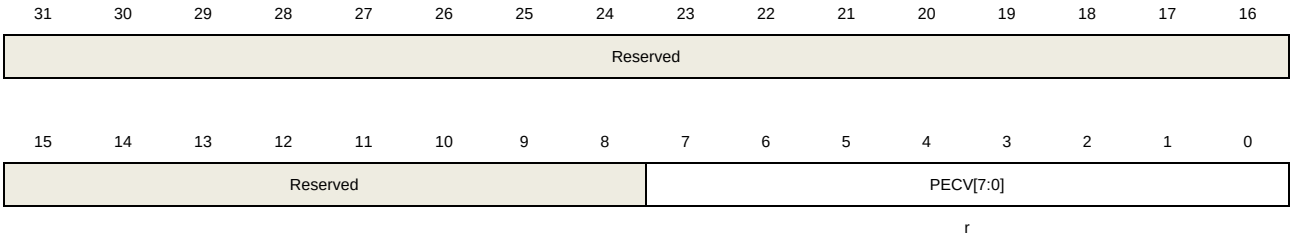
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by writing 1 to this bit
2:0	Reserved	Must be kept at reset value.

25.4.9. PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

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



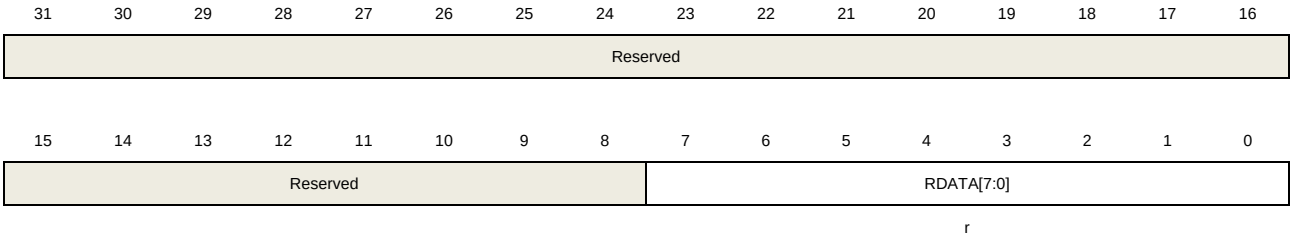
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

25.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

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



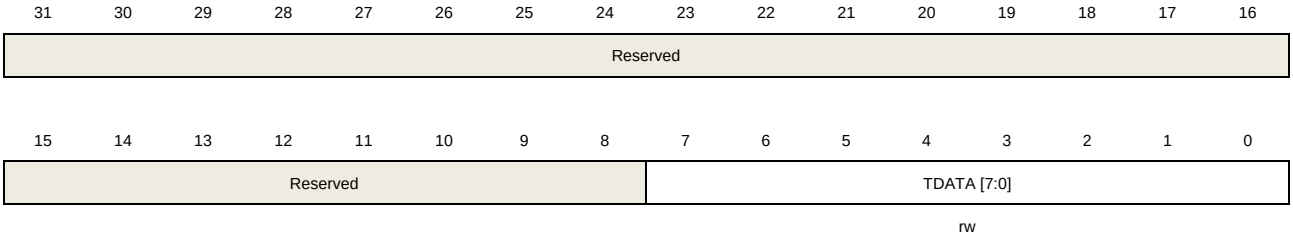
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

25.4.11. Transmit data register (I2C_TDATA)

Address offset: 0x28

Reset value: 0x0000 0000

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



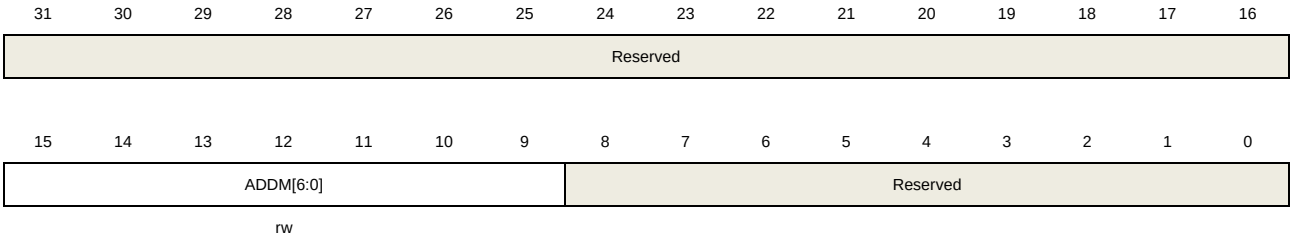
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

25.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

Reset value: 0x0000 FE00

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:9	ADDM[6:0]	Defines which bits of ADDRESS[7:1] are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in ADDM[6:0] enables comparisons with the corresponding bit in ADDRESS[7:1]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
8:0	Reserved	Must be kept at reset value.

26. Controller area network (CAN)

26.1. Overview

CAN bus (Controller Area Network) is a bus standard designed to allow microcontrollers and devices to communicate with each other without a host computer.

As CAN network interface, basic extended CAN supports the CAN protocols version 2.0A, 2.0B, ISO11898-1:2015 and BOSCH CAN FD specification. The CAN interface automatically handles the transmission and the reception of CAN frames. The CAN provides 28 scalable/configurable identifier filter banks. The filters are used for selecting the input message as software requirement and otherwise discarding the message. Three transmit mailboxes are provided to the software for transfer messages. The transmission scheduler decides which mailbox will be transmitted firstly. Three complete messages can be stored in every FIFO. The FIFOs are managed completely by hardware. Two receiving FIFOs are used by hardware to store the incoming messages. In addition, the CAN controller provides all hardware functions, which supports the time-triggered communication option, in safety-critical applications.

26.2. Characteristics

- Supports CAN protocols version 2.0A, B.
- Supports CAN FD Frame with up to 64 data bytes (ISO11898-1 and Bosch CAN FD specification V1.0).
- Baud rates up to 1 Mbit/s when classical frames and 8 Mbit/s when FD frames.
- Supports transmitter delay compensation.
- Supports the time-triggered communication.
- Interrupt enable and clear.

Transmission

- Supports 3 transmit mailboxes.
- Supports priority of transmission message.
- Supports time stamp at SOF transmission.

Reception

- Supports 2 Rx FIFOs and each has 3 messages depth.
- 28 scalable/configurable identifier filter banks.
- FIFO lock.

Time-triggered communication

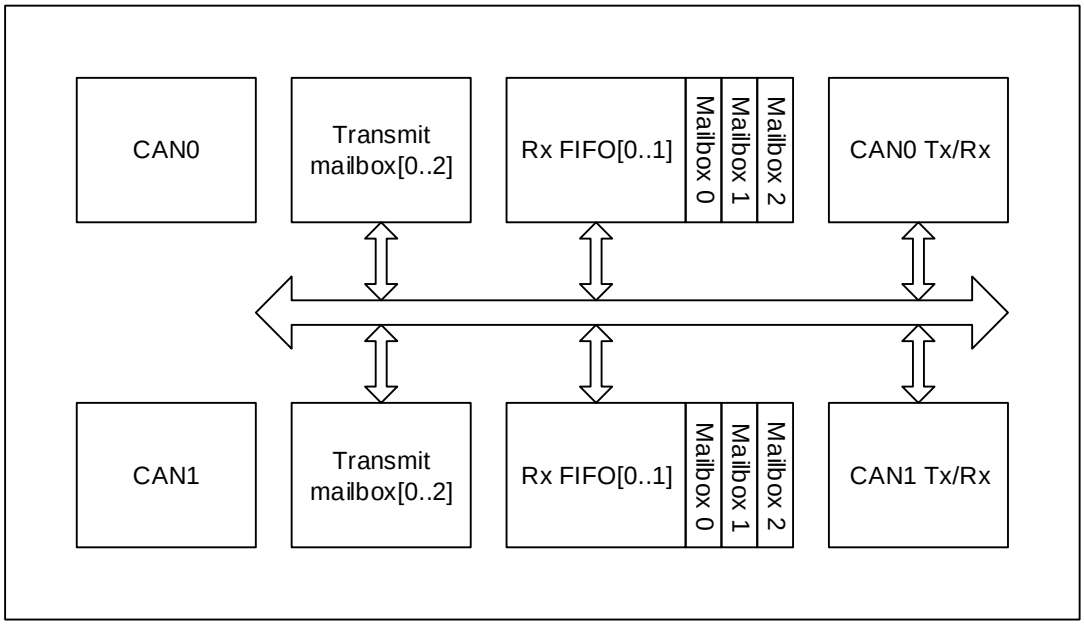
- Disable retransmission automatically in time-triggered communication mode.
- 16-bit free timer.
- Time stamp on SOF reception.

- Time stamp in last two data bytes transmission.

26.3. Function overview

Figure 26-1. CAN module block diagram shows the CAN block diagram.

Figure 26-1. CAN module block diagram



26.3.1. Working mode

The CAN interface has three working modes:

- Sleep working mode.
- Initial working mode.
- Normal working mode.

Sleep working mode

Sleep working mode is the default mode after reset. In sleep working mode, the CAN is in the low-power status and the CAN clock is stopped.

When SLPWMOD bit in CAN_CTL register is set, the CAN enters the sleep working mode. Then the SLPWS bit in CAN_STAT register is set by hardware.

To leave sleep working mode automatically: the AWU bit in CAN_CTL register is set and the CAN bus activity is detected. To leave sleep working mode by software: clear the SLPWMOD bit in CAN_CTL register.

Sleep working mode to initial working mode: set IWMOD bit and clear SLPWMOD bit in CAN_CTL register.

Sleep working mode to normal working mode: clear IWMOD and SLPWMOD bit in CAN_CTL

register.

Initial working mode

When the configuration of CAN bus communication is needed to be changed, the CAN must enter initial working mode.

When IWMOD bit in CAN_CTL register is set, the CAN enters the initial working mode. Then the IWS bit in CAN_STAT register is set.

Initial working mode to sleep working mode: set SLPWMOD bit and clear IWMOD bit in CAN_CTL register.

Initial working mode to normal working mode: clear IWMOD bit and clear SLPWMOD bit in CAN_CTL register.

Normal working mode

The CAN could communicate with other CAN communication nodes in normal working mode

To enter normal working mode: clear IWMOD and SLPWMOD bit in CAN_ CTL register.

Normal working mode to sleep working mode: set SLPWMOD bit in CAN_CTL register and wait the current transmission or reception completed.

Normal working mode to initial working mode: set IWMOD bit in CAN_CTL register, and wait the current transmission or reception completed.

26.3.2. Communication modes

The CAN interface has four communication modes:

- Silent communication mode.
- Loopback communication mode.
- Loopback and silent communication mode.
- Normal communication mode.

Silent communication mode

Silent communication mode means reception available and transmission disable.

The RX pin of the CAN could detect the signal from the network and the TX pin always holds in recessive state.

When the SCMOD bit in CAN_BT register is set, the CAN enters the silent communication mode. When it is cleared, the CAN leaves silent communication mode.

Silent communication mode is useful for monitoring the network messages.

Loopback communication mode

Loopback communication mode means the transmitted messages are transferred into the Rx FIFOs, the RX pin is disconnected from the CAN network and the TX pin can still send messages to the CAN network.

Setting LCMOD bit in CAN_BT register to enter loopback communication mode, while clearing it to leave. Loopback communication mode is useful for self-test.

Loopback and silent communication mode

Loopback and silent communication mode means the RX and TX pins are disconnected from the CAN network while the transmitted messages are transferred into the Rx FIFOs.

Setting LCMOD and SCMOD bit in CAN_BT register to enter loopback and silent communication mode, while clearing them to leave.

Loopback and silent communication mode is used for self-test. The TX pin holds in recessive state. The RX pin holds in high impedance state.

Normal communication mode

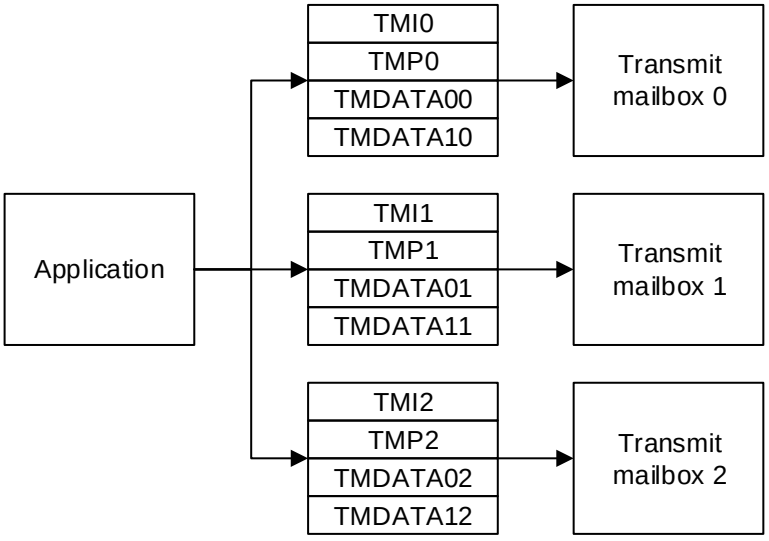
Normal communication mode is the default communication mode when the LCMOD and SCMOD bits in CAN_BT register are cleared.

26.3.3. Data transmission

Transmission register

Three transmit mailboxes are used for the application. Transmit mailboxes are used by configuring four transmission registers: CAN_TMIx, CAN_TMPx, CAN_TMDATA0x and CAN_TMDATA1x. As is shown in *Figure 26-2. Transmission register*.

Figure 26-2. Transmission register

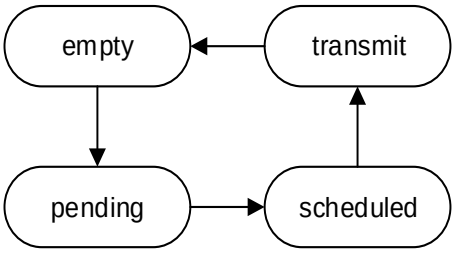


If FD frame would be transmitted, always write TMDATA00 registers when mailbox 0 is used, TMDATA01 register when mailbox 1 is used and TMDATA02 register when mailbox 2 is used until the end. For example, if application wants to transmit 64 bytes data using mailbox0, it needs to write the 64 bytes data through TMDATA00 register for 16 times. The data is stored in internal SRAM.

Transmit mailbox state

A transmit mailbox can be used when it is free (**empty state**). If the mailbox is filled with data, set TEN bit in CAN_TMIx register to prepare for starting the transmission **(pending state**). If more than one mailbox is in the pending state, they need scheduling the transmission (**scheduled state**). A mailbox with highest priority enters into transmit state and starts transmitting the message (**transmit state**). After the message has been sent, the mailbox is free (**empty state**). As is shown in ***Figure 26-3. State of transmit mailbox***.

Figure 26-3. State of transmit mailbox



Transmit status and error

The CAN_TSTAT register includes the transmit status and error bits: MTF, MTFNERR, MAL, MTE.

- MTF: mailbox transmit finished. Typically, MTF is set when the frame in the transmit mailbox has been sent.
- MTFNERR: mailbox transmit finished with no error. MTFNERR is set when the frame in the transmit mailbox has been sent without any error.
- MAL: mailbox arbitration lost. MAL is set when the frame transmission is failed due to the arbitration lost.
- MTE: mailbox transmit error. MTE is set when the frame transmission is failed due to the error detected on the CAN bus.

Steps of sending a frame

To send a frame through the CAN:

Step 1: Select one free transmit mailbox.

Step 2: Configure four transmission registers with the application's acquirement.

Step 3: Set TEN bit in CAN_TMIx register.

Step 4: Check the transmit status. Typically, MTF and MTFNERR are set if transmission is

successful.

Transmission options

Abort

MST bit in CAN_TSTAT register can abort the transmission.

If the transmit mailbox's status is **pending** or **scheduled**, the abort of transmission can be done immediately.

In the **transmit** state, the abort of transmission does not take effect immediately until the transmission is finished. In case that the transmission is successful, the MTFNERR and MTF in CAN_TSTAT are set and state changes to be **empty**. In case that the transmission is failed, the state changes to be **scheduled** and then the abort of transmission can be done immediately.

Priority

When more than one transmit mailbox is pending, the transmission order is given by the TFO bit in CAN_CTL register.

In case that TFO is 1, the three transmit mailboxes work first-in first-out (FIFO).

In case that TFO is 0, the transmit mailbox with lowest identifier has the highest priority of transmission. If the identifiers are equal, the lower mailbox number will be scheduled firstly.

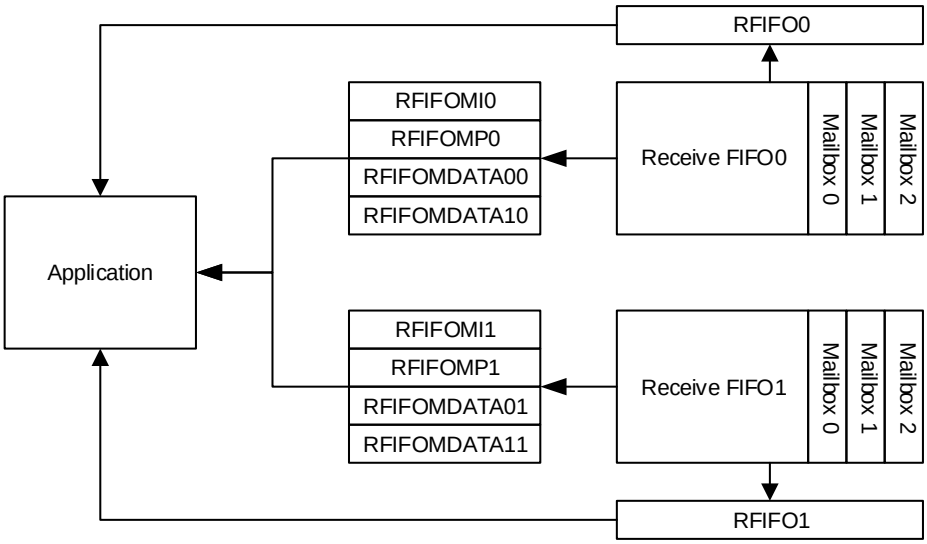
26.3.4. Data reception

Reception register

Two Rx FIFOs are used for the application. Rx FIFOs are managed by five registers: CAN_RFIFOx, CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x. FIFO's status and operation can be handled by CAN_RFIFOx register. Reception frame data can be achieved through the registers: CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x.

Each FIFO consists of three receive mailboxes. As is shown in ***Figure 26-4. Reception register***.

Figure 26-4. Reception register



Rx FIFO

Rx FIFO has three mailboxes. The reception frames are stored in the mailbox according to the arriving sequence. First arrived frame can be accessed by application firstly.

The number of frames in the Rx FIFO and the status can be accessed by the register CAN_RFIFO0 and CAN_RFIFO1.

If at least one frame has been stored in the Rx FIFO0, the frame data is stored in the CAN_RFIFOMI0, CAN_RFIFOMP0, CAN_RFIFOMDATA00 and CAN_RFIFOMDATA10 registers. After reading the current frame, set RFD bit in CAN_RFIFO0 to release a frame in the Rx FIFO and the software can read the next frame.

If FD frame has been received, the data always read from CAN_RFIFOMDATA00 register for FIFO0 and CAN_RFIFOMDATA01 for FIFO1 until the end. For example, if application needs to read 64 bytes data from FIFO0. It needs to read the 64 bytes data through CAN_RFIFOMDATA00 register for 16 times. The received data is stored in internal SRAM.

Rx FIFO status

RFL (Rx FIFO length) bits in CAN_RFIFOx register is 0 when no frame is stored in the Rx FIFO and it is 3 when FIFOx is full.

When RFF bit in CAN_RFIFOx register is set, it indicates FIFOx is full, at this time, RFL is 3.

When a new frame arrives after the FIFO has held three frames, the RFO bit in CAN_RFIFOx register will be set, and it indicates FIFOx is overrun. If the RFOD bit in CAN_CTL register is set, the new frame is discarded. If the RFOD bit in CAN_CTL register is reset, the new frame is stored into the Rx FIFO and the last frame in the Rx FIFO is discarded.

Steps of receiving a message

Step 1: Check the number of frames in the Rx FIFO.

Step 2: Read CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x.

Step 3: Set the RFD bit in CAN_RFIFOx register.

26.3.5. Filtering function

The CAN receives frames from the CAN bus. If the frame passes the filter, it is stored in the Rx FIFOs. Otherwise, the frame will be discarded without intervention by the software.

The identifier of frame is used for the matching of the filter.

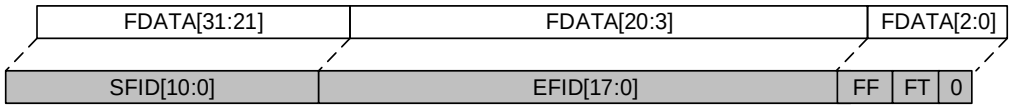
Scale

The filter consists of 28 banks: bank0 to bank27. Each bank has two 32-bit registers: CAN_FxDATA0 and CAN_FxDATA1.

Each filter bank can be configured to 32-bit or 16-bit.

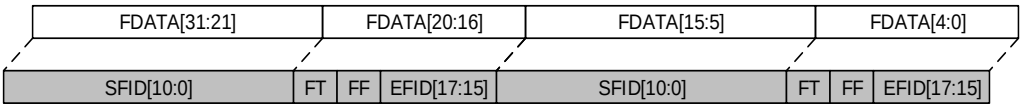
32-bit: SFID[10:0], EFID[17:0], FF and FT bits. As is shown in ***Figure 26-5. 32-bit filter***.

Figure 26-5. 32-bit filter



16-bit: SFID[10:0], FT, FF and EFID[17:15] bits. As is shown in ***Figure 26-6. 16-bit filter***.

Figure 26-6. 16-bit filter



Mask mode

For the Identifier of a data frame to be filtered, the mask mode is used to specify which bits must be the same as the preset Identifier and which bits need not be judged. 32-bit mask mode example is shown in ***Figure 26-7. 32-bit mask mode filter***.

Figure 26-7. 32-bit mask mode filter

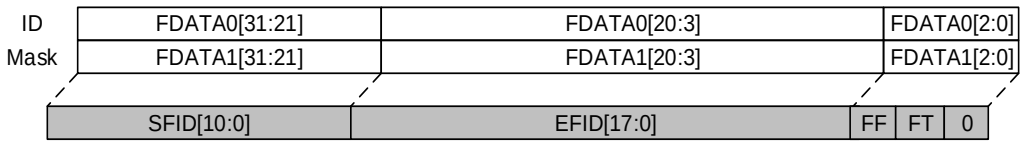
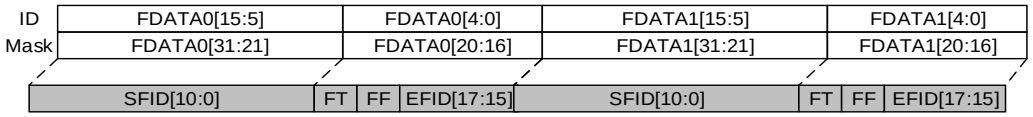


Figure 26-8. 16-bit mask mode filter



List mode

The filter consists of frame identifiers. The filter can determine whether a frame will be discarded or not. When one frame arrived, the filter will check which member can match the identifier of the frame.

32-bit list mode example is shown in ***Figure 26-9. 32-bit list mode filter***.

Figure 26-9. 32-bit list mode filter

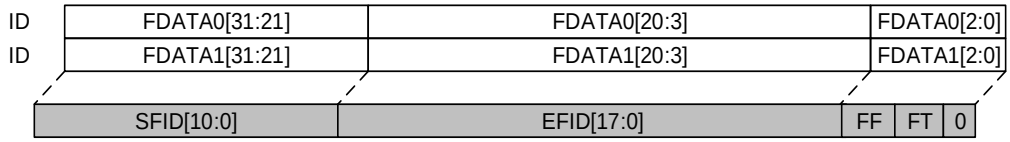
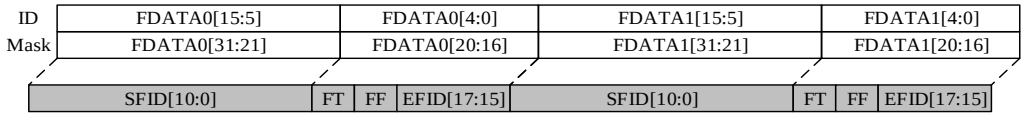


Figure 26-10. 16-bit list mode filter



Filter number

Filter consists of some filter bank. According to the mode and the scale of each of the filter banks, filter has different effects.

For example, there are two filter banks. Bank0 is configured as 32-bit mask mode. Bank1 is configured as 32-bit list mode. The filter number is shown in ***Table 26-1. 32-bit filter number***.

Table 26-1. 32-bit filter number

Filter bank	Filter data register	Filter number
0	F0DATA0-32bit-ID	0
	F0DATA1-32bit-Mask	
1	F1DATA0-32bit-ID	1
	F1DATA1-32bit-ID	2

Associated FIFO

28 banks can be associated with FIFO0 or FIFO1. If the bank is associated with FIFO0, the frames passed the bank will be stored in the FIFO0.

Active

The filter bank needs to be activated if the bank is to be used, otherwise, the filter bank should

be left deactivated.

Filtering index

Each filter number corresponds to a filtering rule. When the frame which is associated with a filter number N passes the filters, the filter index is N. It stores in the FI bits in CAN_RFIFOMPx.

Filter bank has filter index once it is associated with the FIFO no matter whether the bank is active or not.

The example about filtering index is shown in ***Table 26-2. Filtering index***.

Table 26-2. Filtering index

Filter bank	FIFO0	Active	Filter nunber	Filter bank	FIFO1	Active	Filter nunber
0	F0DATA0-32bits-ID	Yes	0	2	F2DATA0[15:0]-16bits-ID	Yes	0
	F0DATA1-32bits-Mask				F2DATA0[31:16]-16bits-Mask		
1	F1DATA0-32bits-ID	Yes	1		F2DATA1[15:0]-16bits-ID		1
	F1DATA1-32bits-ID		2		F2DATA1[31:16]-16bits-Mask		
3	F3DATA0[15:0]-16bits-ID	No	3	4	F4DATA0-32bits-ID	No	2
	F3DATA0[31:16]-16bits-Mask				F4DATA1-32bits-Mask		
	F3DATA1[15:0]-16bits-ID		4	5	F5DATA0-32bits-ID	No	3
	F3DATA1[31:16]-16bits-Mask				F5DATA1-32bits-ID		4
7	F7DATA0[15:0]-16bits-ID	No	5	6	F6DATA0[15:0]-16bits-ID	Yes	5
	F7DATA0[31:16]-16bits-ID		6		F6DATA0[31:16]-16bits-ID		6
	F7DATA1[15:0]-16bits-ID		7		F6DATA1[15:0]-16bits-ID		7
	F7DATA1[31:16]-16bits-ID		8		F6DATA1[31:16]-16bits-ID		8
8	F8DATA0[15:0]-16bits-ID	Yes	9	10	F10DATA0[15:0]-16bits-ID	No	9
	F8DATA0[31:16]-16bits-ID		10		F10DATA0[31:16]-16bits-Mask		
	F8DATA1[15:0]-16bits-ID		11		F10DATA1[15:0]-16bits-ID		10
	F8DATA1[31:16]-16bits-ID		12		F10DATA1[31:16]-16bits-Mask		
9	F9DATA0[15:0]-16bits-	Yes	13	11	F11DATA0[15:0]-16bits-ID	No	11

Filter bank	FIFO0	Active	Filter nunber	Filter bank	FIFO1	Active	Filter nunber
	ID						
	F9DATA0[31:16]-16bits-Mask				F11DATA0[31:16]-16bits-ID		12
	F9DATA1[15:0]-16bits-ID		14		F11DATA1[15:0]-16bits-ID		13
	F9DATA1[31:16]-16bits-Mask				F11DATA1[31:16]-16bits-ID		14
12	F12DATA0-32bits-ID	Yes	15	13	F13DATA0-32bits-ID	Yes	15
	F12DATA1-32bits-Mask				F13DATA1-32bits- ID		16

Priority

The filters have the priority rules:

1. 32-bits mode is higher than 16-bits mode.
2. List mode is higher than mask mode.
3. Smaller filter number has the higher priority.

26.3.6. Time-triggered communication

The time-triggered CAN protocol is a higher layer protocol on top of the CAN data link layer. Time-triggered communication means that activities are triggered by the elapsing of time segments. In a time-triggered communication system, all time points of message transmission are pre-defined.

In this mode, an internal 16-bit counter starts working, incrementing by 1 at each CAN bit time. This internal counter provides time stamps for sending and receiving data, stored in registers CAN _RFIFOMPx and CAN_TMPx.

The automatic retransmission is disabled in the time-triggered CAN communication.

26.3.7. Communication parameters

Automatic retransmission forbid mode

In time-triggered communication mode, the requirement for automatic retransmission must be disabled and CAN be met by setting ARD position 1 of the CAN_CTL register.

In this mode, the data is sent only once, and if the transmission fails due to arbitration failure or bus error, the CAN bus controller does not automatically resend the data as usual.

At the end of sending, the MTF bit of register CAN_TSTAT is hardware set to 1, and the sending status information can be obtained via MTFNERR, MAL, and MTE.

Bit time

On the bit-level, the CAN protocol uses synchronous bit transmission. This not only enhances the transmitting capacity but also requires a sophisticated method of bit synchronization. While bit synchronization in a character-oriented transmission (asynchronous) is performed upon the reception which the start bit is available with each character, the synchronous transmission protocol just need one start bit available at the beginning of a frame. To ensure that the receiver correctly reads the messages, resynchronization is required. Phase buffer segments' sample point of the front-end and back-end should be inserted a bit interval.

The CAN protocol regulates bus access by bit-wise arbitration. The signal propagated from sender to receiver and back to the sender must be completed within one bit-time. For synchronization, in addition to the phase buffer segments, a propagation delay segment is needed. The propagation delay segment is regarded as signal delays caused by transmitting and receiving nodes in the process of the signal propagation on the bus.

The normal bit time from the CAN protocol has three segments as follows:

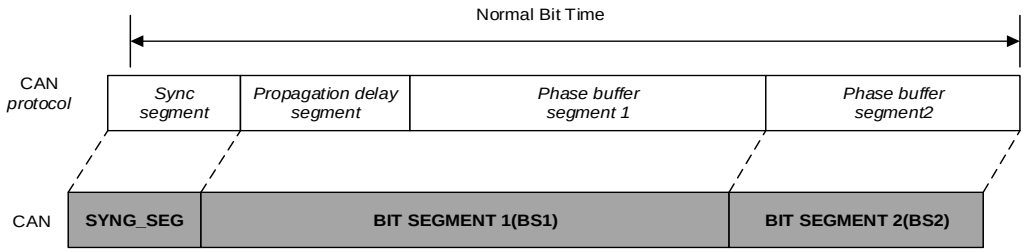
Synchronization segment (SYNC_SEG): a bit change is expected to occur within this time segment. It has a fixed length of one time quantum ($1 \times t_{CAN}$).

Bit segment 1 (BS1): It defines the location of the sample point. It includes the Propagation delay segment and Phase buffer segment 1 in the CAN standard. It may be automatically lengthened to compensate for positive phase drifts due to different frequency of the various nodes of the network.

Bit segment 2 (BS2): It defines the location of the transmit point. It represents the Phase buffer segment 2 in the CAN standard. It may also be automatically shortened to compensate for negative phase drifts.

The bit time is shown as in the ***Figure 26-11. The bit time***.

Figure 26-11. The bit time



The resynchronization Jump Width (SJW): it can be lengthened or shortened to compensate for the Synchronization error of the CAN network node.

The valid edge transition is defined as the first transition from a recessive bit to a dominant bit within a bit time, provided that the bus state detected at the previous sample point was recessive.

If a valid edge is detected in BS1, not in SYNC_SEG, BS1 is added up to SJW maximumly, so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2, not in SYNC_SEG, BS2 is cut down to SJW at most, so that the transmit point is moved earlier.

Note: For detailed explanations of hard synchronization and resynchronization, please refer to the ISO 11898 standard.

Baud rate

The CAN calculates its baud rate as follow:

$$BaudRate=\frac{1}{Normal\ Bit\ Time} \quad (26\text{-}1)$$

$$Normal\ Bit\ Time=t_{SYNC_SEG}+t_{BS1}+t_{BS2} \quad (26\text{-}2)$$

with:

$$t_{SYNC_SEG}=1\times t_q \quad (26\text{-}3)$$

$$t_{BS1}=(1+BT.BS1)\times t_q \quad (26\text{-}4)$$

$$t_{BS2}=(1+BT.BS2)\times t_q \quad (26\text{-}5)$$

$$t_q=(1+BT.BAUDPSC)\times t_{PCLK1} \quad (26\text{-}6)$$

26.3.8. CAN FD operation

The CAN FD function is enabled by setting FDEN to 1 in CAN_FDCTL register. If FDEN bit is cleared, only calassical frames are supported. If FDEN bit is set, it supports calassical frames and FD frames. Whether the current frame is FD or not could be defined by received FDF bit (the previously reserved bit in CAN frames with 11-bit identifiers or the first previously reserved bit in CAN frames with 29-bit identifiers which now is decoded as FDF bit). If FDF bit is recessive, meaning to be the CAN FD frame, otherwise FDF bit is dominant, meaning to be the classical frame.

The CAN FD supports ISO11898-1 or Bosch CAN FD Specification V1.0 by configuring NISO bit in CAN_FDCTL register.

The two bits following the FDF bit are reserved bit and BRS bit respectively. The received BRS bit determines the bit rate of data. When BRS is dominant, bit rate of data could not switch by configuring the CAN_DBT register. When BRS is recessive, bit rate of data could switch to a higher bit rate inside the frame by configuring the CAN_DBT register. The bit rate of data can be switched in the period from BRS bit to CRC delimiter (refer to ISO11898-1 or Bosch CAN FD Specification V1.0).

When Protocol Exception Handling is enabled (PRED bit in CAN_FDCTL register is 0), it causes the operation state to change to be IDLE and interrupts current frame at the next sample point when recessive reserve bit is received. When Protocol Exception Handling is disabled (PRED bit in CAN_FDCTL register is 1), it will treat a recessive reserve bit as a form error and respond with an error frame. If any recessive reserve bit occurs, set PRE bit in

CAN_FDSTAT register to 1.

The transmission of ESI bit (the bit before DLC bits, refer to ISO11898-1 or Bosch CAN FD Specification V1.0) is defined by ESIMOD bit in CAN_FDCTL register and ESI bit in CAN_TMPx register. If ESIMOD bit is 0, it will transmit the dominant bit by error active nodes and transmit the recessive bit by error passive nodes. If ESIMOD bit is set, it will transmit ESI bit in CAN_TMPx register.

The transmission of FDF bit and BRS bit is defined by FDF bit and BRS bit in CAN_TMPx registers.

26.3.9. Transmitter Delay Compensation

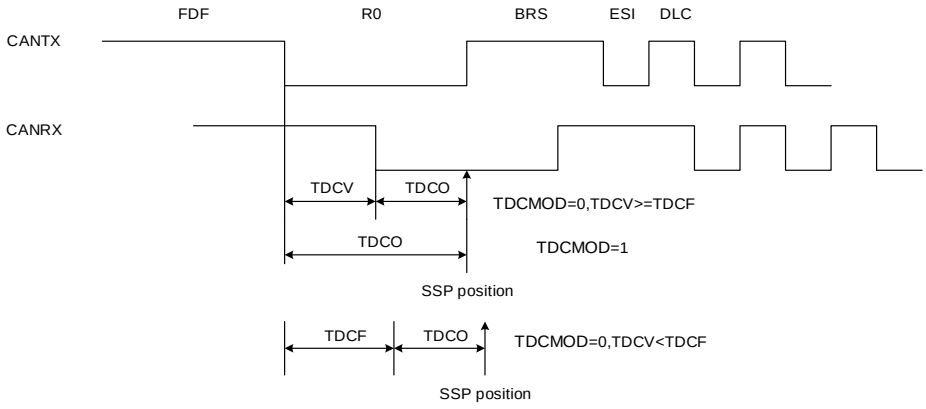
CAN FD supports transmitter delay compensation mechanism, it could be used in applications when the length of the CAN bit time in the DATA-PHASE is shorter than the limit required by the transceiver's internal delay time. The description of transmitter delay compensation, please refer to ISO11898-1 or Bosch CAN FD Specification V1.0.

The transmitter delay compensation is enabled by setting TDCEN bit in CAN_FDCTL register.

The transmitter delay compensation is used to adjust the position of the SSP. The SSP delay is defined as the delay from dominant edges on CANTX to SSP point. If TDCMOD bit in CAN_FDCTL register is set, the SSP delay is defined in TDCO bits of the CAN_FDTDC registers by software. If TDCMOD bit in CAN_FDCTL register is cleared, the hardware automatically uses the falling edges between FDF bit and RES0 bit to calculate the delay from dominant edges on CANTX to dominant edges on CANRX, and store the delay in TDCV bits of CAN_FDSTAT registers. In case that there is dominant glitch, SSP position would be advanced than expected, leading to a calculated error in compensation measurement. To solve this problem, TDCF bits of CAN_FDTDC register could be used to avoid too small TDCV bits. So the value of SSP delay is TDCO bits add TDCV bits if TDCV is larger than TDCF, or else the value of SSP delay is TDCO bits add TDCF bits.

The SSP delay can not exceed 3 data bit time.

Figure 26-12. Transmitter Delay Measurement



26.3.10. Error flags

The state of CAN bus can be reflected by Transmit Error Counter (TECNT) and Receive Error Counter (RECNT) of CAN_ERR register. The value can be increased or decreased by the hardware according to the error, and the software can judge the stability of the CAN network by these values. For details on incorrect counting, refer to the CAN protocol section.

Their values which read by software determine whether the CAN network is stable.By using the CAN_INTEN register (ERRIE bit, etc.), the software can control the interrupt generation when error is detected.

Bus-Off recovery

The CAN controller is in Bus-Off state when TECNT is over than 255. In This state, BOERR bit is set in CAN_ERR register, and no longer able to transmit and receive messages.

According to the ABOR configuration in register CAN_CTL, there are two ways to recover from Bus-Off (to an Error active state). Both of these methods require the CAN bus controller in the offline state to detect the Bus-Off recovery sequence defined by CAN protocol (when CAN_RX detects 128 consecutive 11-bit recessive bits) before automatic recovery.

If ABOR is set, it will be automatically recovered when a Bus-Off recovery sequence is detected.

If ABOR is cleared, CAN controller must be configured to enter initialization mode by setting IWMOD bit in CAN_CTL register, then exit and enter nomal mode. After this operation, it will recover when the recovering sequence is detected.

26.3.11. CAN interrupts

The CAN bus controller occupies 4 interrupt vectors, which are controlled by the register CAN _INTEN.

The interrupt sources can be classified as:

- Transmit interrupt.
- FIFO0 interrupt.
- FIFO1 interrupt.
- Error and status change interrupt.

Transmit interrupt

The transmit interrupt can be generated by any of the following conditions and TMEIE bit in CAN_INTEN register will be set:

- TX mailbox 0 transmit finished: MTF0 bit in the CAN_TSTAT register is set.
- TX mailbox 1 transmit finished: MTF1 bit in the CAN_TSTAT register is set.
- TX mailbox 2 transmit finished: MTF2 bit in the CAN_TSTAT register is set.

Rx FIFO0 interrupt

The Rx FIFO0 interrupt can be generated by the following conditions:

- Rx FIFO0 not empty: RFL0 bits in the CAN_RFIFO0 register are not '00' and RFNEIE0 in CAN_INTEN register is set.
- Rx FIFO0 full: RFF0 bit in the CAN_RFIFO0 register is set and RFFIE0 in CAN_INTEN register is set.
- Rx FIFO0 overrun: RFO0 bit in the CAN_RFIFO0 register is set and RFOIE0 in CAN_INTEN register is set.

Rx FIFO1 interrupt

The Rx FIFO1 interrupt can be generated by the following conditions:

- Rx FIFO1 not empty: RFL1 bits in the CAN_RFIFO1 register are not '00' and RFNEIE1 in CAN_INTEN register is set.
- Rx FIFO1 full: RFF1 bit in the CAN_RFIFO1 register is set and RFFIE1 in CAN_INTEN register is set.
- Rx FIFO1 overrun: RFO1 bit in the CAN_RFIFO1 register is set and RFOIE1 in CAN_INTEN register is set.

Error and working mode change interrupt

The error and working mode change interrupt can be generated by the following conditions:

- Error: ERRIF bit in the CAN_STAT register and ERRIE bit in the CAN_INTEN register are set. Refer to ERRIF description in the CAN_STAT register.
- Wakeup: WUIF bit in the CAN_STAT register is set and WIE bit in the CAN_INTEN register is set.
- Enter sleep working mode: SLPIF bit in the CAN_STAT register is set and SLPWIE bit in the CAN_INTEN register is set.

The CAN bus controller interrupt conditions can refer to ***Table 26-3. CAN Event / Interrupt flags***.

Table 26-3. CAN Event / Interrupt flags

Interrupt event	Interrupt / Event flag	Enable control bit
Transmit interrupt	Mailbox 0 transmit finished flag (MTF0)	TMEIE
	Mailbox 1 transmit finished flag (MTF1)	
	Mailbox 2 transmit finished flag (MTF2)	
FIFO0 interrupt	Rx FIFO0 length (RFL0[1:0])	RFNEIE0
	Rx FIFO0 full (RFF0)	RFFIE0
	Rx FIFO0 overfull (RFO0)	RFOIE0
FIFO1 interrupt	Rx FIFO1 length (RFL0[1:0])	RFNEIE1
	Rx FIFO1 full (RFF0)	RFFIE1
	Rx FIFO1 overfull (RFO0)	RFOIE1

Interrupt event	Interrupt / Event flag		Enable control bit	
EWMC interrupt	Warning error (WERR)	Error interrupt flag (ERRIF)	WERRIE	ERRIE
	Passive error (PERR)		PERRIE	
	Bus-Off error (BOERR)		BOIE	
	Error number (1<= ERRN[2:0] <= 6)		ERRNIE	
	Status change interrupt flag of waking up from sleep working mode (WUIF)		WIE	
	Status change interrupt flag of entering sleep working mode (SLPIF)		SLPWIE	

26.4. Register definition

CAN0 base address: 0x4001 5800

CAN1 base address: 0x4001 5C00

26.4.1. Control register (CAN_CTL)

Address offset: 0x00

Reset value: 0x0001 0002

This register 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															DFZ
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved							TTC	ABOR	AWU	ARD	RFOD	TFO	SLPWMOD	IWMOD
rs								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	DFZ	Debug freeze If the CANx_HOLD in DBG_CTL register is set, this bit defines the CAN controller is in debug freezing mode or normal working mode. If the CANx_HOLD in DBG_CTL register is cleared, this bit takes no effect. 0: CAN reception and transmission work normal even during debug 1: CAN reception and transmission stop working during debug
15	SWRST	Software reset 0: No effect 1: Reset CAN to enter sleep working mode. This bit is automatically reset to 0.
14:8	Reserved	Must be kept at reset value.
7	TTC	Time-triggered communication 0: Disable time-triggered communication 1: Enable time-triggered communication
6	ABOR	Automatic Bus-Off recovery 0: The Bus-Off state is left manually by software 1: The Bus-Off state is left automatically by hardware
5	AWU	Automatic wakeup If this bit is set, the CAN leaves sleep working mode when CAN bus activity is detected, and SLPWMOD bit in CAN_CTL register will be cleared automatically. 0: The sleeping working mode is left manually by software

		1: The sleeping working mode is left automatically by hardware
4	ARD	Automatic retransmission disable 0: Enable automatic retransmission 1: Disable automatic retransmission
3	RFOD	Rx FIFO overwrite disable 0: Enable Rx FIFO overwrite when Rx FIFO is full and overwrite the FIFO with the incoming frame 1: Disable Rx FIFO overwrite when Rx FIFO is full and discard the incoming frame
2	TFO	Tx FIFO order 0: Order with the identifier of the frame (the smaller identifier has higher priority) 1: Order with first-in and first-out
1	SLPWMOD	Sleep working mode If this bit is set by software, the CAN enters sleep working mode after current transmission or reception is completed. This bit can be cleared by software or hardware. If AWU bit in CAN_CTL register is set, this bit is cleared by hardware when CAN bus activity is detected. 0: Disable sleep working mode 1: Enable sleep working mode
0	IWMOD	Initial working mode 0: Disable initial working mode 1: Enable initial working mode

26.4.2. Status register (CAN_STAT)

Address offset: 0x04

Reset value: 0x0000 0C02

This register 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				RXL	LASTRX	RS	TS	Reserved			SLPIF	WUIF	ERRIF	SLPWS	IWS
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	RXL	RX level
10	LASTRX	Last sample value of RX pin
9	RS	Receiving state

0: CAN is not working in the receiving state
1: CAN is working in the receiving state

8	TS	Transmitting state 0: CAN is not working in the transmitting state 1: CAN is working in the transmitting state
7:5	Reserved	Must be kept at reset value.
4	SLPIF	Status change interrupt flag of entering sleep working mode This bit is set by hardware when entering sleep working mode, and cleared by hardware when the CAN is not in sleep working mode. This bit can also be cleared by software when writting 1 to this bit. 0: CAN is not in the sleep working mode 1: CAN is in the sleep working mode
3	WUIF	Status change interrupt flag of waking up from sleep working mode This bit is set when CAN bus activity event is detected in sleep working mode. This bit can be cleared by software when writting 1 to this bit. 0: Wakeup event is not coming 1: Wakeup event is coming
2	ERRIF	Error interrupt flag This bit is set by the following events. The BOERR bit in CAN_ERR register is set and BOIE bit in CAN_INTEN register is set. Or the PERR bit in CAN_ERR register is set and PERRIE bit in CAN_INTEN register is set. Or the WERR bit in CAN_ERR register is set and WERRIE bit in CAN_INTEN register is set. Or the ERRN bits in CAN_ERR register are set to 1 to 6 (not 0 and not 7) and ERRNIE in CAN_INTEN register is set. This bit is cleared by software when writting 1 to this bit. 0: No error interrupt event 1: Any error interrupt event has happened
1	SLPWS	Sleep working state This bit is set by hardware when the CAN enters sleep working mode after setting SLPWMOD bit in CAN_CTL register. If the CAN leaves normal working mode to sleep working mode, it must wait the current frame transmission or reception to be completed. This bit is cleared by hardware when the CAN leaves sleep working mode. Clear SLPWMOD bit in CAN_CTL register or automatically detect the CAN bus activity when AWU bit is set in CAN_CTL register. If leaving sleep working mode to normal working mode, this bit will be cleared after receiving 11 consecutive recessive bits from the CAN bus. 0: CAN is not in the state of sleep working mode 1: CAN is in the state of sleep working mode
0	IWS	Initial working state This bit is set by hardware when the CAN enters initial working mode after setting IWMOD bit in CAN_CTL register. If the CAN leaves normal working mode to initial working mode, it must wait the current frame transmission or reception to be

completed. This bit is cleared by hardware when the CAN leaves initial working mode after clearing IWMOD bit in CAN_CTL register. If leaving initial working mode to normal working mode, this bit will be cleared after receiving 11 consecutive recessive bits from the CAN bus.

0: CAN is not in the state of initial working mode

1: CAN is in the state of initial working mode

26.4.3. Transmit status register (CAN_TSTAT)

Address offset: 0x08

Reset value: 0x1C00 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
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]		MST2	Reserved			MTE2	MAL2	MTFNERR2	MTF2
r	r	r	r	r	r	r		rs				rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	Reserved			MTE1	MAL1	MTFNERR1	MTF1	MST0	Reserved			MTE0	MAL0	MTFNERR0	MTF0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31	TMLS2	Transmit mailbox 2 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 2 has the last sending order in the Tx FIFO with at least two frames pending.
30	TMLS1	Transmit mailbox 1 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 1 has the last sending order in the Tx FIFO with at least two frames pending.
29	TMLS0	Transmit mailbox 0 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 0 has the last sending order in the Tx FIFO with at least two frames pending.
28	TME2	Transmit mailbox 2 empty 0: Transmit mailbox 2 not empty 1: Transmit mailbox 2 empty
27	TME1	Transmit mailbox 1 empty 0: Transmit mailbox 1 not empty 1: Transmit mailbox 1 empty
26	TME0	Transmit mailbox 0 empty 0: Transmit mailbox 0 not empty 1: Transmit mailbox 0 empty

25:24	NUM[1:0]	These bits are the number of the Tx FIFO mailbox in which the frame will be transmitted if at least one mailbox is empty. These bits are the number of the Tx FIFO mailbox in which the frame will be transmitted at last if all mailboxes are full.
23	MST2	Mailbox 2 stop transmitting This bit is set by the software to stop mailbox 2 transmitting. This bit is reset by the hardware while the mailbox 2 is empty.
22:20	Reserved	Must be kept at reset value.
19	MTE2	Mailbox 2 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writing 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
18	MAL2	Mailbox 2 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writting 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
17	MTFNERR2	Mailbox 2 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 2 transmit finished with error 1: Mailbox 2 transmit finished with no error
16	MTF2	Mailbox 2 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI2 is 1. 0: Mailbox 2 transmit is progressing 1: Mailbox 2 transmit finished
15	MST1	Mailbox 1 stop transmitting This bit is set by software to stop mailbox 1 transmitting. This bit is reset by hardware when the mailbox 1 is empty.
14:12	Reserved	Must be kept at reset value.
11	MTE1	Mailbox 1 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writting 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
10	MAL1	Mailbox 1 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writting 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.

9	MTFNERR1	Mailbox 1 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 1 transmit finished with error 1: Mailbox 1 transmit finished with no error
8	MTF1	Mailbox 1 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI1 is 1. 0: Mailbox 1 transmit is progressing 1: Mailbox 1 transmit finished
7	MST0	Mailbox 0 stop transmitting This bit is set by the software to stop mailbox 0 transmitting. This bit is reset by the hardware when the mailbox 0 is empty.
6:4	Reserved	Must be kept at reset value.
3	MTE0	Mailbox 0 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
2	MAL0	Mailbox 0 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
1	MTFNERR0	Mailbox 0 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 0 transmit finished with error 1: Mailbox 0 transmit finished with no error
0	MTF0	Mailbox 0 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI0 is 1. 0: Mailbox 0 transmit is progressing 1: Mailbox 0 transmit finished

26.4.4. Receive message FIFO0 register (CAN_RFIFO0)

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							RFOVRN0	Reserved		RFD0	RFO0	RFF0	Reserved	RFL0[1:0]	
							r			rs	rc_w1	rc_w1		r	

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	RFOVRN0	Rx FIFO0 overrun This bit is set by hardware when Rx FIFO overwrite enabled and the Rx FIFO0 is overrun. 0: The Rx FIFO0 is not overrun 1: The Rx FIFO0 is overrun **Note:** when this bit is set, don't read data from FIFO.
7:6	Reserved	Must be kept at reset value.
5	RFD0	Rx FIFO0 dequeue This bit is set by software to start dequeuing a frame from Rx FIFO0. This bit is reset by hardware when the dequeuing is done.
4	RFO0	Rx FIFO0 overfull This bit is set by hardware when Rx FIFO0 is overfull and reset by software when writing 1 to this bit. 0: The Rx FIFO0 is not overfull 1: The Rx FIFO0 is overfull
3	RFF0	Rx FIFO0 full This bit is set by hardware when Rx FIFO0 is full and reset by software when writing 1 to this bit. 0: The Rx FIFO0 is not full 1: The Rx FIFO0 is full
2	Reserved	Must be kept at reset value.
1:0	RFL0[1:0]	Rx FIFO0 length These bits are the length of the Rx FIFO0.

26.4.5. Receive message FIFO1 register (CAN_RFIFO1)

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							RFOVRN1	Reserved		RFD1	RFO1	RFF1	Reserved	RFL1[1:0]	
							r			rs	rc_w1	rc_w1		r	

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	RFOVRN1	Rx FIFO1 overrun This bit is set by hardware when Rx FIFO overwrite enabled and the Rx FIFO1 is overrun. 0: The Rx FIFO1 is not overrun 1: The Rx FIFO1 is overrun **Note:** when this bit is set, don't read data from FIFO.
7:6	Reserved	Must be kept at reset value.
5	RFD1	Rx FIFO1 dequeue This bit is set by software to start dequeuing a frame from Rx FIFO1. This bit is reset by hardware when the dequeuing is done.
4	RFO1	Rx FIFO1 overfull This bit is set by hardware when Rx FIFO1 is overfull and reset by writing 1 to this bit. 0: The Rx FIFO1 is not overfull 1: The Rx FIFO1 is overfull
3	RFF1	Rx FIFO1 full This bit is set by hardware when Rx FIFO1 is full and reset by writing 1 to this bit. 0: The Rx FIFO1 is not full 1: The Rx FIFO1 is full
2	Reserved	Must be kept at reset value.
1:0	RFL1[1:0]	Rx FIFO1 length These bits are the length of the Rx FIFO1.

26.4.6. Interrupt enable register (CAN_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														SLPWIE	WIE
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			ERRNIE	BOIE	PERRIE	WERRIE	Reserved	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE

rw rw rw rw rw rw rw rw rw rw rw rw rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	SLPWIE	Sleep working interrupt enable 0: Sleep working interrupt disabled 1: Sleep working interrupt enabled
16	WIE	Wakeup interrupt enable 0: Wakeup interrupt disabled 1: Wakeup interrupt enabled
15	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled
14:12	Reserved	Must be kept at reset value.
11	ERRNIE	Error number interrupt enable 0: Error number interrupt disabled 1: Error number interrupt enabled
10	BOIE	Bus-Off interrupt enable 0: Bus-Off interrupt disabled 1: Bus-Off interrupt enabled
9	PERRIE	Passive error interrupt enable 0: Passive error interrupt disabled 1: Passive error interrupt enabled
8	WERRIE	Warning error interrupt enable 0: Warning error interrupt disabled 1: Warning error interrupt enabled
7	Reserved	Must be kept at reset value.
6	RFOIE1	Rx FIFO1 overfull interrupt enable 0: Rx FIFO1 overfull interrupt disabled 1: Rx FIFO1 overfull interrupt enabled
5	RFFIE1	Rx FIFO1 full interrupt enable 0: Rx FIFO1 full interrupt disabled 1: Rx FIFO1 full interrupt enabled
4	RFNEIE1	Rx FIFO1 not empty interrupt enable 0: Rx FIFO1 not empty interrupt disabled 1: Rx FIFO1 not empty interrupt enabled
3	RFOIE0	Rx FIFO0 overfull interrupt enable 0: Rx FIFO0 overfull interrupt disabled

		1: Rx FIFO0 overfull interrupt enabled
2	RFFIE0	Rx FIFO0 full interrupt enable 0: Rx FIFO0 full interrupt disabled 1: Rx FIFO0 full interrupt enabled
1	RFNEIE0	Rx FIFO0 not empty interrupt enable 0: Rx FIFO0 not empty interrupt disabled 1: Rx FIFO0 not empty interrupt enabled
0	TMEIE	Transmit mailbox empty interrupt enable 0: Transmit mailbox empty interrupt disabled 1: Transmit mailbox empty interrupt enabled

26.4.7. Error register (CAN_ERR)

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
RECNT[7:0]								TECNT[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ERRN[2:0]			Reserved	BOERR	PERR	WERR
									rw				r	r	r

Bits	Fields	Descriptions
31:24	RECNT[7:0]	Receive error count defined by the CAN standard
23:16	TECNT[7:0]	Transmit error count defined by the CAN standard
15:7	Reserved	Must be kept at reset value.
6:4	ERRN[2:0]	Error number These bits indicate the error status of bit transformation. They are updated by hardware. When the bit transformation is successful, they are equal to 0. 000: No error 001: Stuff error 010: Form error 011: Acknowledgment error 100: Bit recessive error 101: Bit dominant error 110: CRC error 111: Set by software
3	Reserved	Must be kept at reset value.

2	BOERR	Bus-Off error Whenever the CAN enters Bus-Off state, the bit will be set by hardware.
1	PERR	Passive error Whenever the TECNT or RECNT is greater than 127, the bit will be set by hardware.
0	WERR	Warning error Whenever the TECNT or RECNT is greater than or equal to 96, the bit will be set by hardware.

26.4.8. Bit timing register (CAN_BT)

Address offset: 0x1C

Reset value: 0x0123 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
SCMOD	LCMOD	Reserved.	SJW[4:0]					Reserved	BS2[2:0]			BS1[3:0]			
rw	rw		rw						rw			rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BS2[4:3]		BS1[6:4]			BAUDPSC[9:0]									
	rw		rw			rw									

Bits	Fields	Descriptions
31	SCMOD	Silent communication mode 0: Silent communication disabled 1: Silent communication enabled
30	LCMOD	Loopback communication mode 0: Loopback communication disabled 1: Loopback communication enabled
29	Reserved	Must be kept at reset value.
28:24	SJW[4:0]	Resynchronization jump width Resynchronization jump width time quantum = SJW[4:0]+1
23	Reserved	Must be kept at reset value.
22:20	BS2[2:0]	Bit segment 2 Bit segment 2 time quantum = BS2[4:0]+1
19:16	BS1[3:0]	Bit segment 1 Bit segment 1 time quantum = BS1[6:0]+1
15	Reserved	Must be kept at reset value.
14:13	BS2[4:3]	Bits 4:3 of BS2 See bits 22:20 of CAN_BT.(Configuration is valid when FDEN=1)

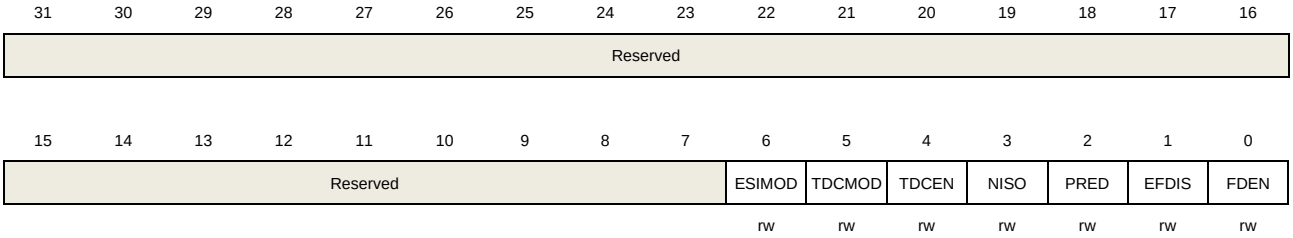
12:10	BS1[6:4]	Bits 6:4 of BS1 See bits 19:16 of CAN_BT. (Configuration is valid when FDEN=1)
9:0	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler

26.4.9. FD control register (CAN_FDCTL)

Address offset: 0x20

Reset value: 0x0000 0002

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



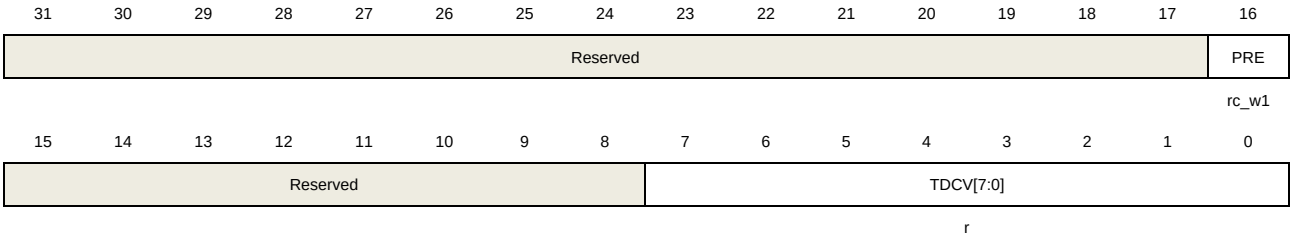
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	ESIMOD	Error state indicator mode 0: Always displays the node error state. Transmit the dominant bit by error active nodes and transmit the recessive bit by error passive nodes. 1: When the node is not in the error passive state: Displays the message buffer error state by configuring ESI bit in CAN_TMPx registers. When the node is in the error passive state: Displays the node error state.
5	TDCMOD	Transmitter delay compensation mode 0: Measurement and offset 1: Only offset
4	TDCEN	Transmitter delay compensation enable 0: Transmitter delay compensation is disabled 1: Transmitter delay compensation is enabled
3	NISO	ISO/Bosch 0: ISO 1: Bosch
2	PRED	Protocol exception event detection disable 0: Protocol exception event detection enabled (to idle) 1: Protocol exception event detection disabled (regarded as a form error)
1	EFDIS	Edge filtering disable

		0: Enable edge filtering 1: Disable edge filtering
0	FDEN	FD operation enable 0: CAN FD function disabled 1: CAN FD function enabled

26.4.10. FD status register (CAN_FDSTAT)

Address offset: 0x24
Reset value: 0x0000 0000

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

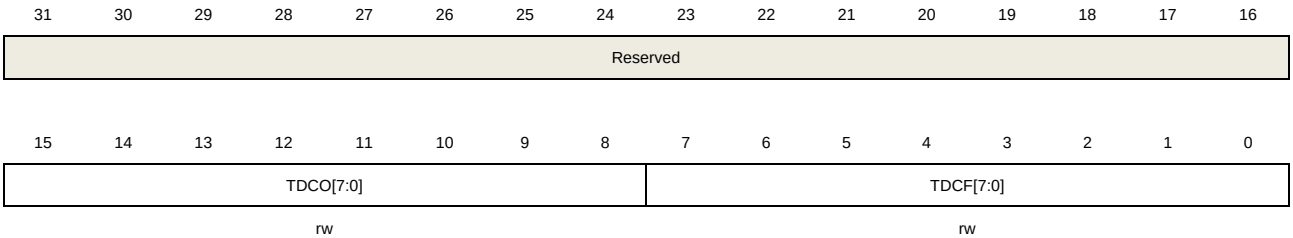


Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	PRE	Protocol exception event This bit is set by hardware when protocol exception event is detected, this bit is cleared by writing 1.
15:8	Reserved	Must be kept at reset value.
7:0	TDCV[7:0]	Transmitter delay compensation value These bits are set by hardware to display transmitter delay compensation value.

26.4.11. FD transmitter delay compensation register (CAN_FDTDC)

Address offset: 0x28
Reset value: 0x0000 0000

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



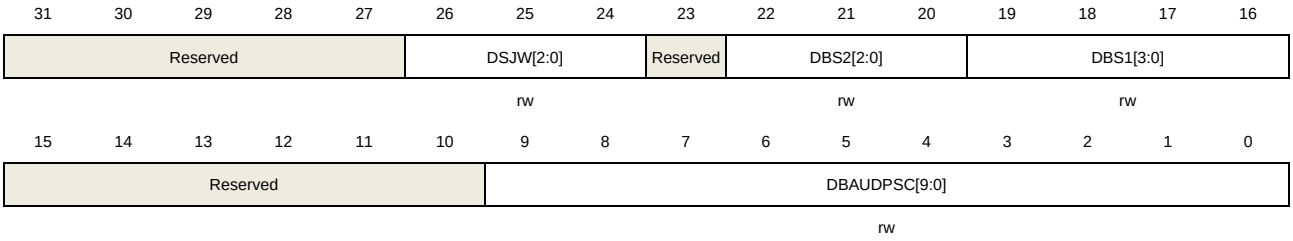
Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:8	TDCO[7:0]	Transmitter delay compensation offset These bits are set to the transmitter delay compensation offset value which defines the distance between the measured delay from CANTX to CANRX and the second sample point.
7:0	TDCF[7:0]	Transmitter delay compensation filter These bits define the minimum value for the SSP position. Dominant edges on CANRX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCV.

26.4.12. Date Bit timing register (CAN_DBT)

Address offset: 0x2C
Reset value: 0x0123 0000

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



Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26:24	DSJW[2:0]	Resynchronization jump width Resynchronization jump width time quantum = DSJW[2:0]+1
23	Reserved	Must be kept at reset value.
22:20	DBS2[2:0]	Bit segment 2 Bit segment 2 time quantum = DBS2[2:0]+1
19:16	DBS1[3:0]	Bit segment 1 Bit segment 1 time quantum = DBS1[3:0]+1
15:10	Reserved	Must be kept at reset value.
9:0	DBAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler

26.4.13. Transmit mailbox identifier register (CAN_TMIx) (x = 0...2)

Address offset: 0x180 + 0x10 * x

Reset value: 0xXXXX XXXX (bit0=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
SFID[10:0]/EFID[28:18]											EFID[17:13]				
rw											rw				

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

Bits	Fields	Descriptions
31:21	SFID[10:0]/EFID[28:18]	The frame identifier SFID[10:0]: Standard format frame identifier EFID[28:18]: Extended format frame identifier
20:16	EFID[17:13]	The frame identifier EFID[17:13]: Extended format frame identifier
15:3	EFID[12:0]	The frame identifier EFID[12:0]: Extended format frame identifier
2	FF	Frame format 0: Standard format frame 1: Extended format frame
1	FT	Frame type 0: Data frame 1: Remote frame
0	TEN	Transmit enable This bit is set by software when one frame will be transmitted and reset by hardware when the transmit mailbox is empty. 0: Transmit disabled 1: Transmit enabled

26.4.14. Transmit mailbox property register (CAN_TMPx) (x = 0…2)

Address offset: 0x184 + 0x10 * x

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
TS[15:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TSEN	FDF	Reserved	BRS	ESI	DLENC[3:0]			
							rw	rw		rw	rw	rw			

Bits	Fields	Descriptions
31:16	TS[15:0]	Time stamp The time stamp of frame in transmit mailbox.
15:9	Reserved	Must be kept at reset value.
8	TSEN	Time stamp enable 0: Time stamp disabled 1: Time stamp enabled. The TS[15:0] will be transmitted in the DB6 and DB7 in DL. This bit is available when the TTC bit in CAN_CTL is set.
7	FDF	CAN FD frame flag 0: Classical frames 1: FD frames
6	Reserved	Must be kept at reset value.
5	BRS	Bit rate of data switch 0: Bit rate not switch 1: The bit rate shall be switched from the nominal bit rate of the arbitration phase to the preconfigured bit rate of data of the data phase
4	ESI	Error status indicator This bit is valid when ESIMOD bit is 1 in CAN_FDCTL register 0: Transmit the dominant bit in ESI phase 1: Transmit the recessive bit in ESI phase
3:0	DLENC[3:0]	Data length code DLENC[3:0] is the number of bytes in a frame.

26.4.15. Transmit mailbox data0 register (CAN_TMDATA0x) (x = 0…2)

Address offset: 0x188 + 0x10 * x
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
DB3[7:0]								DB2[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2

15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

26.4.16. Transmit mailbox data1 register (CAN_TMDATA1x) (x = 0...2)

Address offset: 0x18C + 0x10 * x
Reset value: 0xXXXX XXXX

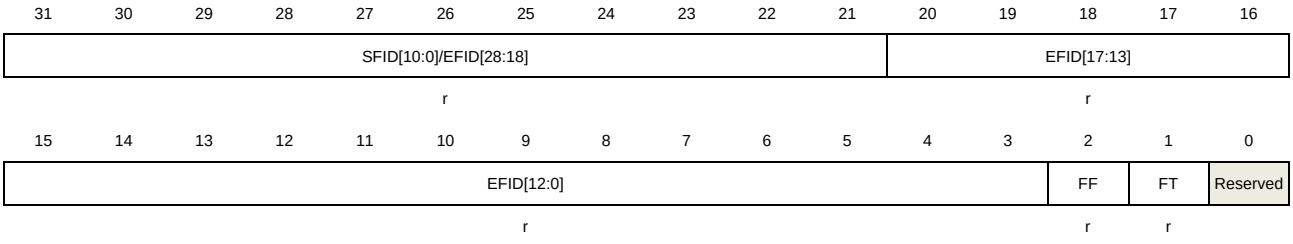
This register has to be accessed by word(32-bit).

Bits	Fields	Descriptions
31:24	DB7[7:0]	Data byte 7
23:16	DB6[7:0]	Data byte 6
15:8	DB5[7:0]	Data byte 5
7:0	DB4[7:0]	Data byte 4

26.4.17. Rx FIFO mailbox identifier register (CAN_RFIFOMIx) (x = 0,1)

Address offset: 0x1B0 + 0x10 * x
Reset value: 0xXXXX XXXX

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



Bits	Fields	Descriptions
31:21	SFID[10:0]/EFID[28:18]	The frame identifier SFID[10:0]: Standard format frame identifier EFID[28:18]: Extended format frame identifier
20:16	EFID[17:13]	The frame identifier EFID[17:13]: Extended format frame identifier

15:3	EFID[12:0]	The frame identifier EFID[12:0]: Extended format frame identifier
2	FF	Frame format 0: Standard format frame 1: Extended format frame
1	FT	Frame type 0: Data frame 1: Remote frame
0	Reserved	Must be kept at reset value.

26.4.18. Rx FIFO mailbox property register (CAN_RFIFOMPx) (x = 0,1)

Address offset: 0x1B4 + 0x10 * x

Reset value: 0xXXXX XXXX

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

Bits	Fields	Descriptions
31:16	TS[15:0]	Time stamp The time stamp of frame in transmit mailbox.
15:8	FI[7:0]	Filtering index The index of the filter which the frame passes.
7	FDF	CAN FD frame flag 0: Classical frames 1: FD frames
6	Reserved	Must be kept at reset value.
5	BRS	Bit rate of data switch 0: Bit rate not switch 1: The bit rate shall be switched from the nominal bit rate of the arbitration phase to the preconfigured bit rate of data of the data phase
4	ESI	Error status indicator 0: Receive the dominant bit in ESI phase 1: Receive the recessive bit in ESI phase
3:0	DLENC[3:0]	Data length code

DLENC[3:0] is the number of bytes in a frame.

26.4.19. Rx FIFO mailbox data0 register (CAN_RFIFOMDATA0x) (x = 0,1)

Address offset: 0x1B8 + 0x10 * x

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
DB3[7:0]								DB2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2
15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

26.4.20. Rx FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x=0,1)

Address offset: 0x1BC, 0x1CC

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
DB7[7:0]								DB6[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DB7[7:0]	Data byte 7
23:16	DB6[7:0]	Data byte 6
15:8	DB5[7:0]	Data byte 5
7:0	DB4[7:0]	Data byte 4

26.4.21. Filter control register (CAN_FCTL)

Address: 0x4001 5A00

Reset value: 0x2A1C 0E01

This register 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		HBC1F[5:0]						Reserved							FLD
		rw													rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	HBC1F[5:0]	Header bank of CAN1 filter These bits are set and cleared by software to define the first bank for CAN1 filter. Bank0 ~ Bank HBC1F-1 is used for CAN0. Bank HBC1F ~ Bank27 is used for CAN1. When set to 0, no bank is used for CAN0. When set to 28, no bank is used for CAN1.
7:1	Reserved	Must be kept at reset value.
0	FLD	Filter lock disable 0: Filter lock enabled 1: Filter lock disabled

26.4.22. Filter mode configuration register (CAN_FMCFG)

Address: 0x4001 5A04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FMOD27	FMOD26	FMOD25	FMOD24	FMOD23	FMOD22	FMOD21	FMOD20	FMOD19	FMOD18	FMOD17	FMOD16
				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
FMOD15	FMOD14	FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
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	FMOD**x**	Filter mode 0: Filter **x** with mask mode

1: Filter **x** with list mode

26.4.23. Filter scale configuration register (CAN_FSCFG)

Address: 0x4001 5A0C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				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
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
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	FS**x**	Filter scale 0: Filter **x** with 16-bit scale 1: Filter **x** with 32-bit scale

26.4.24. Filter associated FIFO register (CAN_FAFIFO)

Address: 0x4001 5A14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16
				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
FAF15	FAF14	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
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	FAF**x**	Filter associated FIFO 0: Filter **x** associated with FIFO0 1: Filter **x** associated with FIFO1

26.4.25. Filter working register (CAN_FW)

Address: 0x4001 5A1C

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				FW27	FW26	FW25	FW24	FW23	FW22	FW21	FW20	FW19	FW18	FW17	FW16
				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
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
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	FW**x**	Filter working 0: Filter **x** working disable 1: Filter **x** working enable

26.4.26. Filter x data y register (CAN_FxDATAy) (x=0..27, y=0,1)

Address: 0x4001 5A40 + 8 * x + 4 * y, (x = 0…27, y=0,1)

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
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FD**x**	Filter data Mask mode 0: Mask match disabled 1: Mask match enabled List mode 0: List identifier bit is 0 1: List identifier bit is 1

27. External memory controller (EXMC)

27.1. Overview

The external memory controller EXMC, is used as a translator for MCU to access a variety of external memory. By configuring the related registers, it can automatically convert AMBA memory access protocol into a specific memory access protocol, such as PSRAM, NOR Flash, NAND Flash. Users can also adjust the timing parameters in the configuration registers to improve memory access efficiency. EXMC access space is divided into multiple banks; each bank is assigned to access a specific memory type with flexible parameter configuration as defined in the control registers.

27.2. Characteristics

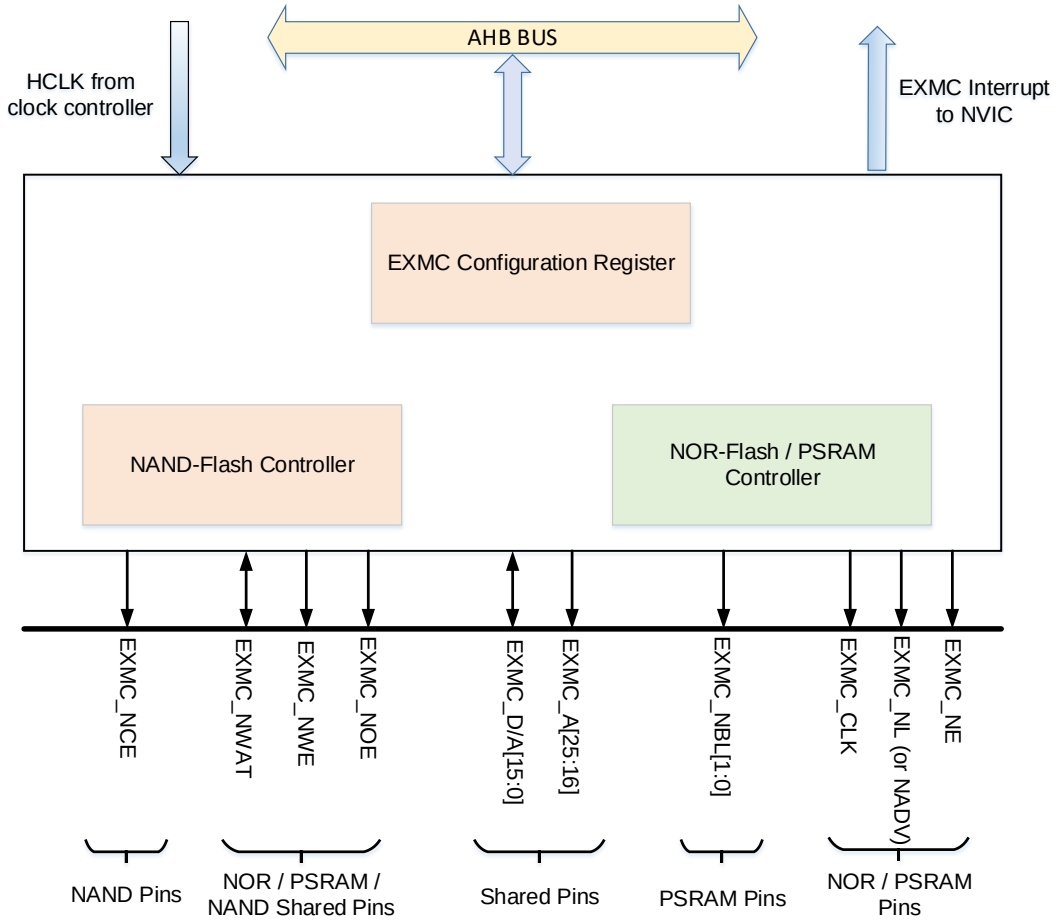
- Supported external memory
  - PSRAM
  - NOR Flash
  - 8-bit or 16-bit NAND Flash
- Protocol translation between the AMBA and the multitude of external memory protocol.
- Offering a variety of programmable timing parameters to meet user's specific needs.
- Each bank has its own chip-select signal which can be configured independently.
- Embedded ECC hardware for NAND Flash access.
- 8-bits ,16-bits or 32-bits bus width.
- Address and data bus multiplexing mechanism for NOR Flash and PSRAM.
- Write enable and byte select are provided as needed.
- Automatic AMBA transaction split when internal and external bus width is not compatible.

27.3. Function overview

27.3.1. Block diagram

EXMC is the combination of five modules: The AHB bus interface, EXMC configuration registers, NOR/PSRAM controller, NAND controller and external device interface. AHB clock (HCLK) is the reference clock.

Figure 27-1. The EXMC block diagram



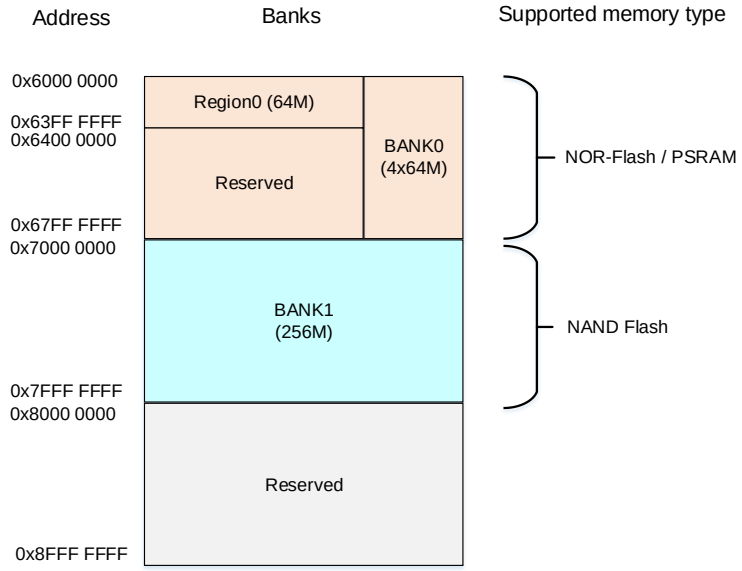
27.3.2. Basic regulation of EXMC access

EXMC is the conversion interface between AHB bus and external device protocol. 32-bit of AHB read/write accesses can be split into several consecutive 8-bit or 16-bit read/write operations respectively. In the process of data transfer, AHB access data width and memory data width may not be the same. In order to ensure consistency of data transmission, EXMC's read/write accesses follows the following basic regulation.

- When the width of AHB bus equals to the memory bus width, no conversion is applied.
- When the width of AHB bus is greater than memory bus width, the AHB accesses will automatically split into several continuous memory accesses.
- When the width of AHB bus is smaller than memory bus width, if the external memory devices have the byte selection function, such as PSRAM, the application can access the corresponding byte through their byte lane EXMC_NBL[1:0]. Otherwise, write operation is prohibited, but read operation is allowed unconditionally.

27.3.3. External device address mapping

Figure 27-2. EXMC memory banks



EXMC access space is divided into multiple banks. Each bank is 256 Mbytes. The first bank (Bank0) is further divided into four regions, and each region is 64 Mbytes. Bank1 is divided into two spaces, the attribute memory space and the common memory space.

Each bank or region has a separate chip-select control signal, which can be configured independently.

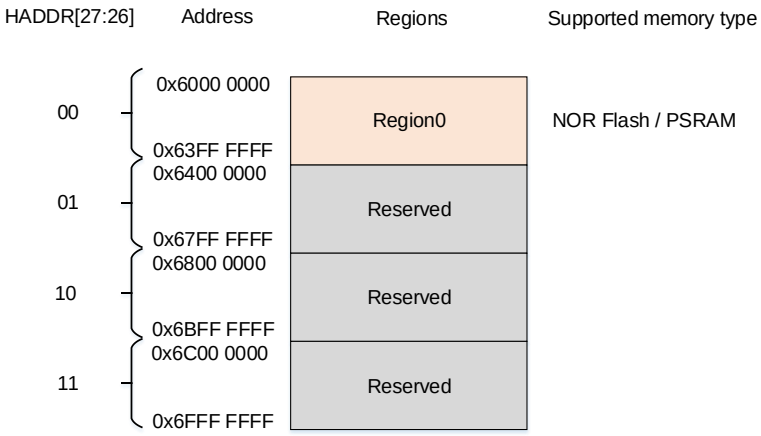
Bank0 is used for NOR-Flash and PSRAM device access.

Bank1 is used to access NAND Flash exclusively.

NOR/PSRAM address mapping

Figure 27-3. Four regions of bank0 address mapping reflects the address mapping of the region of bank0.

Figure 27-3. Four regions of bank0 address mapping



HADDR[25:0] is the byte address whereas the external memory may not be byte accessed,

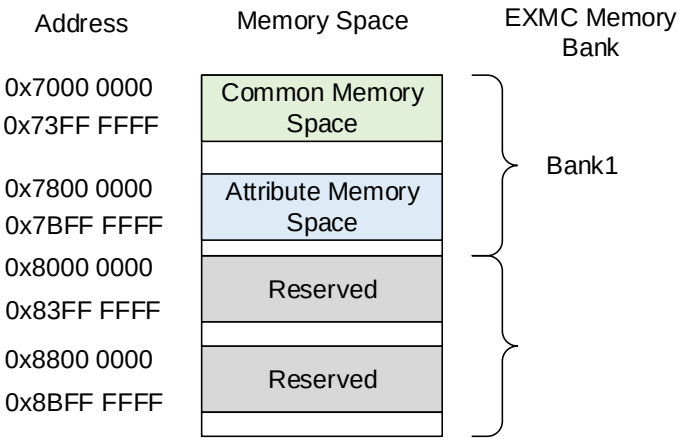
this will lead to address inconsistency. EXMC can adjust HADDR to accommodate the data width of the external memory according to the following rules.

1. When data bus width of the external memory is 8-bits, in this case the memory address is byte aligned. HADDR[25:0] is connected to EXMC_A[25:0] and then the EXMC_A[25:0] is connected to the external memory address lines.
2. When data bus width of the external memory is 16-bits., in this case the memory address is half-word aligned. HADDR byte address must be converted into half-word aligned by connecting HADDR[25:1] with EXMC_A[24:0]. The EXMC_A[24:0] is connected to the external memory address lines.

NAND address mapping

Bank1 is designed to access NAND Flash. Each bank is further divided into several memory spaces as shown in ***Figure 27-4. NAND address mapping***.

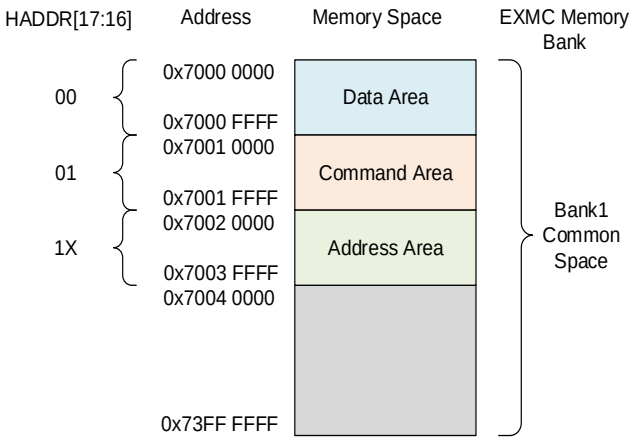
Figure 27-4. NAND address mapping



NAND address mapping

For NAND Flash, the common space and the attribute space are further-divided into three areas individually, the data area, the command area and the address area as shown in ***Figure 27-5. Diagram of bank1 common space***.

Figure 27-5. Diagram of bank1 common space



HADDR[17:16] bits are used to select one of the three areas.

1. When HADDR[17:16] = 00, the data area is selected.
2. When HADDR[17:16] = 01, the command area is selected.
3. When HADDR[17:16] = 1X, the address area is selected.

Application software uses these three areas to access NAND Flash, their definitions are as follows.

1. Address area: This area is where the NAND Flash access address should be issued by software, the EXMC will pull the address latch enable (ALE) signal automatically in address transfer phase. ALE is mapped to EXMC_A[17].
2. Command area: This area is where the NAND Flash access command should be issued by the software, the EXMC will pull the command latch enable (CLE) signal automatically in command transfer phase. CLE is mapped to EXMC_A[16].
3. Data area: This area is where the NAND Flash read/write data should be accessed. When the EXMC is in data transfer mode, software should write the data to be transferred to the NAND Flash in this area. When the EXMC is in data reception mode, software should read the data from the NAND Flash by reading this area. In consecutive mode, the data access address is incremented automatically, and users need not be concerned with the access address area.

27.3.4. NOR/PSRAM controller

NOR/PSRAM memory controller controls bank0, which is designed to support NOR Flash, PSRAM. EXMC has one independent chip-select signals for sub-bank within bank0, named EXMC_NE. Other signals for NOR/PSRAM access are shared. The sub-bank has its own set of configuration register.

Note:

In asynchronous mode, all output signals of controller will change on the rise edge of internal AHB bus clock (HCLK).

In synchronous mode, all output data of controller will change on the fall edge of extern memory device clock (EXMC_CLK).

NOR/PSRAM memory device interface description

Table 27-1. NOR Flash interface signals description

EXMC Pin	Direction	Mode	Functional description
EXMC_CLK	Output	Sync	Clock signal for sync
Muxed EXMC_A[25:16]	Output	Async/Sync	Address bus signal
EXMC_D[15:0]	Input/output	Async/Sync (muxed)	Address/Data bus
EXMC_NE	Output	Async/Sync	Chip selection
EXMC_NOE	Output	Async/Sync	Read enable
EXMC_NWE	Output	Async/Sync	Write enable
EXMC_NWAIT	Input	Async/Sync	Wait input signal
EXMC_NL(NADV)	Output	Async/Sync	Address valid

Table 27-2. PSRAM muxed signal description

EXMC Pin	Direction	Mode	Functional description
EXMC_CLK	Output	Sync	Clock signal for sync
Muxed EXMC_A[25:16]	Output	Async/Sync	Address Bus
EXMC_D[15:0]	Input/output	Async/Sync	Address/Data Bus
EXMC_NE	Output	Async/Sync	Chip selection
EXMC_NOE	Output	Async/Sync	Read enable
EXMC_NWE	Output	Async/Sync	Write enable
EXMC_NWAIT	Input	Async/Sync	Wait input signal
EXMC_NL(NADV)	Output	Async/Sync	Latch enable (address valid enable, NADV)
EXMC_NBL[1]	Output	Async/Sync	Upper byte enable
EXMC_NBL[0]	Output	Async/Sync	Lower byte enable

Supported memory access mode

Table 27-3. EXMC bank 0 supports all transactions lists the access modes supported by EXMC for NOR and PSRAM.

Table 27-3. EXMC bank 0 supports all transactions

Memory	Access Mode	R/W	AHB Transaction Size	Memory Transaction Size	Comments
NOR Flash	Async	R	8	16	
	Async	R	16	16	

Memory	Access Mode	R/W	AHB Transaction Size	Memory Transaction Size	Comments
	Async	W	16	16	
	Async	R	32	16	Split into 2 EXMC accesses
	Async	W	32	16	Split into 2 EXMC accesses
	Sync	R	16	16	
	Sync	R	32	16	
PSRAM	Async	R	8	16	
	Async	W	8	16	Use of byte lanes EXMC_NBL[1:0]
	Async	R	16	16	
	Async	W	16	16	
	Async	R	32	16	Split into 2 EXMC accesses
	Async	W	32	16	Split into 2 EXMC accesses
	Sync	R	16	16	
	Sync	R	32	16	
	Sync	W	8	16	Use of byte lanes EXMC_NBL[1:0]
	Sync	W	16	16	
	Sync	W	32	16	

NOR Flash/PSRAM controller timing

EXMC provides various programmable timing parameters and timing models for PSRAM, NOR Flash and other external static memory.

Table 27-4. NOR / PSRAM controller timing parameters

Parameter	Function	Access mode	Unit	Min	Max
CKDIV	Sync Clock divide ratio	Sync	HCLK	2	16
DLAT	Data latency	Sync	EXMC_CLK	2	17
BUSLAT	Bus latency	Async read/write	HCLK	1	16
DSET	Data setup time	Async	HCLK	2	256
AHLD	Address hold time	Async(muxed)	HCLK	2	16
ASET	Address setup time	Async	HCLK	1	16

Table 27-5. EXMC_timing models

Timing model		Mode description	Write timing parameter	Read timing parameter
Async	Mode AM	NOR Flash address/data mux	DSET AHLD	DSET AHLD

Timing model		Mode description	Write timing parameter	Read timing parameter
			ASET BUSLAT	ASET BUSLAT
Sync	Mode SM	NOR Flash (read) / PSRAM(CRAM) address/data mux	DLAT CKDIV	DLAT CKDIV

As shown in ***Table 27-5. EXMC timing models***, EXMC NOR Flash / PSRAM controller provides a variety of timing model, users can modify those parameters listed in ***Table 27-4. NOR / PSRAM controller timing parameters*** to satisfy different external memory type and user's requirements.

Asynchronous access timing diagram

Mode AM - NOR Flash address / data bus multiplexing

Figure 27-6. Multiplex mode read access

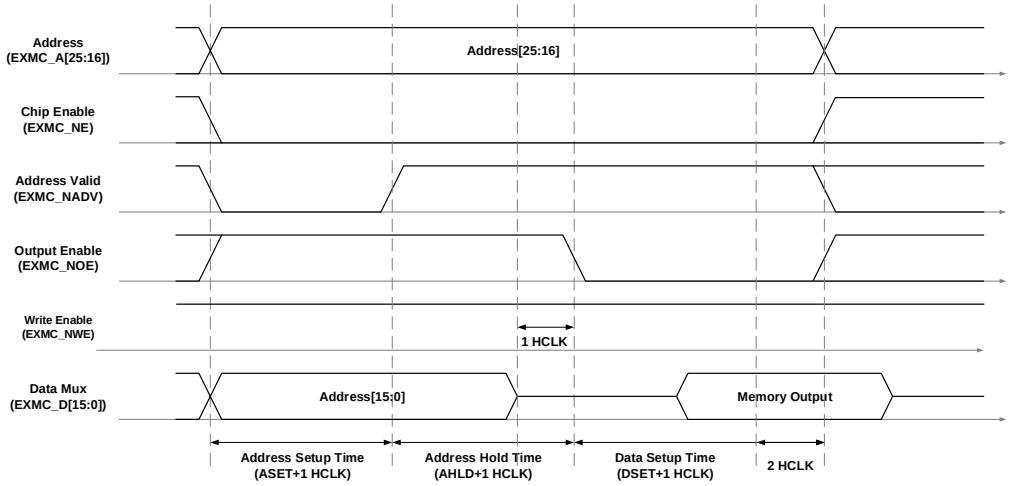


Figure 27-7. Multiplex mode write access

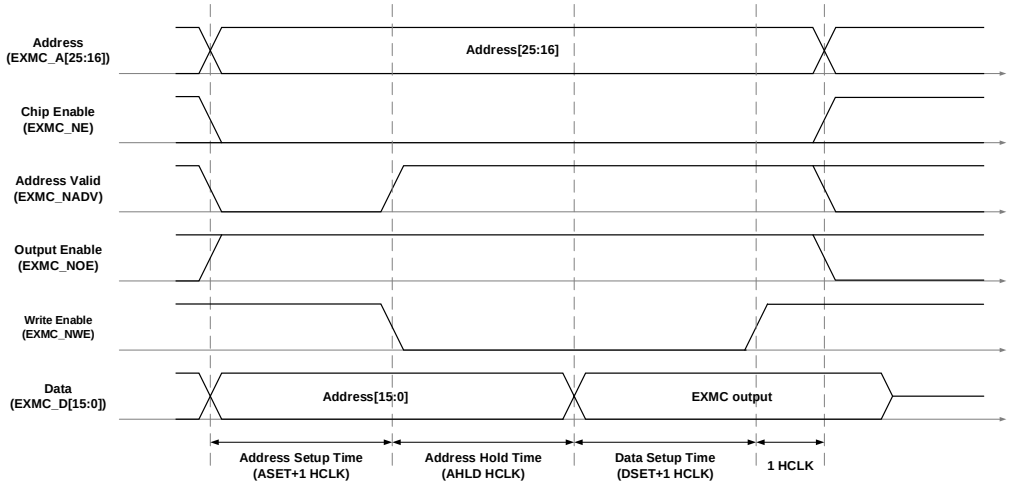


Table 27-6. Multiplex mode related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTL		
31-20	Reserved	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	Depends on memory
14	Reserved	0x0
13	NRWTEN	0x0
12	WEN	Depends on memory
11	NRWTCFG	No effect
10	WRAPEN	0x0
9	NRWTPOL	Meaningful only when the bit 15 is set to 1
8	SBRSTEN	0x0
7	Reserved	0x1
6	NREN	0x1
5-4	NRW	Depends on memory
3-2	NRTP	0x2:NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_PNTCFG		
31-28	Reserved	0x0
27-24	DLAT	No effect
23-20	CKDIV	No effect
19-16	BUSLAT	Minimum time between EXMC_NE rising edge to EXMC_NE falling edge
15-8	DSET	Depends on memory and user (DSET+2 HCLK for write, DSET+3 HCLK for read)
7-4	AHLD	Depends on memory and user
3-0	ASET	Depends on memory and user

Wait timing of asynchronous communication

Wait feature is controlled by the bit ASYNCWAIT in register EXMC_PNCTL. During extern memory access, data setup phase will be automatically extended by the active EXMC_NWAIT signal if ASYNCWAIT bit is set. The extend time is calculated as follows:

If memory wait signal is aligned to EXMC_NOE/ EXMC_NWE:

$$T_{DATA_SETUP} \geq \max T_{WAIT_ASSERTION} + 4HCLK \quad (27\text{-}1)$$

If memory wait signal is aligned to EXMC_NE:

If

$$\max T_{WAIT_ASSERTION} \geq T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (27\text{-}2)$$

be

$$T_{DATA_SETUP} \geq (\max T_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (27\text{-}3)$$

Otherwise

$$T_{DATA_SETUP} \geq 4HCLK \quad (27\text{-}4)$$

Figure 27-8. Read access timing diagram under async-wait signal assertion

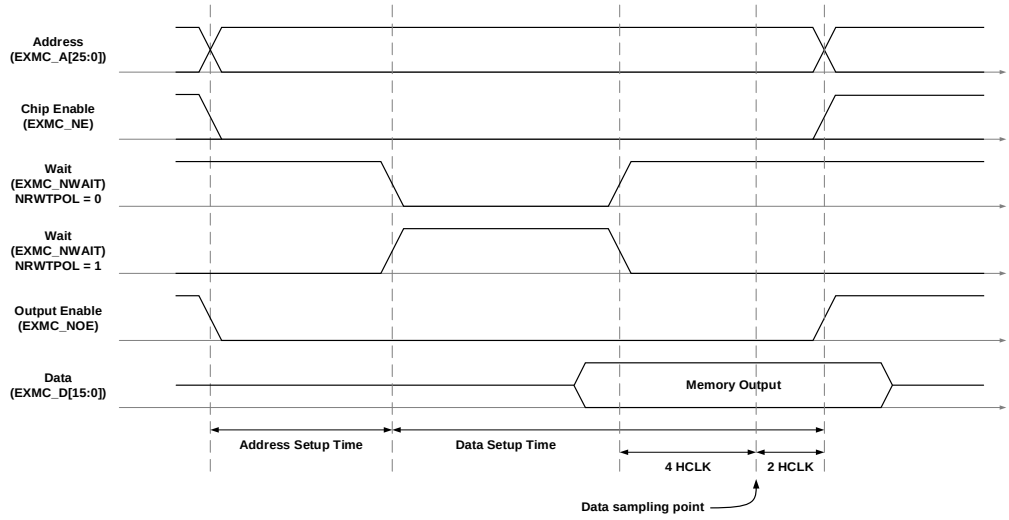
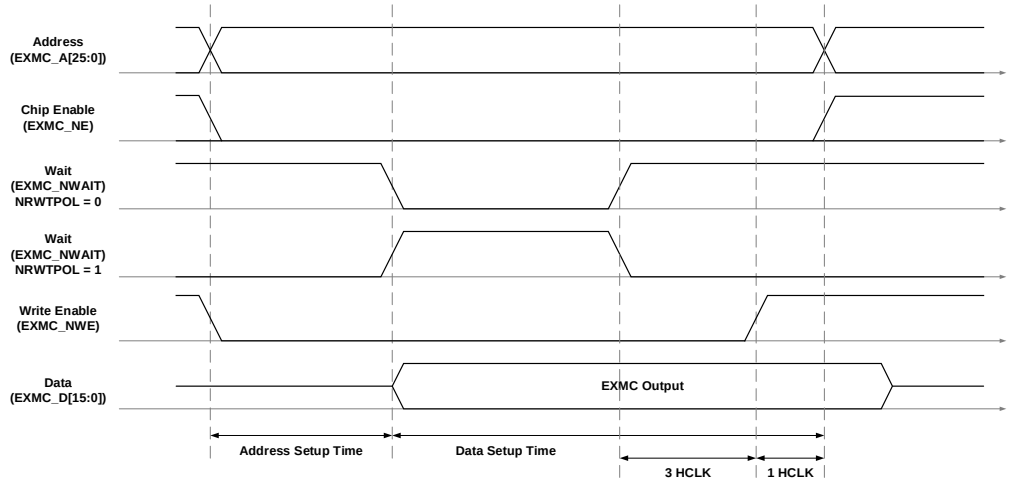


Figure 27-9. Write access timing diagram under async-wait signal assertion



Synchronous access timing diagram

The relations between memory clock (EXMC_CLK) and system clock (HCLK) clock are as follows:

$$EXMC_CLK = \frac{HCLK}{CKDIV+1} \quad (27\text{-}5)$$

CKDIV is the synchronous clock divider ratio, it is configured through the CKDIV control field in the EXMC_PNTCFG register.

1. Data latency and NOR Flash latency

Data latency is the number of EXMC_CLK cycles to wait before sampling the data. The relationship between data latency and NOR Flash specification's latency parameter is as follows:

For NOR Flash's specification excluding the EXMC_NADV cycle, their relationship should be:

NOR Flash latency=DLAT+2 (27-6)

For NOR Flash's specification including the EXMC_NADV cycle, their relationship should be:

NOR Flash latency=DLAT+3 (27-7)

2. Data wait

Users should guarantee that EXMC_NWAIT signal matches that of the external device. This signal is configured through the EXMC_PNCTL registers, it is enabled by the NRWTEN bit, and the active timing could be one data cycle before the wait state or active during the active state by the configuration NRWTCFG bit, while the wait signal's polarity is set by the NRWTPOL bit.

In NOR Flash synchronous burst access mode, when NRWTEN bit in EXMC_PNCTL register is set, EXMC_NWAIT signal will be detected after a period of data latency. If EXMC_NWAIT signal detected is valid, wait cycles will be inserted until EXMC_NWAIT becomes invalid.

- The valid polarity of EXMC_NWAIT:

NRWTPOL= 1: valid level of EXMC_NWAIT signal is high.

NRWTPOL= 0: valid level of EXMC_NWAIT signal is low.

- In synchronous burst mode, EXMC_NWAIT signal has two kinds of configurations:

NRWTCFG = 1: When EXMC_NWAIT signal is active, current cycle data is not valid.

NRWTCFG = 0: When EXMC_NWAIT signal is active, the next cycle data is not valid. It is the default state after reset.

During wait-state inserted via the EXMC_NWAIT signal, the controller continues to send clock pulses to the memory, keep the chip select and output signals availably, and ignore the invalid data signal.

3. Automatic burst split at CRAM page boundary

Crossing page boundary burst access is prohibited in CRAM 1.5, an automatic burst split functionality is implemented by the EXMC. To guarantee correct burst split operation, users should specify CRAM page size by configuring the CPS bit in EXMC_SNCTLx register to inform the EXMC when this functionality should be performed.

4. Mode SM - Single burst transmission

For synchronous burst transmission, if the needed data of AHB is 16-bit, EXMC will perform a burst transmission whose length is 1. If the needed data of AHB is 32-bit, EXMC will make the transmission divided into two 16-bit transmissions, that is, EXMC performs a burst

transmission whose length is 2.

For other configurations please refers to ***Table 27-3. EXMC bank 0 supports all transactions***.

Synchronous mux burst read timing - NOR, PSRAM (CRAM)

Figure 27-10. Read timing of synchronous multiplexed burst mode

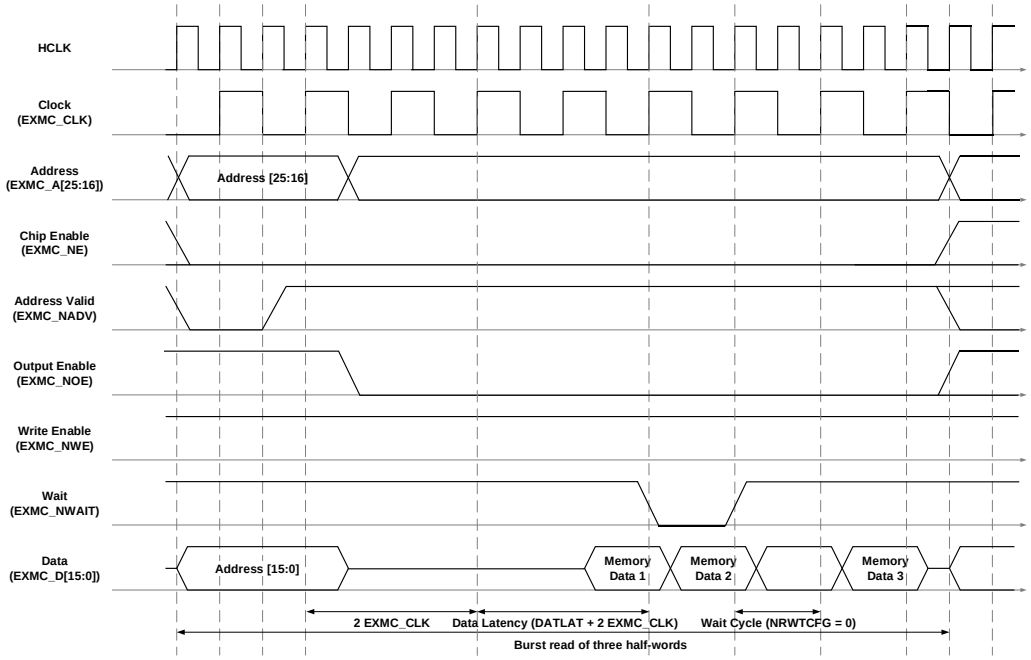


Table 27-7. Timing configurations of synchronous multiplexed read mode

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTL		
31-20	Reserved	0x000
19	SYNCWR	No effect
18-16	CPS	0x0
15	ASYNCWTEN	0x0
14	Reserved	0x0
13	NRWTEN	Depends on memory
12	WEN	No effect
11	NRWTCFG	Depends on memory
10	WRAPEN	0x0
9	NRWTPOL	Depends on memory
8	SBRSTEN	0x1，burst read enable
7	Reserved	0x1
6	NREN	Depends on memory
5-4	NRW	0x1
3-2	NRTP	Depends on memory，0x1/0x2
1	NRMUX	0x1, Depends on memory and users
0	NRBKEN	0x1
EXMC_PNTCFG(Read)		

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTL		
31-28	Reserved	0x0
27-24	DLAT	Data latency
23-20	CKDIV	The figure above：0x1,EXMC_CLK=2HCLK
19-16	BUSLAT	Time between EXMC_NE rising edge to EXMC_NE falling edge
15-8	DSET	No effect
7-4	AHLD	No effect
3-0	ASET	No effect

Mode SM –Synchronous mux burst write timing – PSRAM (CRAM)

Figure 27-11. Write timing of synchronous multiplexed burst mode

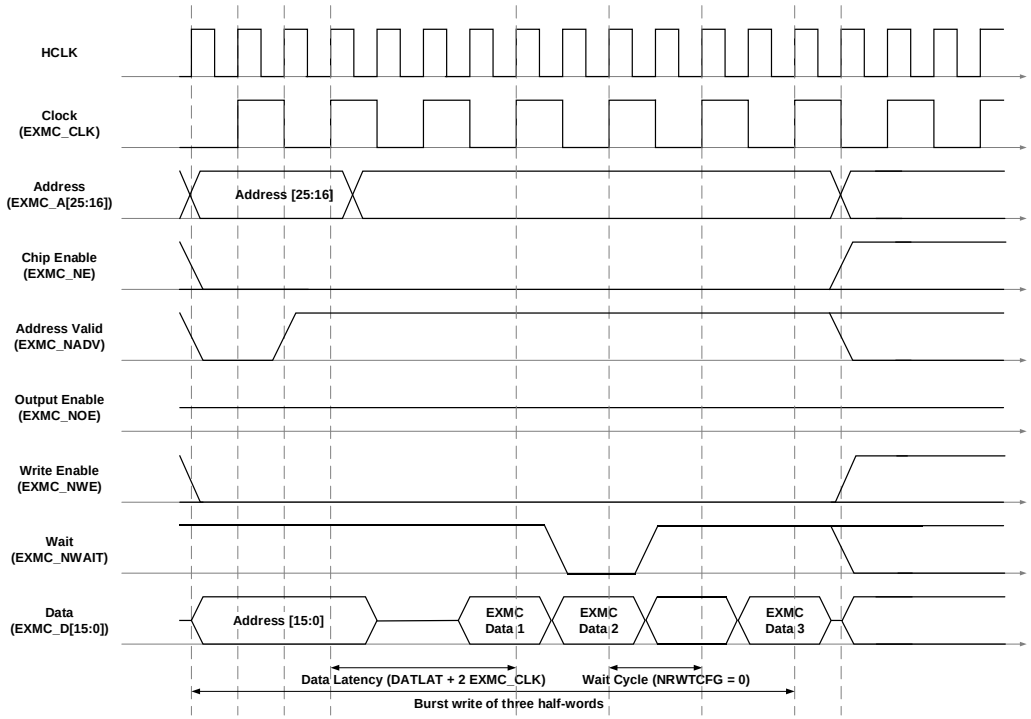


Table 27-8. Timing configurations of synchronous multiplexed write mode

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTL		
31-20	Reserved	0x000
19	SYNCWR	0x1, synchronous write enable
18-16	CPS	0x0
15	AYSNCWAIT	0x0
14	Reserved	0x0
13	NRWTEN	Depends on memory
12	WREN	0x1
11	NRWTCFG	0x0(Here must be zero)
10	WRAPEN	0x0

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTL		
9	NTWTPOL	Depends on memory
8	SBRSTEN	No effect
7	Reserved	0x1
6	NREN	Depends on memory
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1, Depends on users
0	NRBKEN	0x1
EXMC_PNTCFG(Write)		
31-28	Reserved	0x0
27-24	DLAT	Data latency
23-20	CKDIV	The figure above: 0x1,EXMC_CLK=2HCLK
19-16	BUSLAT	Time between EXMC_NE rising edge to EXMC_NE falling edge
15-8	DSET	No effect
7-4	AHLD	No effect
3-0	ASET	No effect

27.3.5. NAND Flash controller

EXMC has partitioned Bank1 as NAND Flash access field, Bank2 is reserved. Bank1 has its own set of control register for access timing configuration. 8- and 16-bit NAND Flash. An ECC hardware is provided for the NAND Flash controller to ensure the robustness of data transfer and storage.

NAND Flash interface function

Table 27-9. 8-bit or 16-bit NAND interface signal

EXMC Pin	Direction	Functional description
EXMC_A[17]	Output	NAND Flash address latch（ALE）
EXMC_A[16]	Output	NAND Flash command latch（CLE）
EXMC_D[7:0]/ EXMC_D[15:0]	Input /Output	8-bit multiplexed, bidirectional address/data bus
		16-bit multiplexed, bidirectional address/data bus
EXMC_NCE	Output	Chip select
EXMC_NOE(NRE)	Output	Output enable
EXMC_NWE	Output	Write enable
EXMC_NWAIT	Input	NAND Flash ready/busy input signal to the EXMC

Supported memory access mode

Table 27-10. Bank1 of EXMC support the memory and access mode

Memory	Mode	R/W	AHB transaction size	Comments
8-bit NAND	Async	R	8	
	Async	W	8	
	Async	R	16	Automatically split into 2 EXMC accesses
	Async	W	16	
	Async	R	32	Automatically split into 4 EXMC accesses
	Async	W	32	
16-bit NAND	Async	R	8	
	Async	W	8	Not support this operation
	Async	R	16	
	Async	W	16	
	Async	R	32	Automatically split into 2 EXMC accesses
	Async	W	32	

NAND Flash controller timing

EXMC can generate the appropriate signal timing for NAND Flash device. Each bank has a corresponding register to manage and control the external memory, such as EXMC_NCTL, EXMC_NSTAT, EXMC_NCTCFG, EXMC_NATCFG and EXMC_NECC. Among these registers, EXMC_NCTCFG, EXMC_NATCFG registers contain four timing parameters individually which are configured according to user specification and features of the external memory.

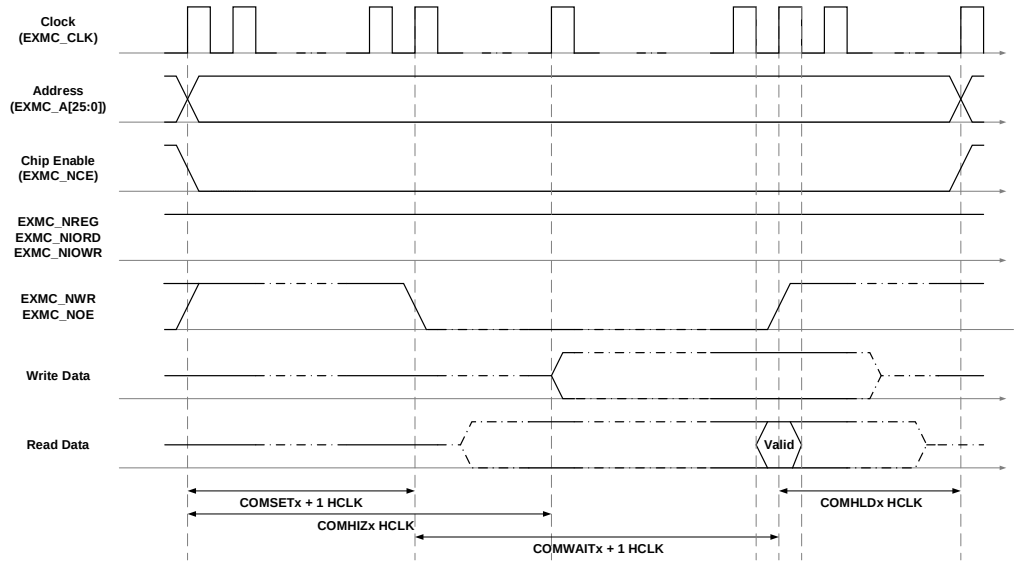
Table 27-11. NAND Flash programmable parameters

Programmable parameter	W/R	Unit	Functional description	NAND Flash	
				Min	Max
High impedance time of the memory data bus (HIZ)	W/R	HCLK	Time to keep the data bus high impedance after starting write operation	0	255
Memory hold time (HLD)	W/R	HCLK	The number of HCLK clock cycles to keep address valid after sending the command. In write mode, it is also data hold time.	1	254
Memory wait time (WAIT)	W/R	HCLK	Minimum duration of sending command	2	256
Memory setup time (SET)	W/R	HCLK	The number of HCLK clock cycles to build address before sending command	1	255

The shows ***Figure 27-12. Access timing of common memory space of NAND flash Controller*** the programmable parameters which are defined in the common memory space

operations. The programmable parameters of Attribute memory space are defined as well.

Figure 27-12. Access timing of common memory space of NAND flash Controller



NAND Flash operation

When EXMC sends command or address to NAND Flash, it needs to use the command latch signal (EXMC_A[16]) or address latch signal (EXMC_A[17]), namely, the CPU needs to perform write operation in particular address.

Example: NAND Flash read operation steps:

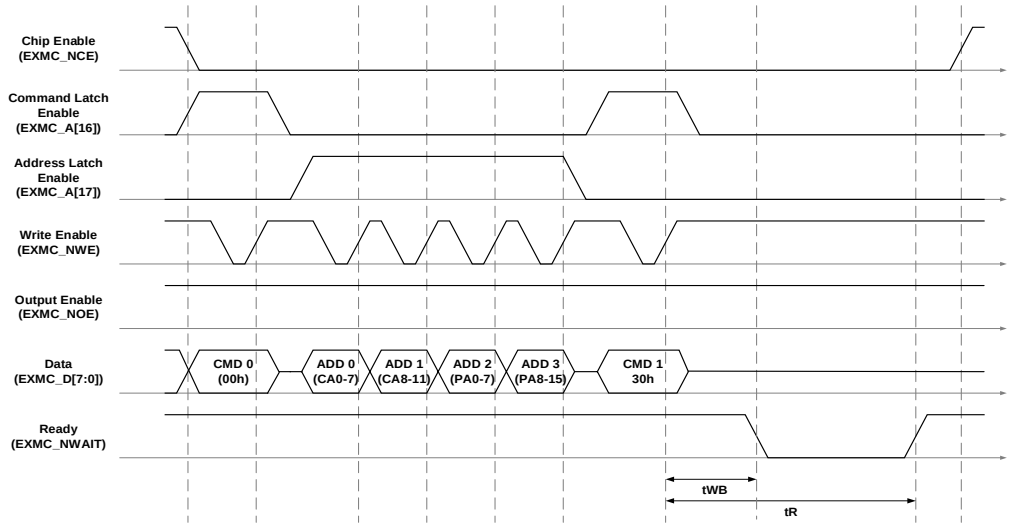
1. Configure EXMC_NCTL1 and EXMC_NCTCFG1 register. When pre-waiting is needed, EXMC_NATCFG1 has to be configured.
2. Send the command of NAND Flash read operation to the common space. Namely, during the valid period of EXMC_NCE and EXMC_NWE, when EXMC_CLE (EXMC_A[16]) becomes valid (high level), data on the I/O pins is regarded as a command by NAND Flash.
3. Send the start address of read operation to the common space. During the valid period of EXMC_NCE and EXMC_NWE, when EXMC_ALE (EXMC_A[17]) becomes valid (high level), the data on the I/O pins is regarded as an address by NAND Flash.
4. Waiting for NAND ready signal. In this period, NAND controller will maintain EXMC_NCE valid.
5. Read data byte by byte from the data area of the common space.
6. If new commands or address haven't been written, data of the next page can be read out automatically. You can also read the data of the next page by going to step 3 and then writing a new address or writing a new command and address in step 2.

NAND Flash pre-wait functionality

Some NAND Flash requires that the controller should wait for NAND Flash to be busy after the first command byte following the address bytes is send, and some EXMC_NCE-sensitive NAND Flash also requires that the EXMC_NCE must remain valid before it is ready.

Taking TOSHIBA128 M x 8 bit NAND Flash as an example:

Figure 27-13. Access to none "NCE don't care" NAND Flash



1. Write CMD0 into NAND Flash bank common space command area.
2. Write ADD0 into NAND Flash bank common space address area.
3. Write ADD1 into NAND Flash bank common space address area.
4. Write ADD2 into NAND Flash bank common space address area.
5. Write ADD3 into NAND Flash bank common space address area.
6. Write CMD1 into NAND Flash bank attribute space command area.

In step 6, EXMC uses the operation timing defined in EXMC_NATCFG register. After a period of ATTHLD, NAND Flash waits for EXMC_NWAIT signal to be busy, and the time period of ATTHLD should be greater than tWB (t_{WB} is defined as the time from EXMC_NWE high to EXMC_NWAIT low). For NCE-sensitive NAND Flash, after the first command byte following address bytes has been entered, EXMC_NCE must remain low until EXMC_NWAIT goes from low to high. The ATTHLD value of attribute space can be set in EXMC_NATCFG register to meet the timing requirements of tWB. CPU can use the attribute space timing when writing the first command byte following address bytes to the NAND Flash device. In other times, the MCU must use the common space timing.

NAND Flash ECC calculation module

An ECC calculation hardware is implemented in bank1 respectively. Users can choose page size according to the ECCSZ control field in the EXMC_NCTL register. ECC offers one-bit error correction and two bits errors detection.

When NAND memory block is enabled, ECC module will detect EXMC_D[15:0], EXMC_NCE and EXMC_NWE signals. When a data size of ECCSZ has been read or written, software must read the calculated ECC in theEXMC_NECC register. When a recalculation of ECC is needed, software must clear the EXMC_NECC register value by resetting ECCEN bit of EXMC_NCTL register to zero, and then restart ECC calculation by setting the ECCEN bit of EXMC_NCTL to 1.

27.4. Registers definition

27.4.1. NOR/PSRAM controller registers

PSRAM/NOR Flash control registers (EXMC_PNCTL)

Address offset: 0x00

Reset value: 0x0000 30DB for region0.

This register 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												SYNCWR	CPS[2:0]		
												rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	Reserved	NRWTEN	WREN	NRWTCFG	WRAPEN	NRWTPOL	SBRSTEN	Reserved	NREN	NRW[1:0]		NRTP[1:0]		NRMUX	NRBKEN
rw		rw	rw	rw	rw	rw	rw		rw	rw		rw		rw	rw

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19	SYNCWR	Synchronous write 0: Asynchronous write 1: Synchronous write
18:16	CPS[2:0]	CRAM page size 000: Automatic burst split on page boundary crossing 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: Reserved
15	ASYNCWAIT	Asynchronous wait 0: Disable the asynchronous wait function 1: Enable the asynchronous wait function
14	Reserved	Must be kept at reset value.
13	NRWTEN	NWAIT signal enable For Flash memory access in burst mode, this bit enables/disables wait-state insertion via the NWAIT signal: 0: Disable NWAI signal 1: Enable NWAIT signal
12	WREN	Write enable

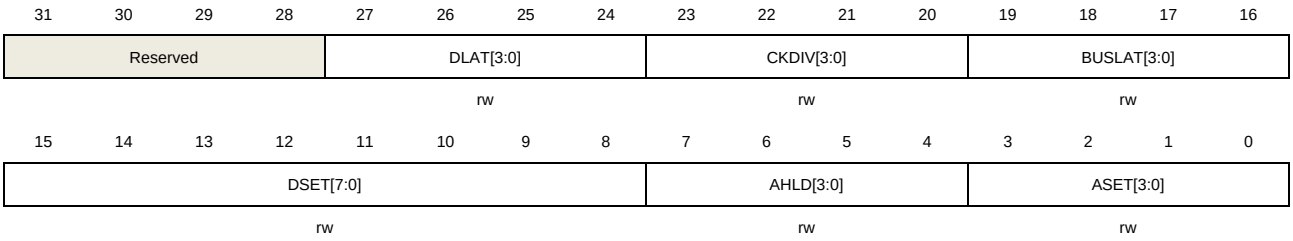
		0: Disabled write in the bank by the EXMC, otherwise an AHB error is reported 1: Enabled write in the bank by the EXMC (default after reset)
11	NRWTCFG	NWAIT signal configuration, only work in synchronous mode 0: NWAIT signal is active one data cycle before wait state 1: NWAIT signal is active during wait state
10	WRAPEN	Wrapped burst mode enable 0: Disable wrap burst mode support 1: Enable wrap burst mode support
9	NRWTPOL	NWAIT signal polarity 0: Low level is active of NWAIT 1: High level is active of NWAIT
8	SBRSTEN	Synchronous burst enable 0: Disable burst access mode 1: Enable burst access mode
7	Reserved	Must be kept at reset value.
6	NREN	NOR Flash access enable 0: Disable NOR Flash access 1: Enable NOR Flash access
5:4	NRW[1:0]	NOR region memory data bus width 00: 8 bits 01: 16 bits(default after reset) 10/11: Reserved
3:2	NRTP[1:0]	NOR region memory type 00: Reserved 01: PSRAM（CRAM） 10: NOR Flash(default after reset for region0) 11: Reserved
1	NRMUX	NOR region memory address/data multiplexing 0: Disable address/data multiplexing function 1: Enable address/data multiplexing function
0	NRBKEN	NOR region enable 0: Disable the corresponding memory bank 1: Enable the corresponding memory bank

PSRAM/NOR Flash timing configuration registers (EXMC_PNTCFG)

Address offset: 0x04

Reset value: 0x0FFF FFFF

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	DLAT[3:0]	Data latency for NOR Flash. Only valid in synchronous access 0x0: Data latency of first burst access is 2 EXMC_CLK 0x1: Data latency of first burst access is 3 EXMC_CLK …… 0xF: Data latency of first burst access is 17 EXMC_CLK
23:20	CKDIV[3:0]	Synchronous clock divide ratio. This filed is only effect in synchronous mode. 0x0: Reserved 0x1: EXMC_CLK period = 2 * HCLK period …… 0xF: EXMC_CLK period = 16 * HCLK period
19:16	BUSLAT[3:0]	Bus latency The bits are defined in asynchronous mode in order to avoid bus contention, and represent the data bus to return to a high impedance state's minimum. 0x0: Bus latency = 1 * HCLK period 0x1: Bus latency = 2 * HCLK period …… 0xF: Bus latency = 16 * HCLK period
15:8	DSET[7:0]	Data setup time This field is meaningful only in asynchronous access. 0x00: Reserved 0x01: Data setup time = 2 * HCLK period …… 0xFF: Data setup time = 256 * HCLK period
7:4	AHLD[3:0]	Address hold time This field is used to set the time of address hold phase, which only used in multiplexed mode. 0x0: Reserved 0x1: Address hold time = 2 * HCLK …… 0xF: Address hold time = 16 * HCLK
3:0	ASET[3:0]	Address setup time This field is used to set the time of address setup phase.

Note: meaningful only in asynchronous access of NOR Flash

0x0: Address setup time = 1 * HCLK

……

0xF: Address setup time = 16 * HCLK

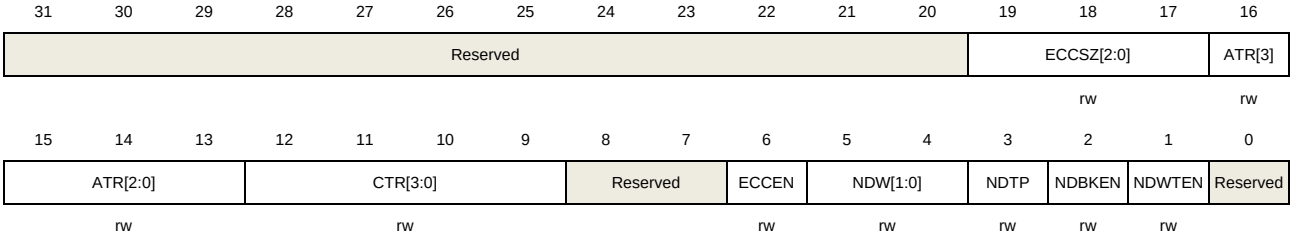
27.4.2. NAND Flash controller registers

NAND Flash control registers (EXMC_NCTL)

Address offset: 0x60

Reset value: 0x0000 0018

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



Bits	Fields	Description
31:20	Reserved	Must be kept at reset value.
19:17	ECCSZ[2:0]	ECC size 000: 256 bytes 001: 512 bytes 010: 1024 bytes 011: 2048 bytes 100: 4096 bytes 101: 8192 bytes
16:13	ATR[3:0]	ALE to RE delay 0x0: ALE to RE delay = 1 * HCLK …… 0xF: ALE to RE delay = 16 * HCLK
12:9	CTR[3:0]	CLE to RE delay 0x0: CLE to RE delay = 1 * HCLK 0x1: CLE to RE delay = 2 * HCLK …… 0xF: CLE to RE delay = 16 * HCLK
8:7	Reserved	Must be kept at reset value.
6	ECCEN	ECC enable 0: Disable ECC, and reset EXMC_NECCx

		1: Enable ECC
5:4	NDW[1:0]	NAND bank memory data bus width 00: 8 bits 01: 16 bits Others: Reserved
3	NDTP	NAND bank memory type 0: Reserved 1: NAND Flash
2	NDBKEN	NAND bank enable 0: Disable corresponding memory bank 1: Enable corresponding memory bank
1	NDWTEN	Wait function enable 0: Disable wait function 1: Enable wait function
0	Reserved	Must be kept at reset value.

NAND Flash state registers (EXMC_NSTAT)

Address offset: 0x64

Reset value: 0x0000 0042 (for bank1)

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

This register contains a FIFO empty status bit, design specifically for ECC purpose. When external memory write is performed, the FIFO can hold up to 2 word from AHB access, freeing the bus temporarily for other peripherals. ECC calculation is based on the data passing through the FIFO, for correct ECC, users should read the ECC register only after the FIFO empty status flag is raised.

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									FFEPT	Reserved					
									r						

Bits	Fields	Description
31:7	Reserved	Must be kept at reset value.
6	FFEPT	FIFO empty flag 0: FIFO is not empty. 1: FIFO is empty.
5:0	Reserved	Must be kept at reset value.

NAND Flash common space timing configuration registers (EXMC_NCTCFG)

Address offset: 0x68

Reset value: 0xFCFC FCFC

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

These operations applicable to common memory space for NAND Flash.

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

Bits	Fields	Description
31:24	COMHIZ[7:0]	Common memory data bus HiZ time The bits are defined as time of bus keep high impedance state after writing the data. 0x00: COMHIZ = 1 * HCLK …… 0xFE: COMHIZ = 255 * HCLK 0xFF: Reserved
23:16	COMHLD[7:0]	Common memory hold time After sending the address, the bits are defined as the address hold time. In write operation, they are also defined as the data signal hold time. 0x00: Reserved 0x01: COMHLD = 1 * HCLK …… 0xFE: COMHLD = 254 * HCLK 0xFF: Reserved
15:8	COMWAIT[7:0]	Common memory wait time Define the minimum time to maintain command 0x00: Reserved 0x01: COMWAIT = 2 * HCLK (+NWAIT active cycles) …… 0xFE: COMWAIT = 255 * HCLK (+NWAIT active cycles) 0xFF: Reserved
7:0	COMSET[7:0]	Common memory setup time Define the time to build address before sending command 0x00: COMSET = 1 * HCLK …… 0xFE: COMSET = 255 * HCLK 0xFF: Reserved

NAND Flash attribute space timing configuration registers (EXMC_NATCFG)

Address offset: 0x6C

Reset value: 0xFCFC FCFC

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

It is used for 8-bit accesses to the attribute memory space of the NAND Flash for the last address or command write access if another timing must be applied.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTHIZ[7:0]								ATTHLD[7:0]							
rw								rw							

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

Bits	Fields	Description
31:24	ATTHIZ[7:0]	Attribute memory data bus HiZ time The bits are defined as time of bus keep high impedance state after writing the data. 0x00: ATTHIZ = 1 * HCLK …… 0xFE: ATTHIZ = 255 * HCLK 0xFF: Reserved
23:16	ATTHLD[7:0]	Attribute memory hold time After sending the address, the bits are defined as the address hold time. In write operation, they are also defined as the data signal hold time. 0x00: Reserved 0x01: ATTHLD = 1 * HCLK …… 0xFE: ATTHLD = 254 * HCLK 0xFF: Reserved
15:8	ATTWAIT[7:0]	Attribute memory wait time Define the minimum time to maintain command 0x00: Reserved 0x01: ATTWAIT = 2 * HCLK (+NWAIT active cycles) …… 0xFE: ATTWAIT = 255 * HCLK (+NWAIT active cycles) 0xFF: ATTWAIT = Reserved
7:0	ATTSET[7:0]	Attribute memory setup time Define the time to build address before sending command 0x00: ATTSET = 1 * HCLK …… 0xFE: ATTSET = 255 * HCLK

0xFF: Reserved

NAND Flash ECC registers (EXMC_NECC)

Address offset: 0x74

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
ECC[31:16]															
r															

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

Bits	Fields	Description
31:0	ECC[31:0]	ECC result

ECCSZ[2:0]	NAND Flash page size	ECC bits
0b000	256	ECC[21:0]
0b001	512	ECC[23:0]
0b010	1024	ECC[25:0]
0b011	2048	ECC[27:0]
0b100	4096	ECC[29:0]
0b101	8192	ECC[31:0]

28. Universal serial bus full-speed interface (USBFS)

28.1. Overview

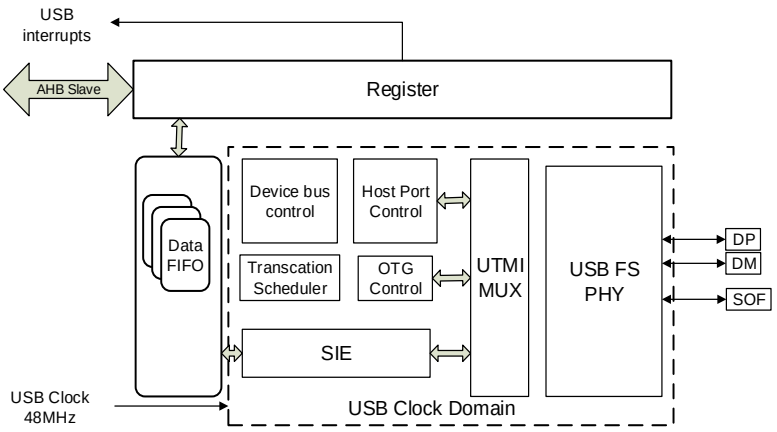
USB Full-Speed (USBFS) controller provides a USB-connection solution for portable devices. USBFS supports host and device modes. USBFS contains a full-speed internal USB PHY and external PHY chip is not contained. USBFS supports all the four types of transfer (control, bulk, Interrupt and isochronous) which defined in USB 2.0 protocol.

28.2. Characteristics

- Supports USB 2.0 host mode at Full-Speed(12Mb/s) or Low-Speed(1.5Mb/s)
- Supports USB 2.0 device mode at Full-Speed(12Mb/s)
- Supports all the 4 types of transfer: control, bulk, interrupt and isochronous
- Includes a USB transaction scheduler in host mode to handle USB transaction request efficiently.
- Includes a 1.25KB FIFO RAM.
- Supports 8 channels in host mode.
- Includes 2 transmit FIFOs (periodic and non-periodic) and a receive FIFO (shared by all channels) in host mode.
- Includes 4 transmit FIFOs (one for each IN endpoint) and a receive FIFO (shared by all OUT endpoints) in device mode.
- Supports 4 OUT and 4 IN endpoints in device mode.
- Supports remote wakeup in device mode.
- Includes a Full-Speed USB PHY.
- Time intervals of SOFs is dynamic adjustable in host mode.
- SOF pulse supports output to pad.
- Needs external component to supply power for connected USB device in host mode.

28.3. Block diagram

Figure 28-1. USBFS block diagram



28.4. Signal description

Table 28-1. USBFS signal description

I/O port	Type	Description
DM	Input/Output	Differential D- port
DP	Input/Output	Differential D+ port
SOF	Output	USB SOF signal output

28.5. Function overview

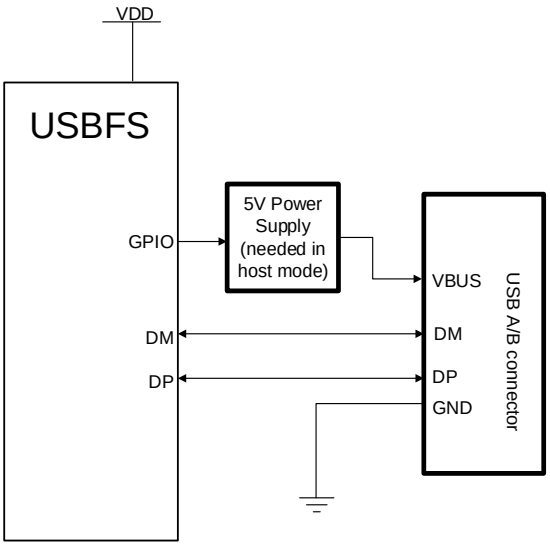
28.5.1. USBFS clocks and working modes

USBFS can operate as a host, a device or a DRD (Dual Role Device), it contains an internal full-speed PHY.The maximum speed supported by USBFS is full-speed.

The internal PHY supports Full-Speed and Low-Speed in host mode, supports Full-speed in device mode. The USB clock used by the USBFS should be 48MHz. The 48MHz USB clock is generated from internal clocks in system, and its source and divider factors are configurable in RCU.

The pull-up and pull-down resistors have already been integrated into the internal PHY and they could be controlled by USBFS automatically according to the current mode (host, device mode) and connection status. A typical connection is shown in ***Figure 28-2. Connection with host or device mode***

Figure 28-2. Connection with host or device mode



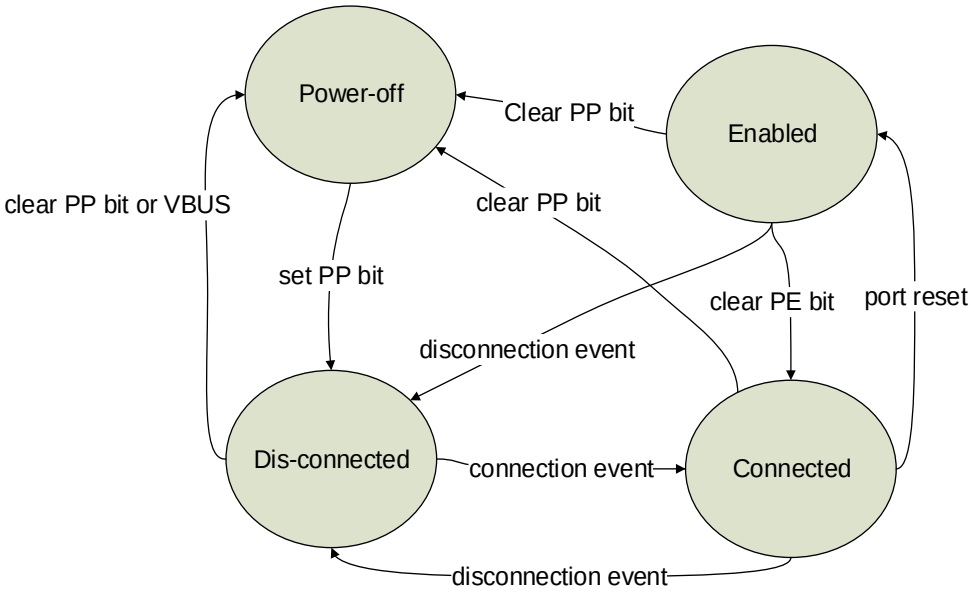
When USBFS works in host mode (FHM bit is set and FDM bit is cleared), the VBUS is 5V power detecting pin used for voltage detection defined in USB protocol. The internal PHY cannot supply 5V VBUS power, so if application needs VBUS power, an external power supply IC is needed. The VBUS connection between USBFS and the USB connector can be omitted in host mode, so USBFS doesn't detect the voltage level on VBUS pin and always assumes that the 5V power is present.

28.5.2. USB host function

USB Host Port State

Host application may control state of the USB port via USBFS_HPCS register. After system initialization, the USB port stays at power-off state. After PP bit is set by software, the internal USB PHY is powered on, and the USB port changes into disconnected state. After a connection is detected, USB port changes into connected state. The USB port changes into enabled state after a port reset is performed on USB bus.

Figure 28-3. State transition diagram of host port



Connection, Reset and Speed identification

As a USB host, USBFS will trigger a connection flag for application after a connection is detected and will trigger a disconnection flag after a disconnection event.

PRST bit is used for USB reset sequence. Application may set this bit to start a USB reset and clear this bit to finish the USB reset. This bit only takes effect when port is at connected or enabled state.

The USBFS performs speed identification during connection, and the speed information will be reported in PS filed in USBFS_HPCS register. USBFS identifies the device speed by the voltage level of DM or DP. As described in USB protocol, full-speed device pulls up DP line while low-speed device pulls up DM line.

Suspend and resume

USBFS supports suspend state and resume operation. When USBFS port is at enabled state, writing 1 to PSP bit in USBFS_HPCS register will cause USBFS to enter suspend state. In suspend state, USBFS stops sending SOFs on USB bus and this will cause the connected USB device to enter suspend state after 3ms. Application can set the PREM bit in USBFS_HPCS register to start a resume sequence to wake up the suspended device and clear this bit to stop the resume sequence. The WKUPIF bit in USBFS_GINTF will be set and the USBFS wake up interrupt will be triggered if a host in suspend state detects a remote wakeup signal.

SOF generate

USBFS sends SOF tokens on USB bus in host mode. As described in USB 2.0 protocol, SOF packets are generated (by the host controller or hub transaction translator) every 1ms in full-speed links.

Each time after USBFS enters into enabled state, it will send the SOF packet periodically

which the time is defined in USB 2.0 protocol. In addition, application may adjust the length of a frame by writing FRI filed in USBFS_HFT registers. The FRI bits define the number of USB clock cycles in a frame, so its value should be calculated based on the frequency of USB clock which is used by USBFS. The FRT filed bits show that the remaining clock cycles of the current frame and stop changing during suspend state.

USBFS is able to generate a pulse signal each SOF packet and output it to a pin. The pulse length is 12 HCLK cycle. If application desires to use this function, it needs to set SOFOEN bit in USBFS_GCCFG register and configure the related pin registers in GPIO.

USB Channels and Transactions

USBFS includes 8 independent channels in host mode. Each channel is able to communicate with an endpoint in USB device. The transfer type, direction, packet length and other information are all configured in channel related registers such as USBFS_HCHxCTL and USBFS_HCHxLEN.

USBFS supports all the four kinds of transfer types: control, bulk, interrupts and isochronous. USB 2.0 protocol divides these transfers into 2 kinds: non-periodic transfer (control and bulk) and periodic transfer (interrupt and isochronous). Based on this, USBFS includes two request queues: periodic request queue and non-periodic request queue, to perform efficient transaction schedule. A request entry in a request queue described above may represent a USB transaction request or a channel operation request.

Application needs to write packet into data FIFO via AHB register interface if it wants to start an OUT transaction on USB bus. USBFS hardware will automatically generate a transaction request entry in request queue after the application writes a whole packet.

The request entries in request queue are processed in order by transaction control module. USBFS always tries to process periodic request queue firstly and secondly process non-periodic request queue.

After a start of frame, USBFS begins to process periodic queue until the queue is empty or bus time required by the current periodic request is not enough, and then process the non-periodic queue. This strategy ensures the bandwidth of periodic transactions in a frame. Each time the USBFS reads and pops a request entry from request queue. If this is a channel disable request, it immediately disables the channel and prepares to process the next entry.

If the current request is a transaction request and the USB bus time is enough for this transaction, USBFS will employ SIE to generate this transaction on USB bus.

When the required bus time for the current request is not enough in the current frame, and if this is a periodic request, USBFS stops processing the periodic queue and starts to process non-periodic request. If this is a non-periodic queue the USBFS will stop processing any queue and wait until the end of current frame.

28.5.3. USB device function

USB Device Connection

In device mode, USBFS stays at power-off state after initialization. After connecting to a USB host with 5V power supply through VBUS pin, USBFS enters into powered state. USBFS begins to switch on the pull-up resistor on DP line and thus, host side will detect a connection event.

Reset and Speed-Identification

The USB host always starts a USB reset when it detects a device connection, and USBFS in device mode will trigger a reset interrupt by hardware when it detects the reset event on USB bus.

After reset sequence, USBFS will trigger an ENUMF interrupt in USBFS_GINTF register and reports current enumerated device speed in ES bits in USBFS_DSTAT register, this bit field is always 0b'11'(full-speed).

As required by USB 2.0 protocol, USBFS doesn't support low-speed in device mode.

Suspend and Wake-up

A USB device will enter into suspend state when the USB bus stays at IDLE state and there is no change on data lines for 3ms. When USB device is in suspend state, most of its clock are closed to save power. The USB host is able to wake up the suspended device by generating a resume signal on USB bus. When USBFS detects the resume signal, the WKUPIF flag in USBFS_GINTF register will be set and the USBFS wake up interrupt will be triggered.

In suspend mode, USBFS is also able to remotely wake up the USB bus. Software may set RWKUP bit in USBFS_DCTL register to send a remote wake-up signal, and if remote wake-up is supported in USB host, the host will begin to send resume signal on USB bus.

Soft Disconnection

USBFS supports soft disconnection. After the device is powered on, USBFS will switch on the pull-up resistor on DP line so that the host can detect the connection. It is able to force a disconnection by setting the SD bit in USBFS_DCTL register. After the SD bit is set, USBFS will directly switch off the pull-up resistor, so that USB host will detect a disconnection on USB bus.

SOF tracking

When USBFS receives a SOF packet on USB bus, it will trigger a SOF interrupt and begin to count the bus time using local USB clock. The frame number of the current frame is reported in FNRSOF filed in USBFS_DSTAT register. When the USB bus time reaches EOF1 or EOF2 point (End of Frame, described in USB 2.0 protocol), USBFS will trigger an EOPFIF interrupt in USBFS_GINTF register. These flags and registers can be used to get current bus time and position information.

28.5.4. Data FIFO

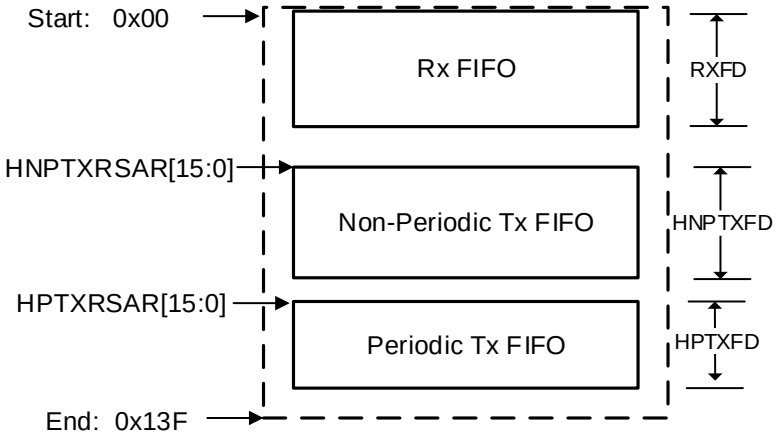
The USBFS contains a 1.25K bytes data FIFO for packet data storage. The data FIFO is

implemented by using an internal SRAM in USBFS.

Host Mode

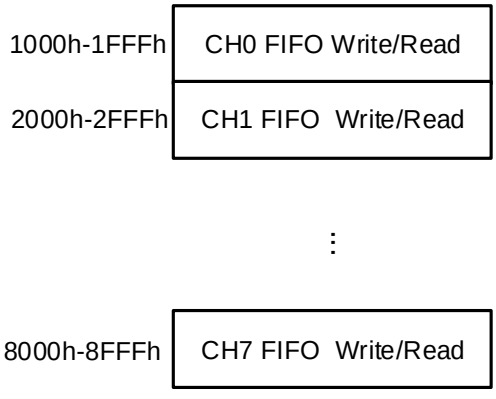
In host mode, the data FIFO space is divided into 3 parts: Rx FIFO for received packet, Non-Periodic Tx FIFO for non-period transmission packet and Periodic Tx FIFO for periodic transmission packet. All IN channels shares the Rx FIFO for packets reception. All the periodic OUT channels share the periodic Tx FIFO to packets transmission. All the non-periodic OUT channels share the non-Periodic Tx FIFO for transmit packets. The size and start offset of these data FIFOs should be configured using these registers: USBFS_GRFLEN, USBFS_HNPTFLEN and USBFS_HPTFLEN. ***Figure 28-4. HOST mode FIFO space in SRAM*** describes the structure of these FIFOs in SRAM. The values in the figure are in term of 32-bit words.

Figure 28-4. HOST mode FIFO space in SRAM



USBFS provides a special register area for the internal data FIFO reading and writing. ***Figure 28-5. Host mode FIFO access register map*** describes the register memory area that the data FIFO can write. This area can be read by any channel data FIFO. The addresses in the figure are addressed in bytes. Each channel has its own FIFO access register space, although all Non-periodic channels share the same FIFO and all the Periodic channels also share the same FIFO. It is important for USBFS to know which channel the current pushed packet belongs to. Rx FIFO is also able to be accessed using USBFS_GRSTATR / USBFS_GRSTATP register.

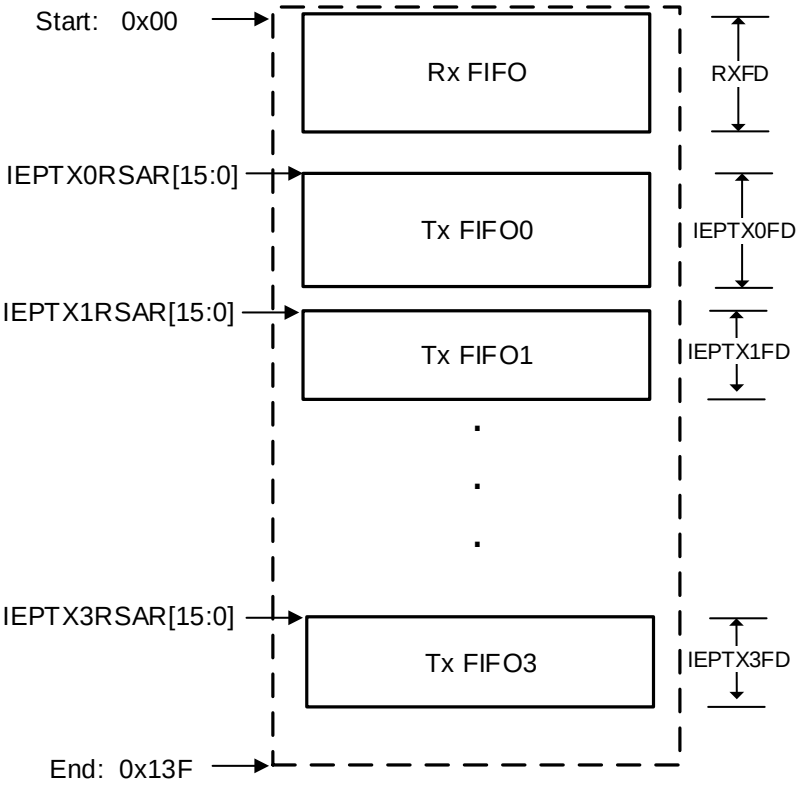
Figure 28-5. Host mode FIFO access register map



Device mode

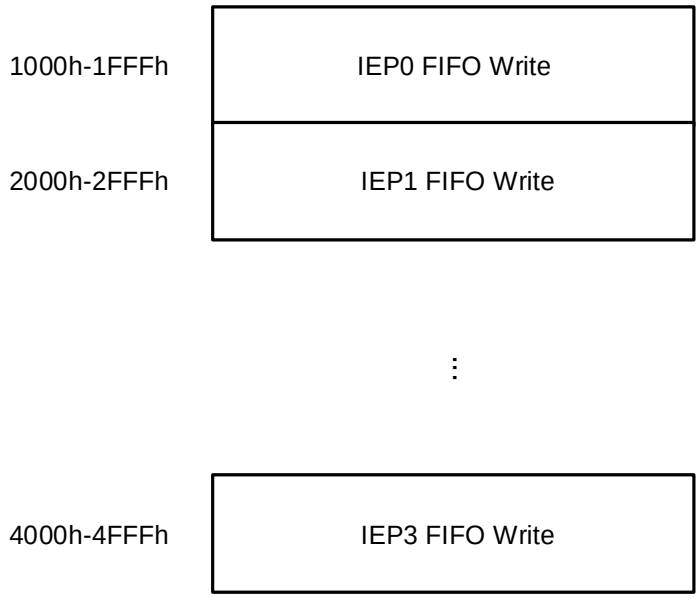
In device mode, the data FIFO is divided into several parts: one Rx FIFO, and 4 Tx FIFOs (one for each IN endpoint). All the OUT endpoints share the Rx FIFO for receiving packets. The size and start offset of these data FIFOs should be configured using USBFS_GRFLEN and USBFS_DIEPxTFLEN (x=0…3) registers. ***Figure 28-6. Device mode FIFO space in SRAM*** describes the structure of these FIFOs in SRAM. The values in the figure are in term of 32-bit words.

Figure 28-6. Device mode FIFO space in SRAM



USBFS provides a special register area for the internal data FIFO reading and writing. ***Figure 28-7. Device mode FIFO access register map*** describes the register memory area where the data FIFO can write. This area can be read by any endpoint FIFO. The addresses in the figure are addressed in bytes. Each endpoint has its own FIFO access register space. Rx FIFO is also able to be accessed using USBFS_GRSTATR/USBFS_GRSTATP register.

Figure 28-7. Device mode FIFO access register map



28.5.5. Operation guide

This section describes the advised operation guide for USBFS.

Host mode

Global register initialization sequence

1. Program USBFS_GAHBCS register according to application's demand, such as the TxFIFO's empty threshold, etc. GINTEN bit should be kept cleared at this time.
2. Program USBFS_GUSBCS register according to application's demand, such as the operation mode (host, device) and USB protocols.
3. Program USBFS_GCCFG register according to application's demand.
4. Program USBFS_GRFLEN, USBFS_HNPTFLEN_DIEP0TFLEN and USBFS_HPTFLEN register to configure the data FIFOs according to application's demand.
5. Program USBFS_GINTEN register to enable Mode Fault and Host Port interrupt and set GINTEN bit in USBFS_GAHBCS register to enable global interrupt.
6. Program USBFS_HPCS register and set PP bit.
7. Wait for a device's connection, and once a device is connected, the connection interrupt PCD in USBFS_HPCS register will be triggered. Then set PRST bit to perform a port reset. Wait for at least 10ms and then clear PRST bit.
8. Wait PEDC interrupt in USBFS_HPCS register and then read PE bit to ensure that the port is successfully enabled. Read PS [1:0] bits to get the connected device's speed and then program USBFS_HFT register to change the SOF interval if needed.

Channel initialization and enable sequence

1. Program USBFS_HCHxCTL registers with desired transfer type, direction, packet size,

etc. Ensure that CEN and CDIS bits keep cleared during configuration.

2. Program USBFS_HCHxINTEN register. Set the desired interrupt enable bits.
3. Program USBFS_HCHxLEN register. PCNT is the number of packets in a transfer and TLEN is the total bytes` number of all the transmitted or received packets in a transfer.

For OUT channel: If PCNT=1, the single packet's size is equal to TLEN. If PCNT>1, the former PCNT-1 packets are considered as max-packet-length packets whose size are defined by MPL field in USBFS_HCHxCTL register, and the last packet's size is calculated based on PCNT, TLEN and MPL. If software wants to send out a zero-length packet, it should program TLEN=0, PCNT=1.

For IN channel: Because the application doesn't know the actual received data size before the IN transaction finishes, TLEN can be set to a maximum possible value supported by Rx FIFO.

4. Set CEN bit in USBFS_HCHxCTL register to enable the channel.

Channel disable sequence

Software can disable the channel by setting both CEN and CDIS bits at the same time. USBFS will generate a channel disable request entry in request queue after the register setting operation. When the request entry reaches the top of request queue, it is processed by USBFS immediately:

For OUT channels, the specified channel will be disabled immediately. Then, a CH flag will be generated and the CEN and CDIS bits will be cleared by USBFS.

For IN channels, USBFS pushes a channel disable status entry into Rx FIFO. Software should then handle the Rx FIFO not empty event: read and pop this status entry, then, a CH flag will be generated and the CEN and CDIS bits will be cleared.

IN transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize the channel.
3. Enable the channel.
4. After the IN channel is enabled by software, USBFS generates an Rx request entry in the corresponding request queue.
5. When the Rx request entry reaches the top of the request queue, USBFS begins to process this request entry. If bus time for the IN transaction indicated by the request entry is enough, USBFS starts the IN transaction on USB bus.
6. If the IN transaction finishes successfully (ACK handshake received), USBFS pushes the received data packet into the Rx FIFO and triggers ACK flag. Otherwise, the status flag (NAK) reports the transaction result.
7. If the IN transaction described in step 5 is successful and PCNT is larger than 1 in step2, return to step 3 and continues to receive the remaining packets. If the IN transaction described in step 5 is not successful, return to step 3 to re-receive the packet again.
8. After all the transactions in a transfer are successfully received on USB bus, USBFS pushes a TF status entry into the Rx FIFO on top of the last packet data. Thus after

reading and popping all the received data packet, the TF status entry is need, USBFS generates TF flag to indicate that the transfer successfully finishes.

9. Disable the channel. Now the channel is in IDLE state and is ready for other transfers.

OUT transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize and enable the channel.
3. Write a packet into the channel's Tx FIFO (Periodic Tx FIFO or non-periodic Tx FIFO). After the whole packet data is written into the FIFO, USBFS generates a Tx request entry in the corresponding request queue and decreases the TLEN field in USBFS_HCHxLEN register by the written packet's size.
4. When the request entry reaches the top of the request queue, USBFS begins to process this request entry. If bus time for the transaction indicated by the request entry is enough, USBFS starts the OUT transaction on USB bus.
5. When the OUT transaction indicated by the request entry finishes on USB bus, PCNT in USBFS_HCHxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flag (NAK) reports the transaction result.
6. If the OUT transaction described in step 5 is successful and PCNT is larger than 1 in step2, return to step 3 and continues to send the remaining packets. If the OUT transaction described in step 5 is not successful, return to step 3 to resend the packet again.
7. After all the transactions in a transfer are successfully sent on USB bus, USBFS generates TF flag to indicate that the transfer successfully finishes.
8. Disable the channel. Now the channel is in IDLE state and is ready for other transfers.

Device mode

Global register initialization sequence

1. Program USBFS_GAHBCS register according to application's demand, such as the TxFIFO's empty threshold, etc. GINTEN bit should be kept cleared at this time.
2. Program USBFS_GUSBCS register according to application's demand, such as: the operation mode (host, device) and USB protocols.
3. Program USBFS_GCCFG register according to application's demand.
4. Program USBFS_GRFLEN, USBFS_HNPTFLEN_DIEP0TFLEN, USBFS_DIEPxTFLEN register to configure the data FIFOs according to application's demand.
5. Program USBFS_GINTEN register to enable Mode Fault, Suspend, SOF, Enumeration Done and USB Reset interrupt and then, set GINTEN bit in USBFS_GAHBCS register to enable global interrupt.
6. Program USBFS_DCFG register according to application's demand, such as the device address, etc.
7. After the device is connected to a host, the host will perform port reset on USB bus and this will trigger the RST interrupt in USBFS_GINTF register.
8. Wait for ENUMF interrupt in USBFS_GINTF register.

Endpoint initialization and enable sequence

1. Program USBFS_DIEPxCTL or USBFS_DOEPxCTL register with desired transfer type, packet size, etc.
2. Program USBFS_DIEPINTEN or USBFS_DOEPINTEN register. Set the desired interrupt enable bits.
3. Program USBFS_DIEPxLEN or USBFS_DOEPxLEN register. PCNT is the number of packets in a transfer and TLEN is the total bytes` number of all the transmitted or received packets in a transfer.

For IN endpoint：If PCNT=1, the single packet's size is equal to TLEN. If PCNT>1, the former PCNT-1 packets are considered as max-packet-length packets whose size are defined by MPL field in USBFS_DIEPxCTL register, and the last packet's size is calculated based on PCNT, TLEN and MPL. If a zero-length packet is required to be sent, it should program TLEN=0, PCNT=1.

For OUT endpoint：Because the application doesn't know the actual received data size before the OUT transaction finishes, TLEN can be set to a maximum possible value supported by Rx FIFO.

4. Set EPEN bit in USBFS_DIEPxCTL or USBFS_DOEPxCTL register to enable the endpoint.

Endpoint disable sequence

The endpoint can be disabled anytime when the EPEN bit in USBFS_DIEPxCTL or USBFS_DOEPxCTL registers is cleared.

IN transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize and enable the IN endpoint.
3. Write packets into the endpoint's Tx FIFO. Each time a data packet is written into the FIFO, USBFS decreases the TLEN field in USBFS_DIEPxLEN register by the written packet's size.
4. When an IN token received, USBFS transmits the data packet, and after the transaction finishes on USB bus, PCNT in USBFS_DIEPxLEN register is decreased by 1. If the transaction finishes successfully (ACK handshake received), the ACK flag is triggered. Otherwise, the status flags reports the transaction result.
5. After all the data packets in a transfer are successfully sent on USB bus, USBFS generates TF flag to indicate that the transfer successfully finishes and disables the IN endpoint.

OUT transfers operation sequence

1. Initialize USBFS global registers.
2. Initialize the endpoint and enable the endpoint.
3. When an OUT token received, USBFS receives the data packet or response with an NAK handshake based on the status of Rx FIFO and register configuration. If the

transaction finishes successfully (USBFS receives and saves the data packet into Rx FIFO successfully and sends ACK handshake on USB bus), PCNT in USBFS_DOEPxLEN register is decreased by 1 and the ACK flag is triggered, otherwise, the status flags report the transaction result.

4. After all the data packets in a transfer are successfully received on USB bus, USBFS pushes a TF status entry into the Rx FIFO on top of the last packet data. Thus after reading and popping all the received data packet, the TF status entry is read, USBFS generates TF flag to indicate that the transfer successfully finishes and disables the OUT endpoint.

28.6. Interrupts

USBFS has two interrupts: global interrupt and wake-up interrupt.

The source flags of the global interrupt are readable in USBFS_GINTF register and are listed in ***Table 28-2. USBFS global interrupt***.

Table 28-2. USBFS global interrupt

Interrupt Flag	Description	Operation Mode
SESIF	Session interrupt	Host or device mode
DISCIF	Disconnect interrupt flag	Host Mode
PTXFEIF	Periodic Tx FIFO empty interrupt flag	Host Mode
HCIF	Host channels interrupt flag	Host Mode
HPIF	Host port interrupt flag	Host Mode
PXNCIF / ISOONCIF	Periodic transfer Not Complete Interrupt flag /Isochronous OUT transfer Not Complete Interrupt Flag	Host or device mode
ISOINCIF	Isochronous IN transfer Not Complete Interrupt Flag	Device mode
OEPIF	OUT endpoint interrupt flag	Device mode
IEPIF	IN endpoint interrupt flag	Device mode
EOPFIF	End of periodic frame interrupt flag	Device mode
ISOOPDIF	Isochronous OUT packet dropped interrupt flag	Device mode
ENUMF	Enumeration finished	Device mode
RST	USB reset	Device mode
SP	USB suspend	Device mode
ESP	Early suspend	Device mode
GONAK	Global OUT NAK effective	Device mode
GNPINAK	Global IN Non-Periodic NAK effective	Device mode
NPTXFEIF	Non-Periodic Tx FIFO empty interrupt flag	Host Mode
RXFNEIF	Rx FIFO non-empty interrupt flag	Host or device mode
SOF	Start of frame	Host or device mode
MFIF	Mode fault interrupt flag	Host or device mode

Wake-up interrupt can be triggered when USBFS is in suspend state, even when the USBFS's clocks are stopped. The source of the wake-up interrupt is WKUPIF bit in USBFS_GINTF register.

28.7. Register definition

USBFS base address: 0x5000 0000

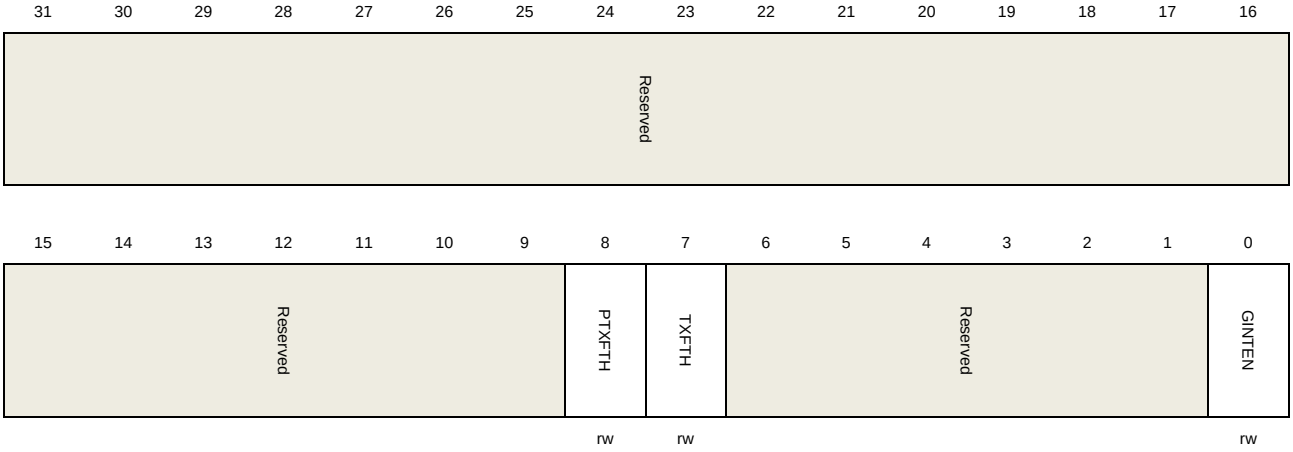
28.7.1. Global control and status registers

Global AHB control and status register (USBFS_GAHBCS)

Address offset: 0x0008

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	PTXFTH	Periodic Tx FIFO threshold 0: PTXFEIF will be triggered when the periodic transmit FIFO is half empty 1: PTXFEIF will be triggered when the periodic transmit FIFO is completely empty **Note:** Only accessible in host mode.
7	TXFTH	Tx FIFO threshold Device mode: 0: TXFEIF will be triggered when the IN endpoint transmit FIFO is half empty 1: TXFEIF will be triggered when the IN endpoint transmit FIFO is completely empty Host mode: 0: NPTXFEIF will be triggered when the non-periodic transmit FIFO is half empty 1: NPTXFEIF will be triggered when the non-periodic transmit FIFO is completely empty
6:1	Reserved	Must be kept at reset value.
0	GINTEN	Global interrupt enable 0: Global interrupt is not enabled. 1: Global interrupt is enabled.

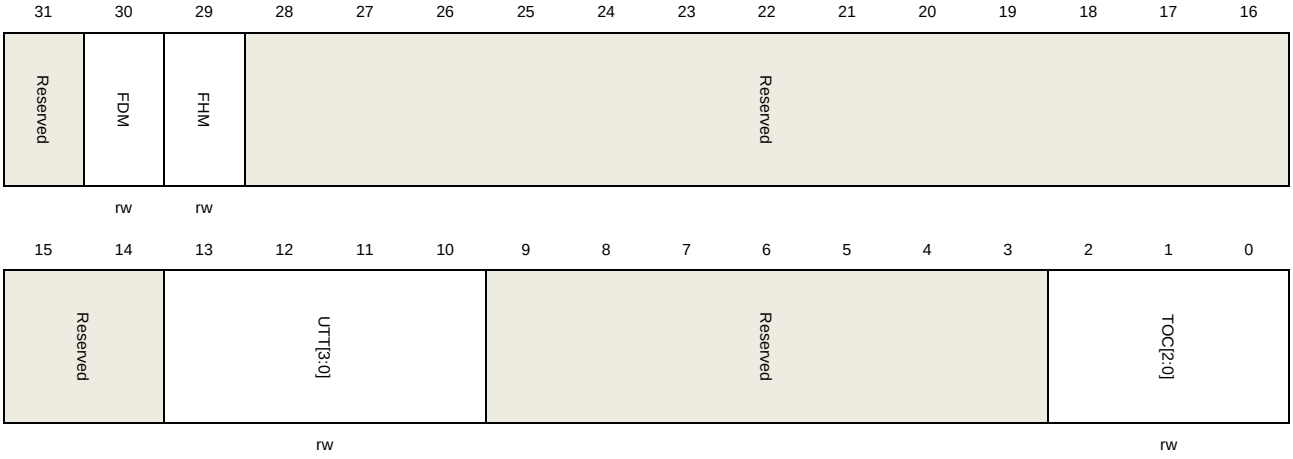
Note: Accessible in both device and host modes.

Global USB control and status register (USBFS_GUSBCS)

Address offset: 0x000C

Reset value: 0x0000 0880

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	FDM	Force device mode Setting this bit will force the core to device mode irrespective of the USBFS ID input pin. 0: Normal mode 1: Device mode The application must wait at least 25 ms for the change taking effect after setting the force bit. **Note:** Accessible in both device and host modes.
29	FHM	Force host mode Setting this bit will force the core to host mode irrespective of the USBFS ID input pin. 0: Normal mode 1: Host mode The application must wait at least 25 ms for the change taking effect after setting the force bit. **Note:** Accessible in both device and host modes.
28:14	Reserved	Must be kept at reset value.
13:10	UTT[3:0]	USB turnaround time Turnaround time in PHY clocks. **Note:** Only accessible in device mode.

9:3	Reserved	Must be kept at reset value.
2:0	TOC[2:0]	Timeout calibration USBFS always uses time-out value required in USB 2.0 when waiting for a packet. Application may use TOC [2:0] to add the value is in terms of PHY clock. (The frequency of PHY clock is 48MHZ.).

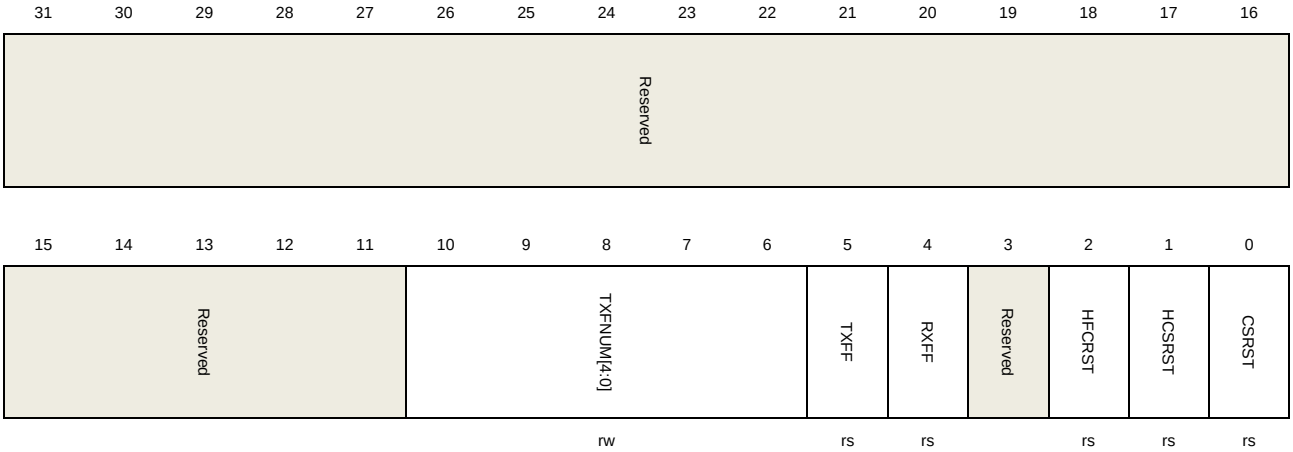
Global reset control register (USBFS_GRSTCTL)

Address offset: 0x0010

Reset value: 0x8000 0000

The application uses this register to reset various hardware features inside the core.

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



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10:6	TXFNUM[4:0]	Tx FIFO number Indicates which Tx FIFO will be flushed when TXFF bit in the same register is set. Host Mode: 00000: Only non-periodic Tx FIFO is flushed 00001: Only periodic Tx FIFO is flushed 1XXXX: Both periodic and non-periodic Tx FIFOs are flushed Other: Non data FIFO is flushed Device Mode: 00000: Only Tx FIFO0 is flushed 00001: Only Tx FIFO1 is flushed … 00011: Only Tx FIFO3 is flushed 1XXXX: All Tx FIFOs are flushed Other: Non data FIFO is flushed
5	TXFF	Tx FIFO flush Application set this bit to flush data Tx FIFOs and TXFNUM[4:0] bits decide the

FIFO number to be flushed. Hardware automatically clears this bit after the flush process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS.
Note: Accessible in both device and host modes.

4	RXFF	Rx FIFO flush Application set this bit to flush data Rx FIFO. Hardware automatically clears this bit after the flush process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS. **Note:** Accessible in both device and host modes.
3	Reserved	Must be kept at reset value.
2	HFCRST	Host frame counter reset Set by the application to reset the frame number counter in USBFS. After this bit is set, the frame number of the following SOF returns to 0. Hardware automatically clears this bit after the reset process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS. **Note:** Only accessible in host mode.
1	HCSRST	HCLK soft reset Set by the application to reset AHB clock domain circuit. Hardware automatically clears this bit after the reset process completes. After setting this bit, application should wait until this bit is cleared before any other operation on USBFS. **Note:** Accessible in both device and host modes.
0	CSRST	Core soft reset Resets the AHB and USB clock domains circuits, as well as most of the registers.

Global interrupt flag register (USBFS_GINTF)

Address offset: 0x0014
Reset value: 0x0400 0021

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIF	Reserved.	DISCIF	Reserved.		PTXFEIF	HCIF	HPIF	Reserved		PXNCIF/ ISOONCIF	ISOINCIF	OEPIF	IEPIF	Reserved	
rc_w1		rc_w1			r	r	r			rc_w1	rc_w1	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIF	ISOOPDIF	ENUMF	RST	SP	ESP	Reserved		GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	Reserved.	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1		rc_w1	r

Bits	Fields	Descriptions
31	WKUPIF	Wakeup interrupt flag This interrupt is triggered when a resume signal (in device mode) or a remote wakeup signal (in host mode) is detected on the USB. **Note:** Accessible in both device and host modes.
30	Reserved	Must be kept at reset value.
29	DISCIF	Disconnect interrupt flag This interrupt is triggered after a device disconnection. **Note:** Only accessible in host mode.
28:27	Reserved	Must be kept at reset value.
26	PTXFEIF	Periodic Tx FIFO empty interrupt flag This interrupt is triggered when the periodic transmit FIFO is either half or completely empty. The threshold is determined by the periodic Tx FIFO empty level bit (PTXFTH) in the USBFS_GAHBCS register. **Note:** Only accessible in host mode.
25	HCIF	Host channels interrupt flag Set by USBFS when one of the channels in host mode has raised an interrupt. First read USBFS_ HACHINT register to get the channel number, and then read the corresponding USBFS_HCHxINTF register to get the flags of the channel that cause the interrupt. This bit will be automatically cleared after the respective channel's flags which cause channel interrupt are cleared. **Note:** Only accessible in host mode.
24	HPIF	Host port interrupt flag Set by the core when USBFS detects that port status changes in host mode. Software should read USBFS_HPCS register to get the source of this interrupt. This bit will be automatically cleared after the flags that causing a port interrupt are cleared. **Note:** Only accessible in host mode.
23:22	Reserved	Must be kept at reset value.
21	PXNCIF	Periodic transfer Not Complete Interrupt flag USBFS sets this bit when there are periodic transactions for current frame not completed at the end of frame. (Host mode)
	ISOONCIF	Isochronous OUT transfer Not Complete Interrupt Flag At the end of a periodic frame (defined by EOPFT bit in USBFS_DCFG), USBFS will set this bit if there are still isochronous OUT endpoints for that not completed transactions. (Device Mode)
20	ISOINCIF	Isochronous IN transfer Not Complete Interrupt Flag At the end of a periodic frame (defined by EOPFT [1:0] bits in USBFS_DCFG),

		USBFS will set this bit if there are still isochronous IN endpoints for that not completed transactions. (Device Mode) **Note:** Only accessible in device mode.
19	OEPIF	OUT endpoint interrupt flag Set by USBFS when one of the OUT endpoints in device mode has raised an interrupt. Software should first read USBFS_DAEPINT register to get the device number, and then read the corresponding USBFS_DOEPxINTF register to get the flags of the endpoint that cause the interrupt. This bit will be automatically cleared after the respective endpoint's flags which cause this interrupt are cleared. **Note:** Only accessible in device mode.
18	IEPIF	IN endpoint interrupt flag Set by USBFS when one of the IN endpoints in device mode has raised an interrupt. Software should first read USBFS_DAEPINT register to get the device number, and then read the corresponding USBFS_DIEPxINTF register to get the flags of the endpoint that cause the interrupt. This bit will be automatically cleared after the respective endpoint's flags which cause this interrupt are cleared. **Note:** Only accessible in device mode.
17:16	Reserved	Must be kept at reset value.
15	EOPFIF	End of periodic frame interrupt flag When USB bus time in a frame reaches the value defined by EOPFT [1:0] bits in USBFS_DCFG register, USBFS sets this flag. **Note:** Only accessible in device mode.
14	ISOOPDIF	Isochronous OUT packet dropped interrupt flag USBFS set this bit if it receives an isochronous OUT packet but cannot save it into Rx FIFO because the FIFO doesn't have enough space. **Note:** Only accessible in device mode.
13	ENUMF	Enumeration finished USBFS sets this bit after the speed enumeration finishes. Read USBFS_DSTAT register to get the current device speed. **Note:** Only accessible in device mode.
12	RST	USB reset USBFS sets this bit when it detects a USB reset signal on bus. **Note:** Only accessible in device mode.
11	SP	USB suspend USBFS sets this bit when it detects that the USB bus is idle for 3 ms and enters suspend state. **Note:** Only accessible in device mode.
10	ESP	Early suspend USBFS sets this bit when it detects that the USB bus is idle for 3 ms.

		Note: Only accessible in device mode.
9:8	Reserved	Must be kept at reset value.
7	GONAK	Global OUT NAK effective Write 1 to SGONAK bit in the USBFS_DCTL register and USBFS will set GONAK flag after the writing to SGONAK takes effect. **Note:** Only accessible in device mode.
6	GNPINAK	Global Non-Periodic IN NAK effective Write 1 to SGINAK bit in the USBFS_DCTL register and USBFS will set GNPINAK flag after the writing to SGINAK takes effect. **Note:** Only accessible in device mode.
5	NPTXFEIF	Non-Periodic Tx FIFO empty interrupt flag This interrupt is triggered when the non-periodic transmit FIFO is either half or completely empty. The threshold is determined by the non-periodic Tx FIFO empty level bit (TXFTH) in the USBFS_GAHBCS register. **Note:** Only accessible in host mode.
4	RXFNEIF	Rx FIFO non-empty interrupt flag USBFS sets this bit when there is at least one packet or status entry in the Rx FIFO. **Note:** Accessible in both host and device modes.
3	SOF	Start of frame Host Mode: USBFS sets this bit when it prepares to transmit a SOF or Keep-Alive on USB bus. Software can clear this bit by writing 1. Device Mode: USBFS sets this bit after it receives a SOF token. The application can read the Device Status register to get the current frame number. Software can clear this bit by writing 1. **Note:** Accessible in both host and device modes.
2	Reserved	Must be kept at reset value.
1	MFIF	Mode fault interrupt flag USBFS sets this bit when software operates host-only register in device mode, or operates device-mode in host mode. These fault operations won't take effect. **Note:** Accessible in both host and device modes.
0	COPM	Current operation mode 0: Device mode 1: Host mode **Note:** Accessible in both host and device modes.

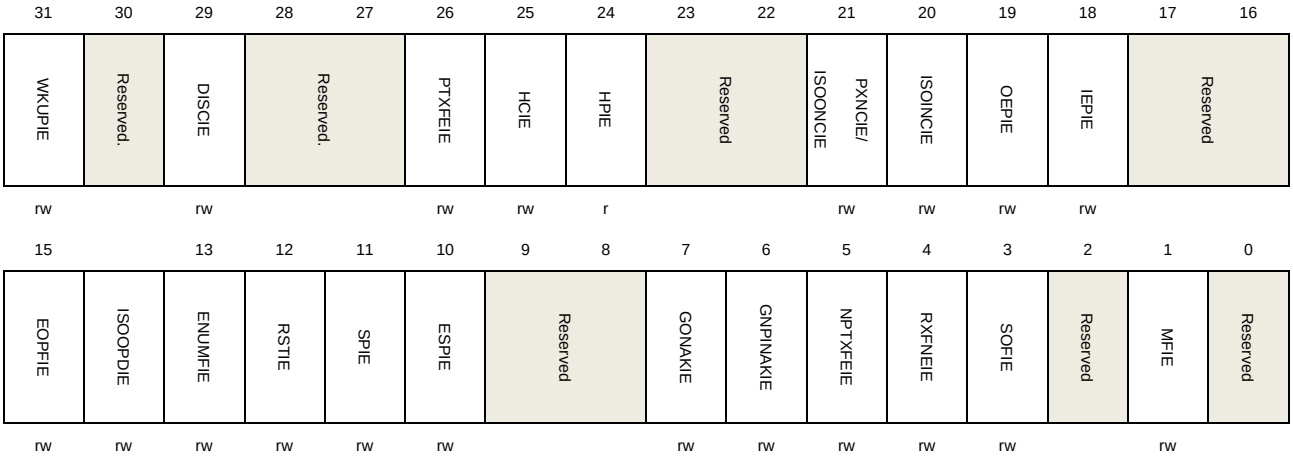
Global interrupt enable register (USBFS_GINTEN)

Address offset: 0x0018
Reset value: 0x0000 0000

This register works with the global interrupt flag register (USBFS_GINTF) to interrupt the

application. When an interrupt enable bit is disabled, the interrupt associated with that bit is not generated. However, the global Interrupt flag register bit corresponding to that interrupt is still set.

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



Bits	Fields	Descriptions
31	WKUPIE	Wakeup interrupt enable 0: Disable wakeup interrupt 1: Enable wakeup interrupt **Note:** Accessible in both host and device modes.
30	Reserved	Must be kept at reset value.
29	DISCIE	Disconnect interrupt enable 0: Disable disconnect interrupt 1: Enable disconnect interrupt **Note:** Only accessible in device mode.
28:27	Reserved	Must be kept at reset value.
26	PTXFEIE	Periodic Tx FIFO empty interrupt enable 0: Disable periodic Tx FIFO empty interrupt 1: Enable periodic Tx FIFO empty interrupt **Note:** Only accessible in host mode.
25	HCIE	Host channels interrupt enable 0: Disable host channels interrupt 1: Enable host channels interrupt **Note:** Only accessible in host mode.
24	HPIE	Host port interrupt enable 0: Disable host port interrupt 1: Enable host port interrupt **Note:** Only accessible in host mode.

23:22	Reserved	Must be kept at reset value.
21	PXNCIE	Periodic transfer not complete Interrupt enable 0: Disable periodic transfer not complete interrupt 1: Enable periodic transfer not complete interrupt **Note:** Only accessible in host mode.
	ISOONCIE	Isochronous OUT transfer not complete interrupt enable 0: Disable isochronous OUT transfer not complete interrupt 1: Enable isochronous OUT transfer not complete interrupt **Note:** Only accessible in device mode.
20	ISOINCIE	Isochronous IN transfer not complete interrupt enable 0: Disable isochronous IN transfer not complete interrupt 1: Enable isochronous IN transfer not complete interrupt **Note:** Only accessible in device mode.
19	OEPIE	OUT endpoints interrupt enable 0: Disable OUT endpoints interrupt 1: Enable OUT endpoints interrupt **Note:** Only accessible in device mode.
18	IEPIE	IN endpoints interrupt enable 0: Disable IN endpoints interrupt 1: Enable IN endpoints interrupt **Note:** Only accessible in device mode.
17:16	Reserved	Must be kept at reset value.
15	EOPFIE	End of periodic frame interrupt enable 0: Disable end of periodic frame interrupt 1: Enable end of periodic frame interrupt **Note:** Only accessible in device mode.
14	ISOOPDIE	Isochronous OUT packet dropped interrupt enable 0: Disable isochronous OUT packet dropped interrupt 1: Enable isochronous OUT packet dropped interrupt **Note:** Only accessible in device mode.
13	ENUMFIE	Enumeration finish enable 0: Disable enumeration finish interrupt 1: Enable enumeration finish interrupt **Note:** Only accessible in device mode.
12	RSTIE	USB reset interrupt enable 0: Disable USB reset interrupt 1: Enable USB reset interrupt **Note:** Only accessible in device mode.
11	SPIE	USB suspend interrupt enable

		0: Disable USB suspend interrupt 1: Enable USB suspend interrupt **Note:** Only accessible in device mode.
10	ESPIE	Early suspend interrupt enable 0: Disable early suspend interrupt 1: Enable early suspend interrupt **Note:** Only accessible in device mode.
9:8	Reserved	Must be kept at reset value.
7	GONAKIE	Global OUT NAK effective interrupt enable 0: Disable global OUT NAK interrupt 1: Enable global OUT NAK interrupt **Note:** Only accessible in device mode.
6	GNPINAKIE	Global non-periodic IN NAK effective interrupt enable 0: Disable global non-periodic IN NAK effective interrupt 1: Enable global non-periodic IN NAK effective interrupt **Note:** Only accessible in device mode.
5	NPTXFEIE	Non-periodic Tx FIFO empty interrupt enable 0: Disable non-periodic Tx FIFO empty interrupt 1: Enable non-periodic Tx FIFO empty interrupt **Note:** Only accessible in Host mode.
4	RXFNEIE	Receive FIFO non-empty interrupt enable 0: Disable receive FIFO non-empty interrupt 1: Enable receive FIFO non-empty interrupt **Note:** Accessible in both device and host modes.
3	SOFIE	Start of frame interrupt enable 0: Disable start of frame interrupt 1: Enable start of frame interrupt **Note:** Accessible in both device and host modes.
2	Reserved	Must be kept at reset value.
1	MFIE	Mode fault interrupt enable 0: Disable mode fault interrupt 1: Enable mode fault interrupt **Note:** Accessible in both device and host modes.
0	Reserved	Must be kept at reset value.

Global receive status read/receive status read and pop registers (USBFS_GRSTATR/USBFS_GRSTATP)

Address offset for Read: 0x001C

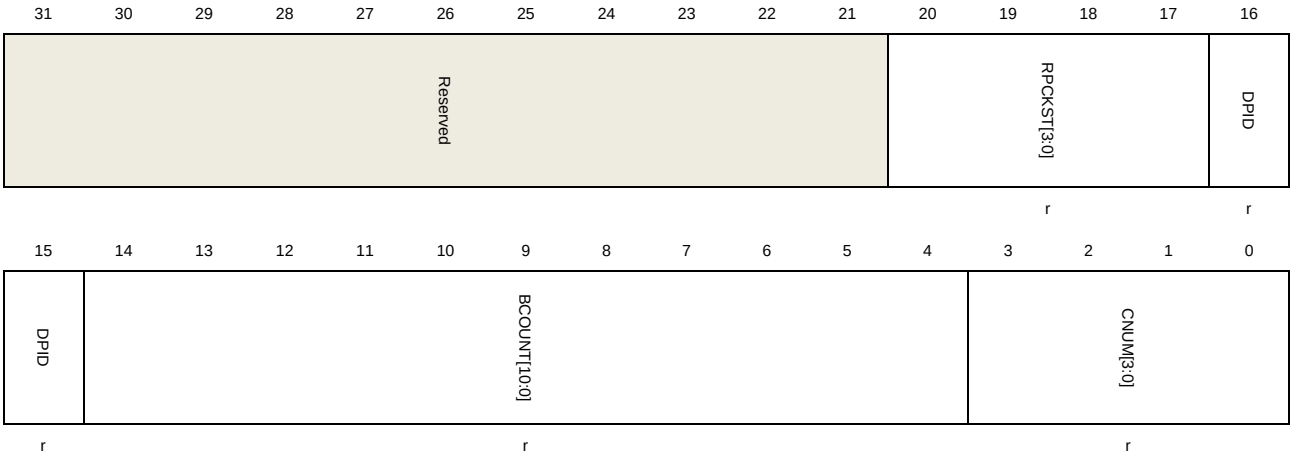
Address offset for Pop: 0x0020
Reset value: 0x0000 0000

A read to the receive status read register returns the entry of the top of the Rx FIFO. A read to the Receive status read and pop register additionally pops the top entry out of the Rx FIFO.

The entries in RxFIFO have different meanings in host and device modes. Software should only read this register after when Receive FIFO non-empty interrupt flag bit of the global interrupt flag register (RXFNEIF bit in USBFS_GINTF) is triggered.

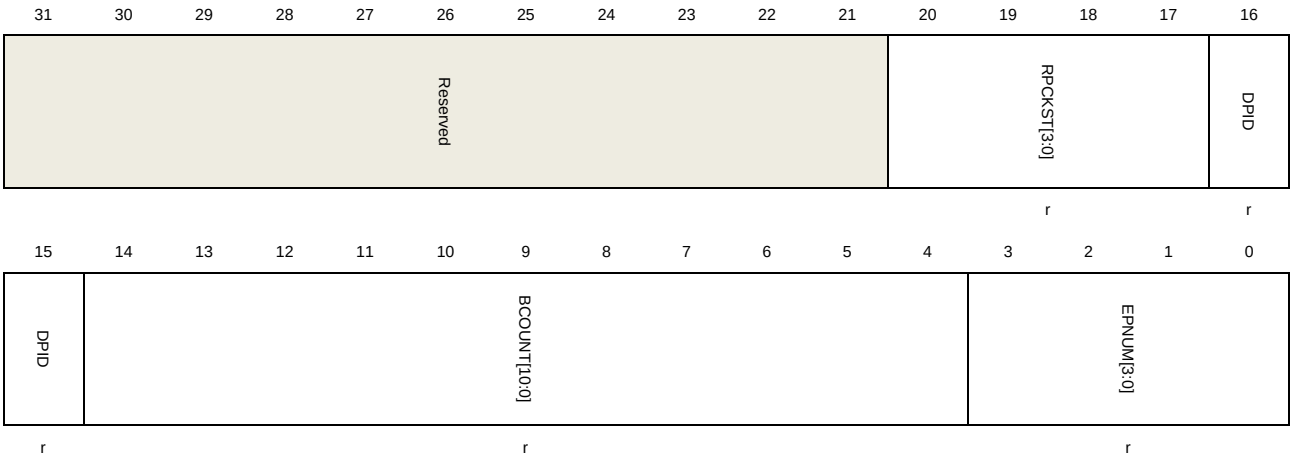
This register has to be accessed by word (32-bit)

Host mode:



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:17	RPCKST[3:0]	Received packet status 0010: IN data packet received 0011: IN transfer completed (generates an interrupt if popped) 0101: Data toggle error (generates an interrupt if popped) 0111: Channel halted (generates an interrupt if popped) Others: Reserved
16:15	DPID[1:0]	Data PID The Data PID of the received packet 00: DATA0 10: DATA1 Others: Reserved
14:4	BCOUNT[10:0]	Byte count The byte count of the received IN data packet.
3:0	CNUM[3:0]	Channel number The channel number to which the current received packet belongs.

Device mode:



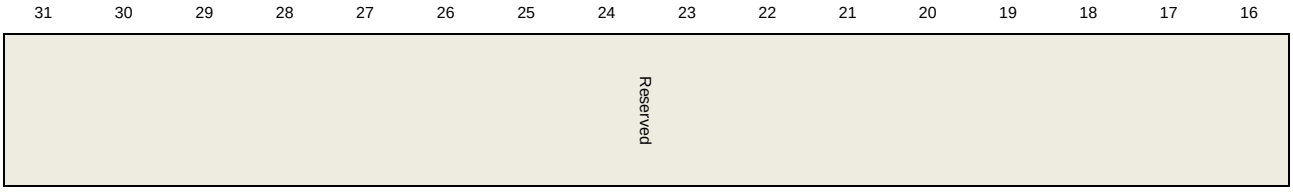
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:17	RPCKST[3:0]	Received packet status 0001: Global OUT NAK (generates an interrupt) 0010: OUT data packet received 0011: OUT transfer completed (generates an interrupt) 0100: SETUP transaction completed (generates an interrupt) 0110: SETUP data packet received Others: Reserved
16:15	DPID[1:0]	Data PID The Data PID of the received OUT data packet 00: DATA0 10: DATA1 Others: Reserved
14:4	BCOUNT[10:0]	Byte count The byte count of the received data packet.
3:0	EPNUM[3:0]	Endpoint number The endpoint number to which the current received packet belongs.

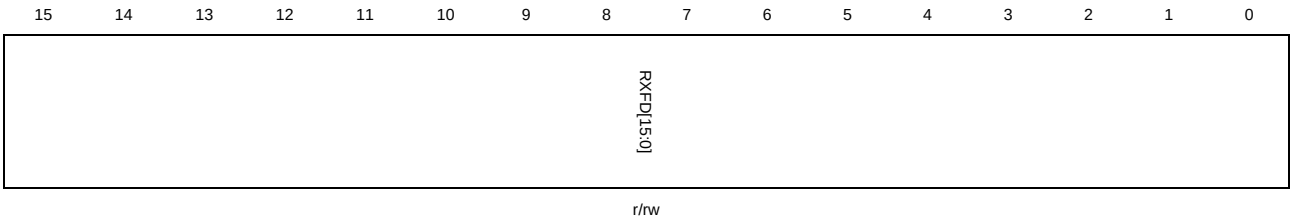
Global receive FIFO length register (USBFS_GRFLEN)

Address offset: 0x024

Reset value: 0x0000 0200

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





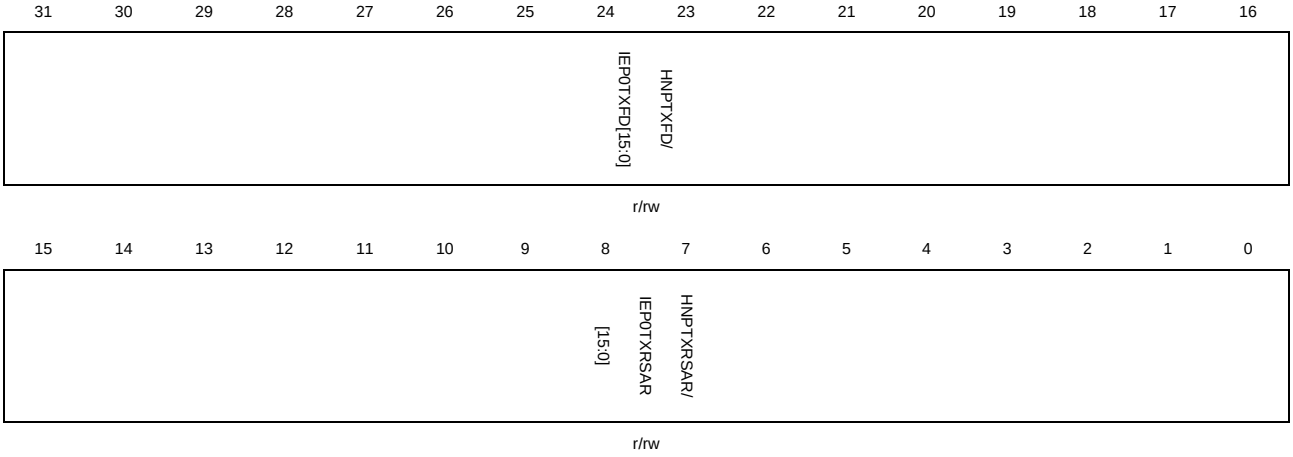
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RXFD[15:0]	Rx FIFO depth In terms of 32-bit words. 17≤RXFD≤256

Host non-periodic transmit FIFO length register /Device IN endpoint 0 transmit FIFO length (USBFS_HNPTFLEN _DIEP0TFLEN)

Address offset: 0x028

Reset value: 0x0200 0200

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



Host Mode:

Bits	Fields	Descriptions
31:16	HNPTXFD[15:0]	Host Non-periodic Tx FIFO depth In terms of 32-bit words. 17≤HNPTXFD≤256
15:0	HNPTXRSAR[15:0]	Host Non-periodic Tx RAM start address The start address for non-periodic transmit FIFO RAM is in term of 32-bit words.

Device Mode:

Bits	Fields	Descriptions
31:16	IEP0TXFD[15:0]	IN Endpoint 0 Tx FIFO depth In terms of 32-bit words.

		17≤IEP0TXFD≤140
15:0	IEP0TXRSAR[15:0]	IN Endpoint 0 TX RAM start address The start address for endpoint0 transmit FIFO RAM is in term of 32-bit words.

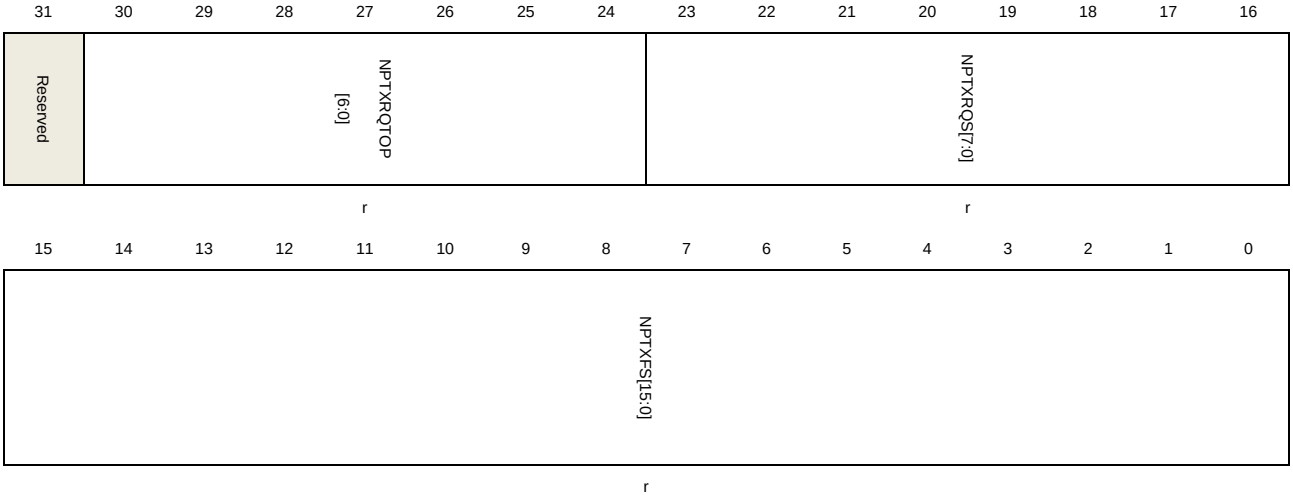
Host non-periodic transmit FIFO/queue status register (USBFS_HNPTFQSTAT)

Address offset: 0x002C
Reset value: 0x0008 0200

This register reports the current status of the non-periodic Tx FIFO and request queue. The request queue holds IN, OUT or other request entries in host mode.

Note: In Device mode, this register is not valid.

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:24	NPTXRQTOP[6:0]	Top entry of the non-periodic Tx request queue Entry in the non-periodic transmit request queue. Bits 30:27: Channel number Bits 26:25: – 00: IN/OUT token – 01: Zero-length OUT packet – 11: Channel halt request Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	NPTXRQS[7:0]	Non-periodic Tx request queue space The remaining space of the non-periodic transmit request queue. 0: Request queue is Full 1: 1 entry 2: 2 entries

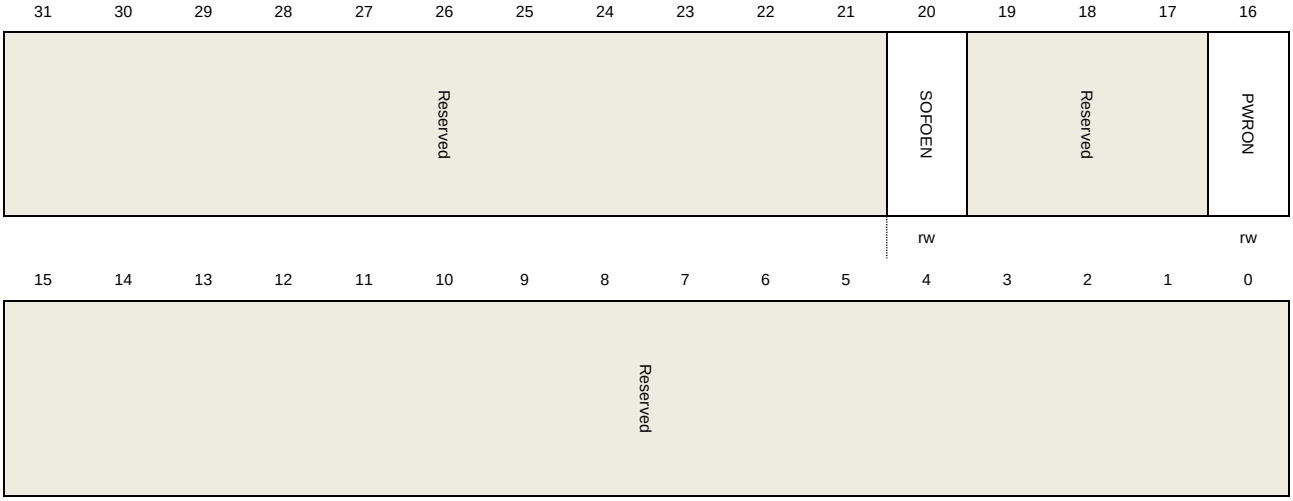
…
n: n entries (0≤n≤8)
Others: Reserved

15:0	NPTXFS[15:0]	Non-periodic Tx FIFO space The remaining space of the non-periodic transmit FIFO. In terms of 32-bit words. 0: Non-periodic Tx FIFO is full 1: 1 word 2: 2 words n: n words (0≤n≤NPTXFD) Others: Reserved

Global core configuration register (USBFS_GCCFG)

Address offset: 0x0038
Reset value: 0x000C 0000

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	SOFOEN	SOF output enable 0: SOF pulse output disabled. 1: SOF pulse output enabled.
19:17	Reserved	Must be kept at reset value.
16	PWRON	Power on This bit is the power switch for the internal embedded Full-Speed PHY. 0: Embedded Full-Speed PHY power off. 1: Embedded Full-Speed PHY power on.

15:0	Reserved	Must be kept at reset value.

Core ID register (USBFS_CID)

Address offset: 0x003C

Reset value: 0x0000 1000

This register contains the Product ID.

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

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

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

Bits	Fields	Descriptions
31:0	CID[31:0]	Core ID Software can write or read this field and uses this field as a unique ID for its application

Host periodic transmit FIFO length register (USBFS_HPTFLEN)

Address offset: 0x0100

Reset value: 0x0200 0600

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

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

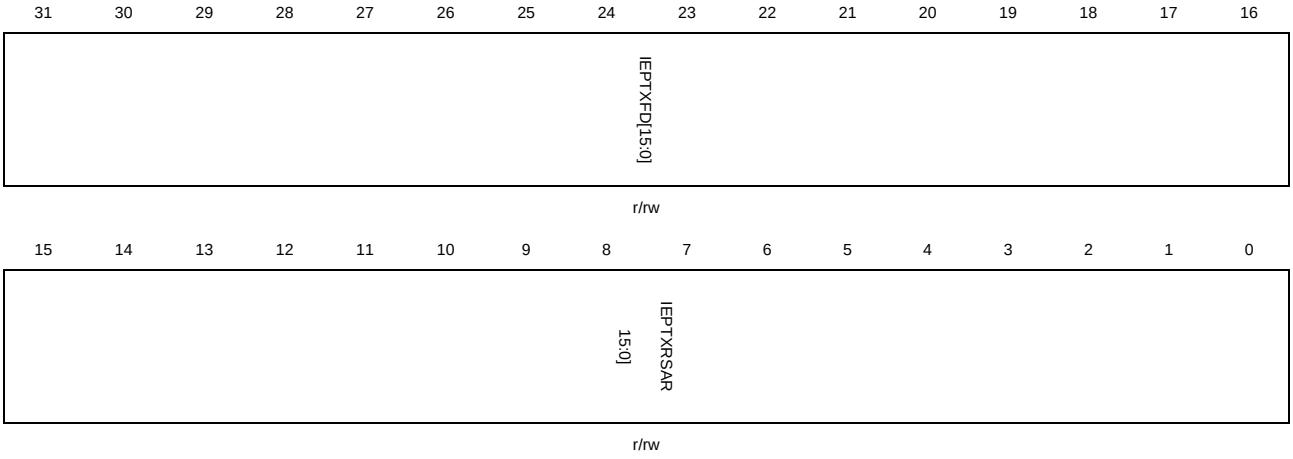
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HPTXFSAR [15:0]															
r/rw															

Bits	Fields	Descriptions
31:16	HPTXFD[15:0]	Host Periodic Tx FIFO depth In terms of 32-bit words. 1≤HPTXFD≤1024
15:0	HPTXFSAR[15:0]	Host periodic Tx FIFO RAM start address The start address for host periodic transmit FIFO RAM is in term of 32-bit words.

Device IN endpoint transmit FIFO length register (USBFS_DIEPxTFLEN) (x = 1 .. 3, where x = FIFO_number)

Address offset: 0x0104 + (FIFO_number – 1) × 0x04
Reset value: 0x0200 0400+ (FIFO_number – 1) × 0x200

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



Bits	Fields	Descriptions
31:16	IEPTXFD[15:0]	IN endpoint Tx FIFO depth In terms of 32-bit words. 1≤HPTXFD≤1024
15:0	IEPTXRSAR[15:0]	IN endpoint FIFO Tx RAM start address The start address for IN endpoint transmit FIFOx is in term of 32-bit words.

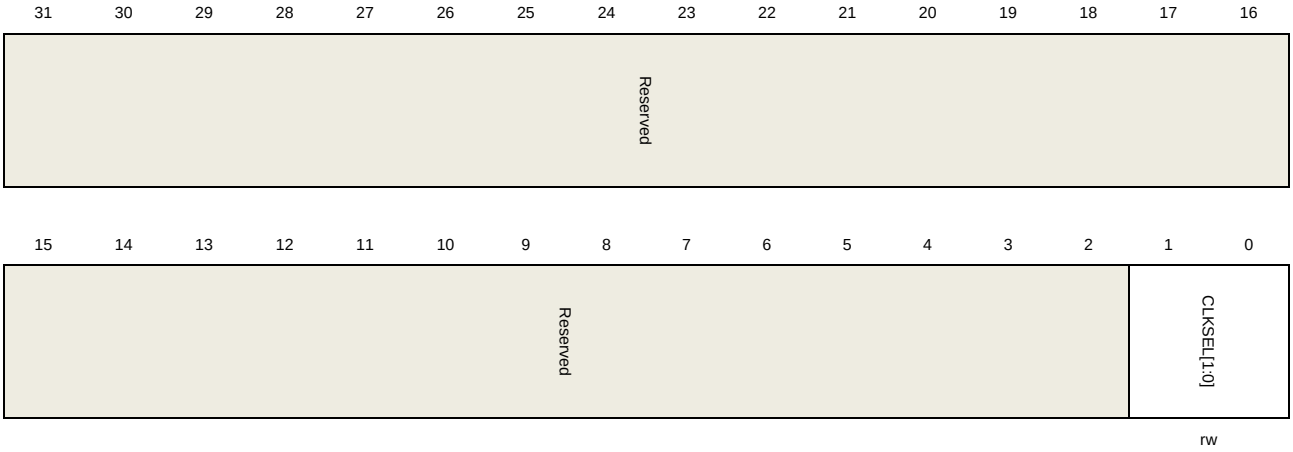
28.7.2. Host control and status registers

Host control register (USBFS_HCTL)

Address offset: 0x0400
Reset value: 0x0000 0000

This register configures the core after power on in host mode. Do not modify it after host initialization.

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



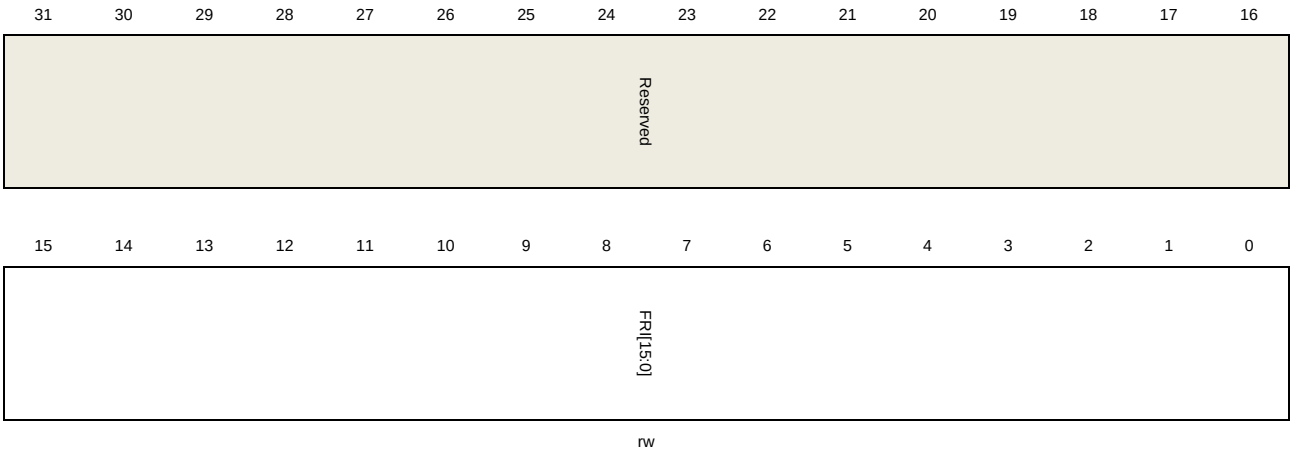
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	CLKSEL[1:0]	Clock select for usb clock. 01: 48MHz clock others: reserved

Host frame interval register (USBFS_HFT)

Address offset: 0x0404
Reset value: 0x0000 BB80

This register sets the frame interval for the current enumerating speed when USBFS controller is enumerating. The modification of the frame interval will take effect in the next frame.

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	FRI[15:0]	Frame interval

This value describes the frame time in terms of PHY clocks. Each time when port is enabled after a port reset operation, USBFS use a proper value according to the current speed, and software can write to this field to change the value. This value should be calculated using the frequency described below:

Full-Speed: 48MHz

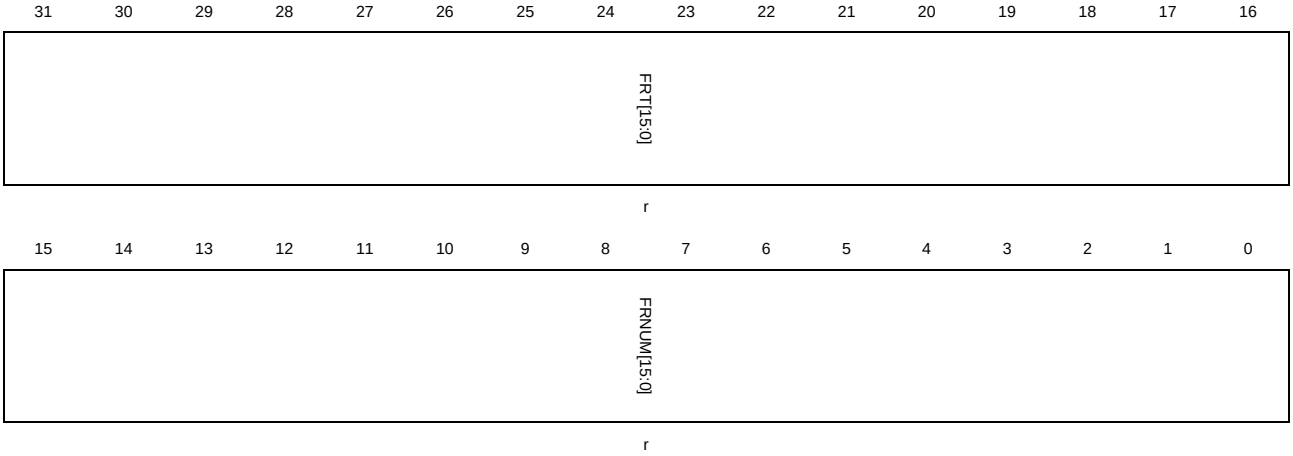
Low-Speed: 6MHz

Host frame information remaining register (USBFS_HFINFR)

Address offset: 0x408

Reset value: 0xBB80 0000

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



Bits	Fields	Descriptions
31:16	FRT[15:0]	Frame remaining time This field reports the remaining time of current frame in terms of PHY clocks.
15:0	FRNUM[15:0]	Frame number This field reports the frame number of current frame and returns to 0 after it reaches 0x3FFF.

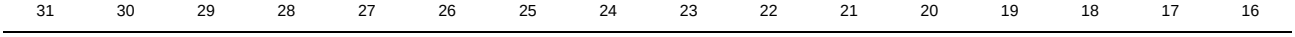
Host periodic transmit FIFO/queue status register (USBFS_HPTFQSTAT)

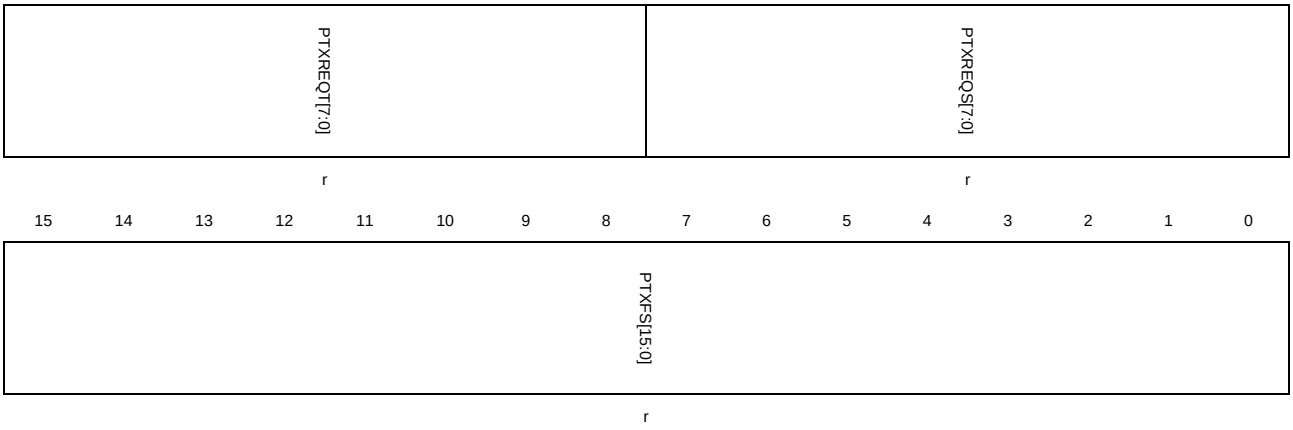
Address offset: 0x0410

Reset value: 0x0008 0200

This register reports the current status of the host periodic Tx FIFO and request queue. The request queue holds IN, OUT or other request entries in host mode.

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





Bits	Fields	Descriptions
31:24	PTXREQT[7:0]	Top entry of the periodic Tx request queue Entry in the periodic transmit request queue. Bits 30:27: Channel Number Bits 26:25: 00: IN/OUT token 01: Zero-length OUT packet 11: Channel halt request Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	PTXREQS[7:0]	Periodic Tx request queue space The remaining space of the periodic transmit request queue. 0: Request queue is Full 1: 1 entry 2: 2 entries … n: n entries (0≤n≤8) Others: Reserved
15:0	PTXFS[15:0]	Periodic Tx FIFO space The remaining space of the periodic transmit FIFO. In terms of 32-bit words. 0: periodic Tx FIFO is full 1: 1 word 2: 2 words n: n words (0≤n≤PTXFD) Others: Reserved

Host all channels interrupt register (USBFS_HACHINT)

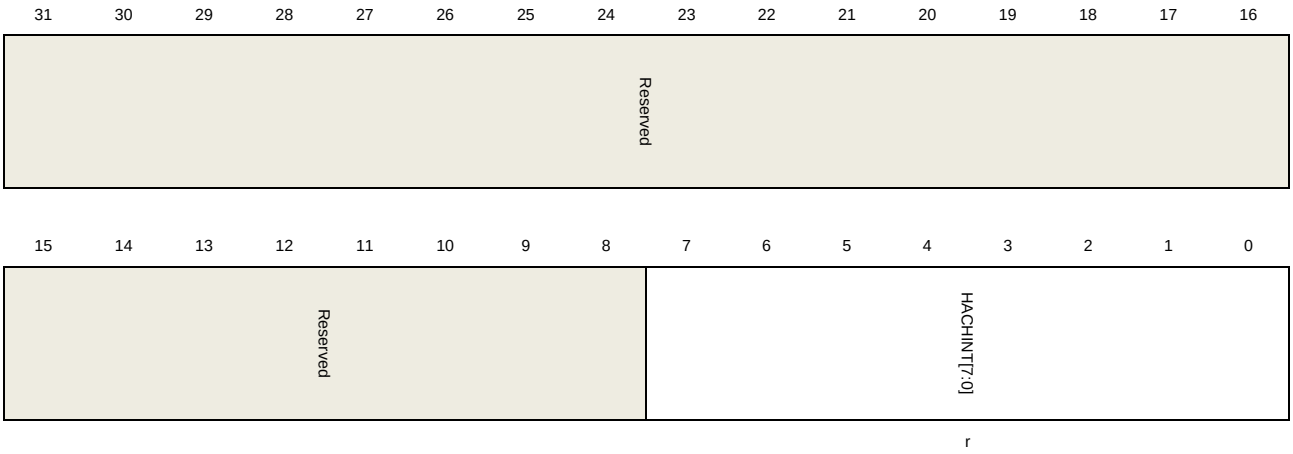
Address offset: 0x0414

Reset value: 0x0000 0000

When a channel interrupt is triggered, USBFS set corresponding bit in this register and

software should read this register to know which channel is asserting interrupts.

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	HACHINT[7:0]	Host all channel interrupts Each bit represents a channel: Bit 0 for channel 0, bit 7 for channel 7.

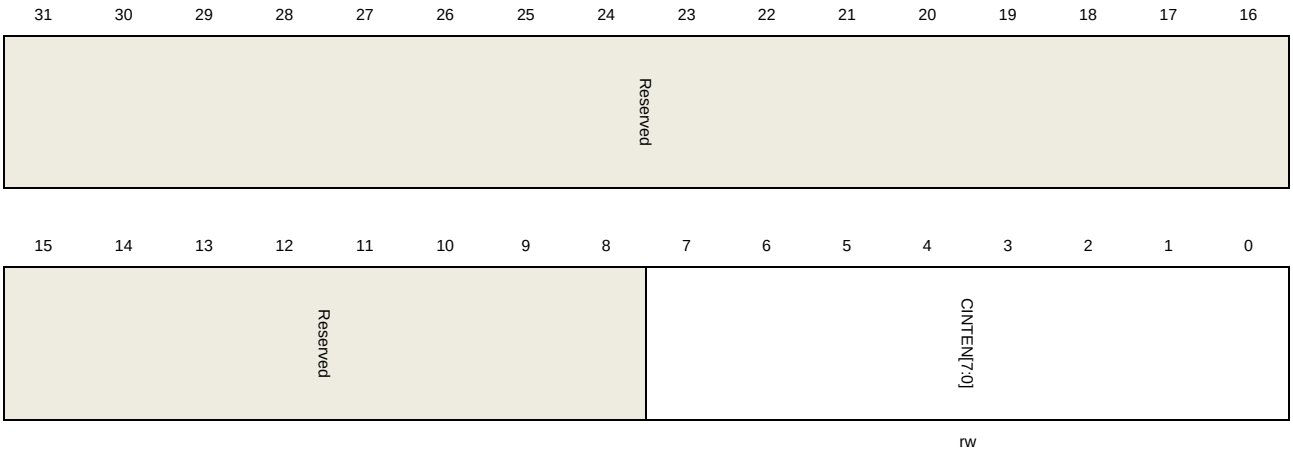
Host all channels interrupt enable register (USBFS_HACHINTEN)

Address offset: 0x0418

Reset value: 0x0000 0000

This register can be used by software to enable or disable a channel's interrupt. Only the channel whose corresponding bit in this register is set is able to cause the channel interrupt flag HCIF in USBFS_GINTF register.

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



Bits	Fields	Descriptions

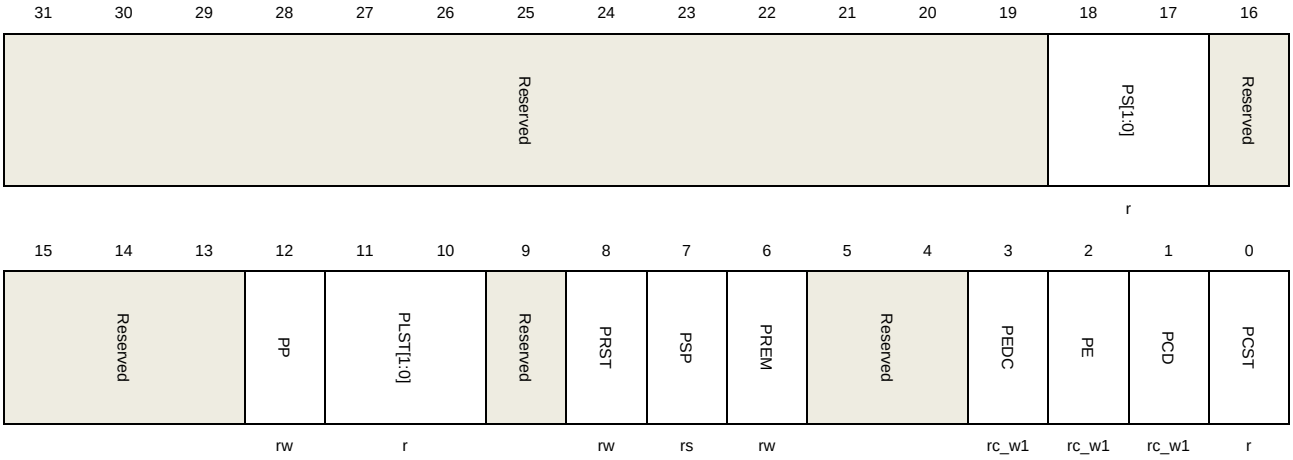
31:8	Reserved	Must be kept at reset value.
7:0	CINTEN[7:0]	Channel interrupt enable 0: Disable channel-n interrupt 1: Enable channel-n interrupt Each bit represents a channel: Bit 0 for channel 0, bit 7 for channel 7.

Host port control and status register (USBFS_HPCS)

Address offset: 0x0440
Reset value: 0x0000 0000

This register controls the port's behavior and also has some flags which report the status of the port. The HPIF flag in USBFS_GINTF register will be triggered if one of these flags in this register is set by USBFS: PRST, PEDC and PCD.

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



Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18:17	PS[1:0]	Port speed Report the enumerated speed of the device attached to this port. 01: Full speed 10: Low speed Others: Reserved
16:13	Reserved	Must be kept at reset value.
12	PP	Port power This bit should be set before a port is used. Because USBFS doesn't have power supply ability, it only uses this bit to know whether the port is in powered state. Software should ensure the true power supply on VBUS before setting this bit. 0: Port is powered off 1: Port is powered on

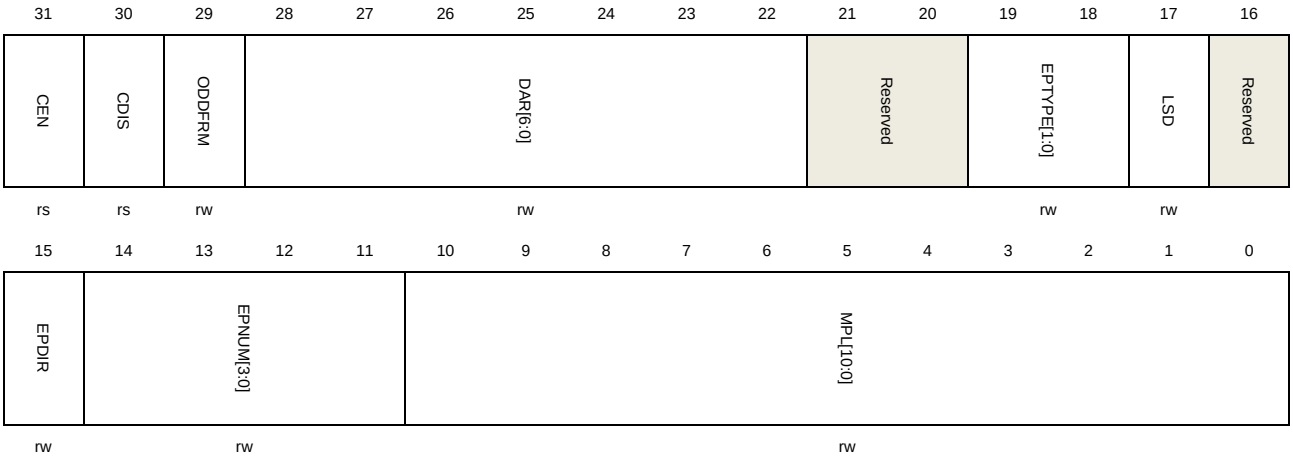
11:10	PLST[1:0]	Port line status Report the current state of USB data lines Bit 10: State of DP line Bit 11: State of DM line
9	Reserved	Must be kept at reset value.
8	PRST	Port reset Application sets this bit to start a reset signal on USB port. Application should clear this bit when it wants to stop the reset signal. 0: Port is not in reset state 1: Port is in reset state
7	PSP	Port suspend Application sets this bit to put port into suspend state. When this bit is set the port stops sending SOF tokens. This bit can only be cleared by the following operations: PRST bit in this register is set by application PREM bit in this register is set A remote wakeup signal is detected A device disconnect is detected 0: Port is not in suspend state 1: Port is in suspend state
6	PREM	Port resume Application sets this bit to start a resume signal on USB port. Application should clear this bit when it wants to stop the resume signal. 0: No resume driven 1: Resume driven
5:4	Reserved	Must be kept at reset value.
3	PEDC	Port enable/disable change Set by the core when the status of the Port enable bit 2 in this register changes.
2	PE	Port Enable This bit is automatically set by USBFS after a USB reset signal finishes and cannot be set by software. This bit is cleared by the following events: A disconnect condition Software clearing this bit 0: Port disabled 1: Port enabled
1	PCD	Port connect detected Set by USBFS when a device connection is detected. This bit can be cleared by writing 1 to this bit.
0	PCST	Port connect status 0: Device is not connected to the port

1: Device is connected to the port

Host channel-x control register (USBFS_HCHxCTL) (x = 0 .. 7, where x = channel_number)

Address offset: 0x0500 + (channel_number × 0x20)
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	CEN	Channel enable Set by the application and cleared by USBFS. 0: Channel disabled 1: Channel enabled Software should following the operation guide to disable or enable a channel.
30	CDIS	Channel disable Software can set this bit to disable the channel from processing transactions. Software should follow the operation guide to disable or enable a channel.
29	ODDFRM	Odd frame For periodic transfers (interrupt or isochronous transfer), this bit controls that whether in an odd frame or even frame this channel's transaction is desired to be processed. 0: Even frame 1: Odd frame
28:22	DAR[6:0]	Device address The address of the USB device that this channel wants to communicate with.
21:20	Reserved	Must be kept at reset value.
19:18	EPTYPE[1:0]	Endpoint type The transfer type of the endpoint that this channel wants to communicate with.

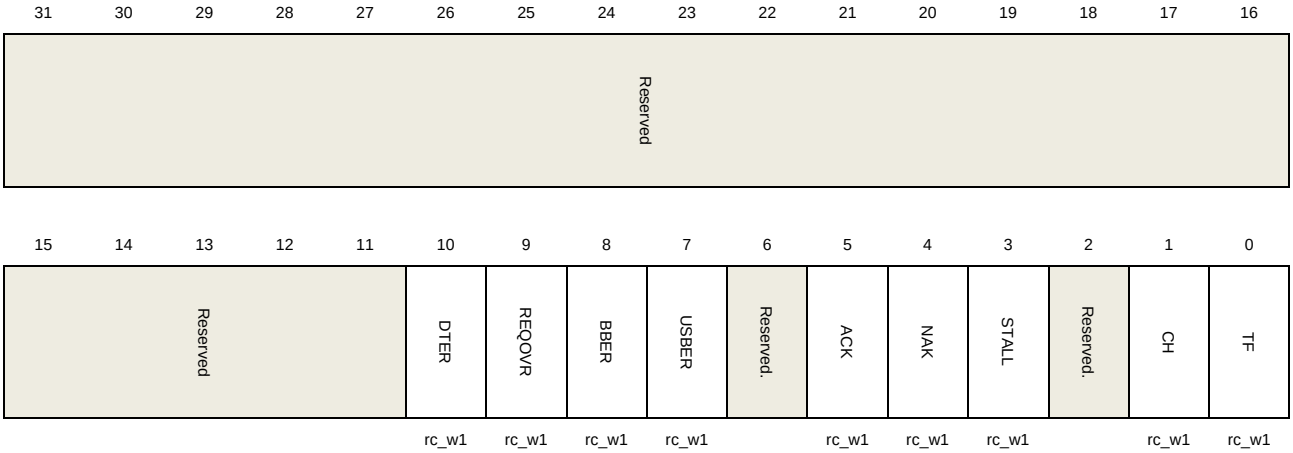
		00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	LSD	Low-Speed device The device that this channel wants to communicate with is a Low-Speed Device.
16	Reserved	Must be kept at reset value.
15	EPDIR	Endpoint direction The transfer direction of the endpoint that this channel wants to communicate with. 0: OUT 1: IN
14:11	EPNUM[3:0]	Endpoint number The number of the endpoint that this channel wants to communicate with.
10:0	MPL[10:0]	Maximum packet length The target endpoint's maximum packet length.

Host channel-x interrupt flag register (USBFS_HCHxINTF) (x = 0..7, where x = channel number)

Address offset: 0x0508 + (channel_number × 0x20)
Reset value: 0x0000 0000

This register contains the status and events of a channel, when software get a channel interrupt, it should read this register for the respective channel to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1.

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



Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.

10	DTER	Data toggle error The IN transaction gets a data packet but the PID of this packet doesn't match DPID [1:0] bits in USBFS_HCHxLEN register.
9	REQOVR	Request queue overrun The periodic request queue is full when software starts new transfers.
8	BBER	Babble error A babble condition occurs on USB bus. A typical reason for babble condition is that a device sends a data packet and the packet length exceeds the endpoint's maximum packet length.
7	USBER	USB Bus Error The USB error flag is set when the following conditions occurs during receiving a packet: A received packet has a wrong CRC field A stuff error detected on USB bus Timeout when waiting for a response packet
6	Reserved	Must be kept at reset value.
5	ACK	ACK An ACK response is received or transmitted
4	NAK	NAK A NAK response is received.
3	STALL	STALL A STALL response is received.
2	Reserved	Must be kept at reset value.
1	CH	Channel halted This channel is disabled by a request, and it will not response to other requests during the request processing.
0	TF	Transfer finished All the transactions of this channel finish successfully, and no error occurs. For IN channel, this flag will be triggered after PCNT bits in USBFS_HCHxLEN register reach zero. For OUT channel, this flag will be triggered when software reads and pops a TF status entry from the RxFIFO.

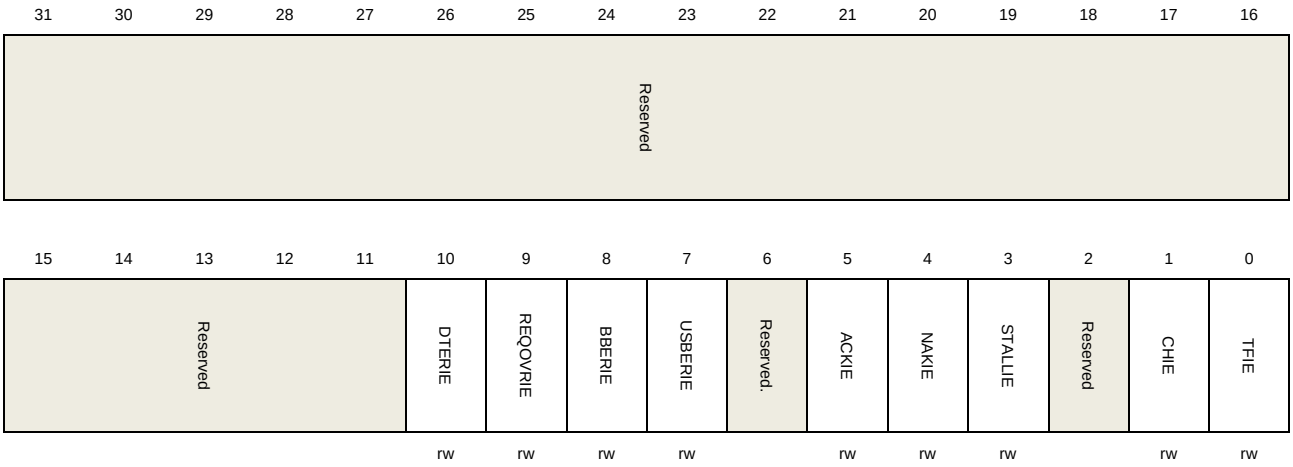
Host channel-x interrupt enable register (USBFS_HCHxINTEN) (x = 0 .. 7, where x = channel number)

Address offset: 0x050C + (channel_number × 0x20)
Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBFS_HCHxINTF register. If a bit in this register is set by software, the corresponding bit in USBFS_HCHxINTF register is

able to trigger a channel interrupt. The bits in this register are set and cleared by software.

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



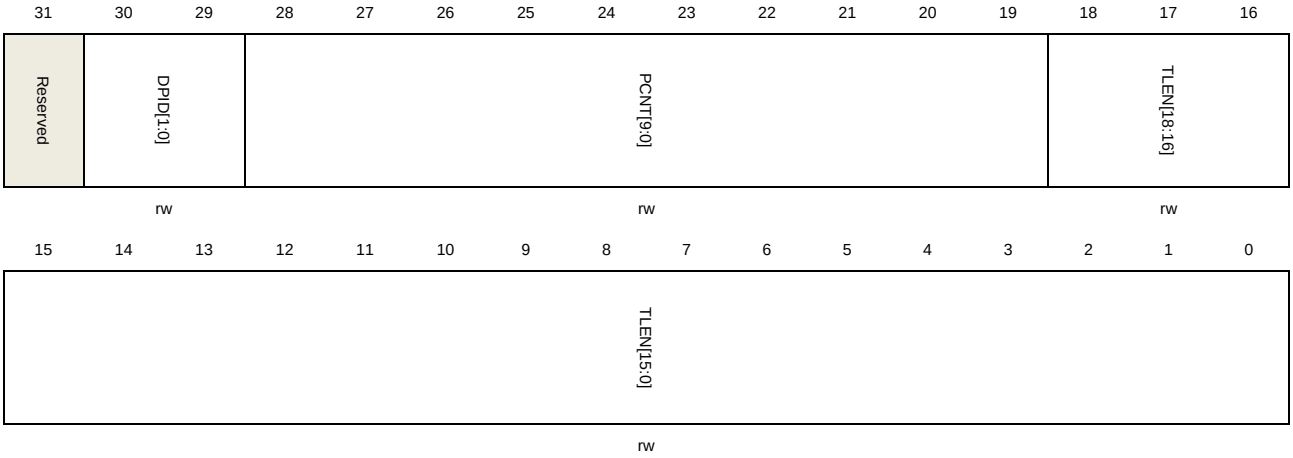
Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10	DTERIE	Data toggle error interrupt enable 0: Disable data toggle error interrupt 1: Enable data toggle error interrupt
9	REQOVRIE	Request queue overrun interrupt enable 0: Disable request queue overrun interrupt 1: Enable request queue overrun interrupt
8	BBERIE	Babble error interrupt enable 0: Disable babble error interrupt 1: Enable babble error interrupt
7	USBERIE	USB bus error interrupt enable 0: Disable USB bus error interrupt 1: Enable USB bus error interrupt
6	Reserved	Must be kept at reset value.
5	ACKIE	ACK interrupt enable 0: Disable ACK interrupt 1: Enable ACK interrupt
4	NAKIE	NAK interrupt enable 0: Disable NAK interrupt 1: Enable NAK interrupt
3	STALLIE	STALL interrupt enable 0: Disable STALL interrupt 1: Enable STALL interrupt

2	Reserved	Must be kept at reset value.
1	CHIE	Channel halted interrupt enable 0: Disable channel halted interrupt 1: Enable channel halted interrupt
0	TFIE	Transfer finished interrupt enable 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

Host channel-x transfer length register (USBFS_HCHxLEN) (x = 0..7, where x = channel number)

Address offset: 0x0510 + (channel_number × 0x20)
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	DPID[1:0]	Data PID Software should write this field before the transfer starts. For OUT transfers, this field controls the Data PID of the first transmitted packet. For IN transfers, this field controls the expected Data PID of the first received packet, and DTERR will be triggered if the Data PID doesn't match. After the transfer starts, USBFS changes and toggles this field automatically following the USB protocol. 00: DATA0 10: DATA1 11: SETUP (For control transfer only) 01: Reserved
28:19	PCNT[9:0]	Packet count The number of data packets desired to be transmitted (OUT) or received (IN) in a transfer.

Software should program this field before the channel is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet transmission.

18:0	TLEN[18:0]	Transfer length The total data bytes` number of a transfer. For OUT transfers, this field is the total data bytes of all the data packets desired to be transmitted in an OUT transfer. Software should program this field before the channel is enabled. When software successfully writes a packet into the channel's data TxFIFO, this field is decreased by the byte size of the packet. For IN transfer each time software or DMA reads out a packet from the RxFIFO, this field is decreased by the byte size of the packet.

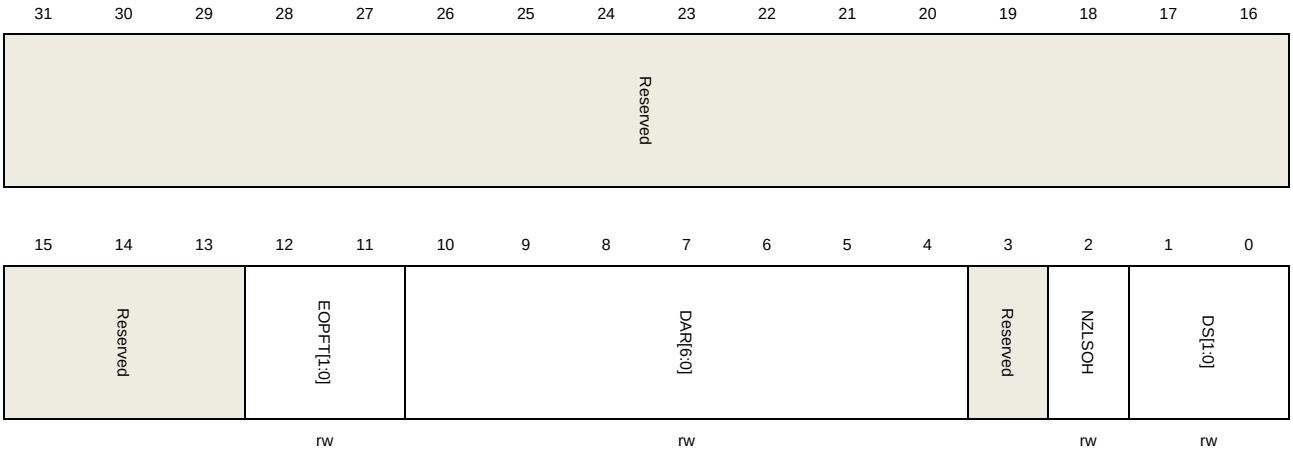
28.7.3. Device control and status registers

Device configuration register (USBFS_DCFG)

Address offset: 0x0800
Reset value: 0x0000 0003

This register configures the core in device mode after power on or after certain control commands or enumeration. Do not change this register after device initialization.

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



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:11	EOPFT[1:0]	End of periodic frame time This field defines the percentage time point in a frame that the end of periodic frame (EOPF) flag should be triggered. 00: 80% of the frame time 01: 85% of the frame time 10: 90% of the frame time

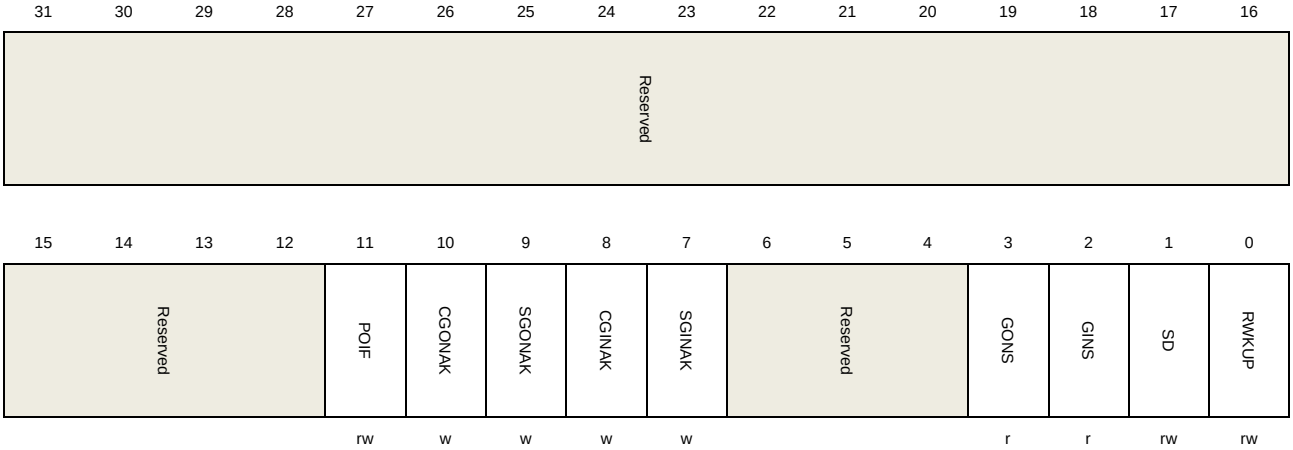
		11: 95% of the frame time
10:4	DAR[6:0]	Device address This field defines the USB device's address. USBFS uses this field to match with the incoming token's device address field. Software should program this field after receiving a Set Address command from USB host.
3	Reserved	Must be kept at reset value.
2	NZLSOH	Non-zero-length status OUT handshake When a USB device receives a non-zero-length data packet during status OUT stage, this field controls that either USBFS should receive this packet or reject this packet with a STALL handshake. 0: Treat this packet as a normal packet and response according to the status of NAKS and STALL bits in USBFS_DOEPxCTL register. 1: Send a STALL handshake and don't save the received OUT packet.
1:0	DS[1:0]	Device speed This field controls the device speed when the device connected to a host. 11: Full speed Others: Reserved

Device control register (USBFS_DCTL)

Address offset: 0x0804

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	POIF	Power-on initialization finished Software should set this bit to notify USBFS that the registers are initialized after waking up from power down state.

10	CGONAK	Clear global OUT NAK Software sets this bit to clear GONS bit in this register.
9	SGONAK	Set global OUT NAK Software sets this bit to set GONS bit in this register. When GONS bit is zero, setting this bit will also cause GONAK flag in USBFS_GINTF register triggered after a while. Software should clear the GONAK flag before writing this bit again.
8	CGINAK	Clear global IN NAK Software sets this bit to clear GINS bit in this register.
7	SGINAK	Set global IN NAK Software sets this bit to set GINS bit in this register. When GINS bit is zero, setting this bit will also cause GINAK flag in USBFS_GINTF register triggered after a while. Software should clear the GINAK flag before writing this bit again.
6:4	Reserved	Must be kept at reset value.
3	GONS	Global OUT NAK status 0: The handshake that USBFS response to OUT transaction packet and whether to save the OUT data packet are decided by Rx FIFO status, endpoint's NAK and STALL bits. 1: USHBS always responses to OUT transaction with NAK handshake and doesn't save the incoming OUT data packet.
2	GINS	Global IN NAK status 0: The response to IN transaction is decided by Tx FIFO status, endpoint's NAK and STALL bits. 1: USBFS always responses to IN transaction with a NAK handshake.
1	SD	Soft disconnect Software can use this bit to generate a soft disconnect condition on USB bus. After this bit is set, USBFS switches off the pull up resistor on DP line. This will cause the host to detect a device disconnect. 0: No soft disconnect generated. 1: Generate a soft disconnection.
0	RWKUP	Remote wakeup In suspend state, software can use this bit to generate a Remote wake up signal to inform host that it should resume the USB bus. 0: No remote wakeup signal generated. 1: Generate remote wakeup signal.

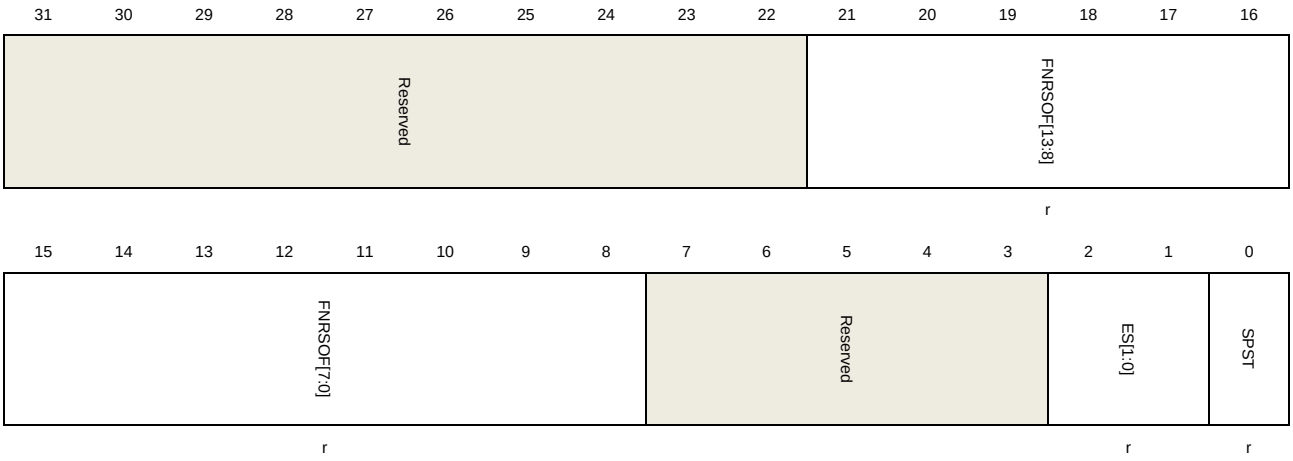
Device status register (USBFS_DSTAT)

Address offset: 0x0808

Reset value: 0x0000 0000

This register contains status and information of the USBFS in device mode.

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



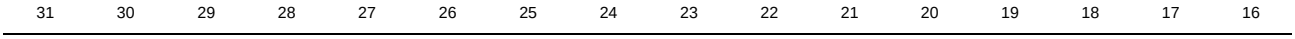
Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:8	FNRSOF[13:0]	The frame number of the received SOF. USBFS always update this field after receiving a SOF token
7:3	Reserved	Must be kept at reset value.
2:1	ES[1:0]	Enumerated speed This field reports the enumerated device speed. Read this field after the ENUMF flag in USBFS_GINTF register is triggered. 11: Full speed Others: reserved
0	SPST	Suspend status This bit reports whether device is in suspend state. 0: Device is not in suspend state. 1: Device is in suspend state.

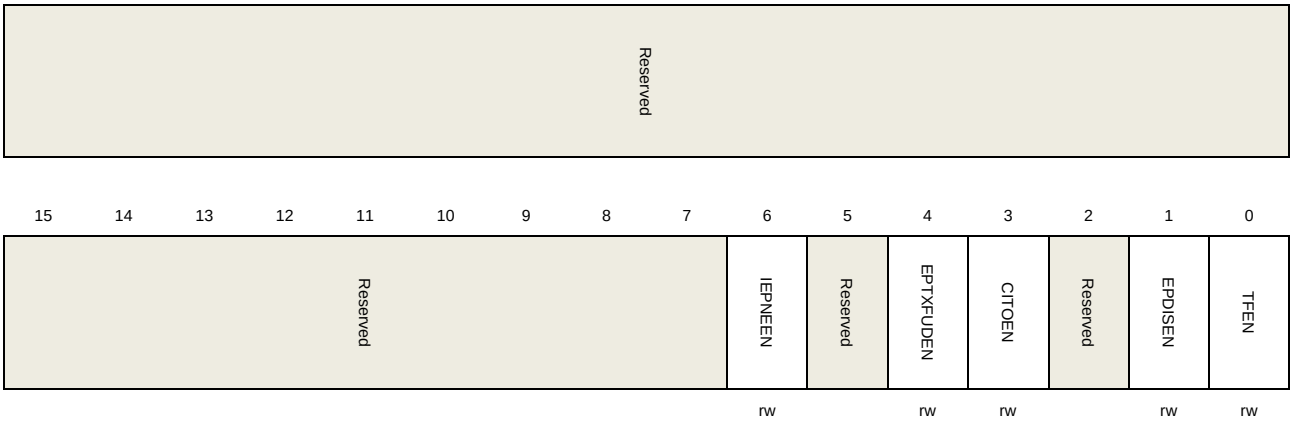
Device IN endpoint common interrupt enable register (USBFS_DIEPINTEN)

Address offset: 0x810
Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBFS_DIEPxINTF register. If a bit in this register is set by software, the corresponding bit in USBFS_DIEPxINTF register is able to trigger an endpoint interrupt in USBFS_DAEPINT register. The bits in this register are set and cleared by software.

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





Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	IEPNEEN	IN endpoint NAK effective interrupt enable bit 0: Disable IN endpoint NAK effective interrupt 1: Enable IN endpoint NAK effective interrupt
5	Reserved	Must be kept at reset value.
4	EPTXFUDEN	Endpoint Tx FIFO underrun interrupt enable bit 0: Disable endpoint Tx FIFO underrun interrupt 1: Enable endpoint Tx FIFO underrun interrupt
3	CITOEN	Control IN timeout interrupt enable bit 0: Disable control IN timeout interrupt 1: Enable control IN timeout interrupt
2	Reserved	Must be kept at reset value.
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable endpoint disabled interrupt 1: Enable endpoint disabled interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

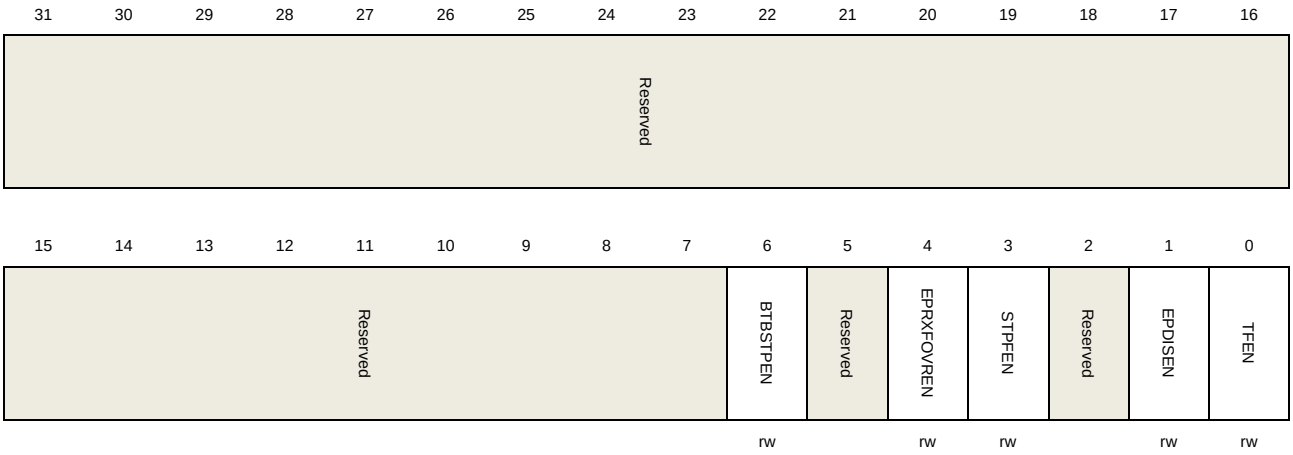
Device OUT endpoint common interrupt enable register (USBFS_DOEPINTEN)

Address offset: 0x0814

Reset value: 0x0000 0000

This register contains the interrupt enable bits for the flags in USBFS_DOEPxINTF register. If a bit in this register is set by software, the corresponding bit in USBFS_DOEPxINTF register is able to trigger an endpoint interrupt in USBFS_DAEPINT register. The bits in this register are set and cleared by software.

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



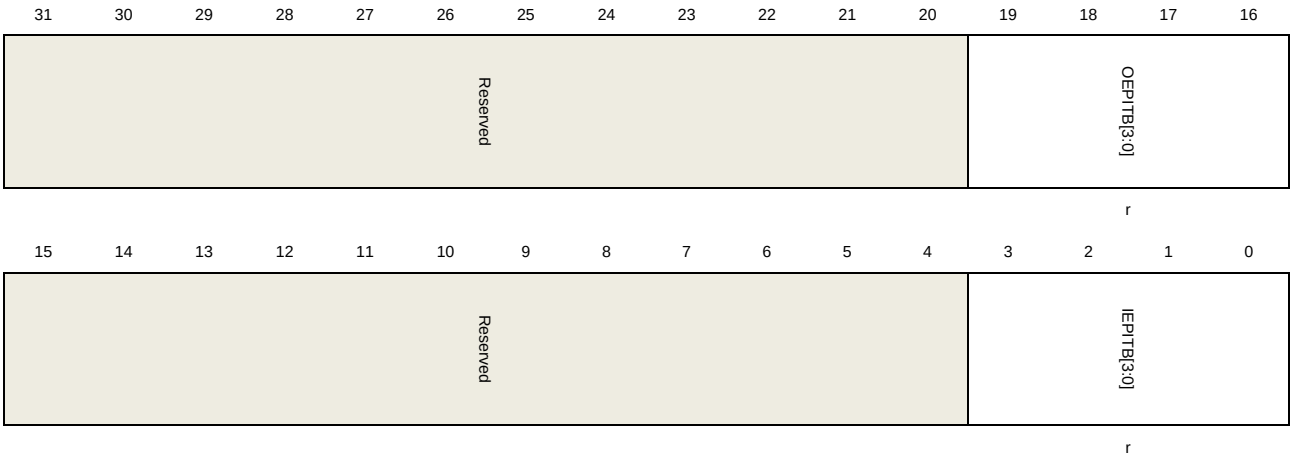
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	BTBSTPEN	Back-to-back SETUP packets (Only for control OUT endpoint) interrupt enable bit 0: Disable back-to-back SETUP packets interrupt 1: Enable back-to-back SETUP packets interrupt
5	Reserved	Must be kept at reset value.
4	EPRXFOVREN	Endpoint Rx FIFO overrun interrupt enable bit 0: Disable endpoint Rx FIFO overrun interrupt 1: Enable endpoint Rx FIFO overrun interrupt
3	STPFEN	SETUP phase finished (Only for control OUT endpoint) interrupt enable bit 0: Disable SETUP phase finished interrupt 1: Enable SETUP phase finished interrupt
2	Reserved	Must be kept at reset value.
1	EPDISEN	Endpoint disabled interrupt enable bit 0: Disable endpoint disabled interrupt 1: Enable endpoint disabled interrupt
0	TFEN	Transfer finished interrupt enable bit 0: Disable transfer finished interrupt 1: Enable transfer finished interrupt

Device all endpoints interrupt register (USBFS_DAEPINT)

Address offset: 0x0818
Reset value: 0x0000 0000

When an endpoint interrupt is triggered, USBFS sets corresponding bit in this register and software should read this register to know which endpoint is asserting an interrupt.

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



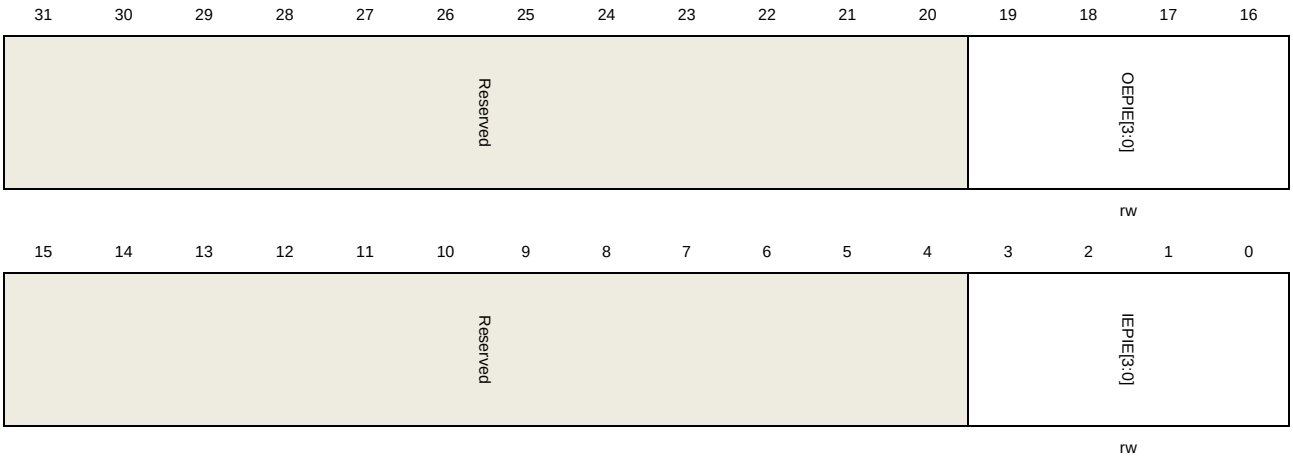
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	OEPITB[3:0]	Device all OUT endpoint interrupt bits Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 19 for OUT endpoint 3.
15:4	Reserved	Must be kept at reset value.
3:0	IEPITB[3:0]	Device all IN endpoint interrupt bits Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 3 for IN endpoint 3.

Device all endpoints interrupt enable register (USBFS_DAEPINTEN)

Address offset: 0x081C
Reset value: 0x0000 0000

This register can be used by software to enable or disable an endpoint's interrupt. Only the endpoint whose corresponding bit in this register is set is able to cause the endpoint interrupt flag OEPIF or IEPIF in USBFS_GINTF register.

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



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	OEPIE[3:0]	Out endpoint interrupt enable 0: Disable OUT endpoint-n interrupt 1: Enable OUT endpoint-n interrupt Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 19 for OUT endpoint 3.
15:4	Reserved	Must be kept at reset value.
3:0	IEPIE[3:0]	IN endpoint interrupt enable bits 0: Disable IN endpoint-n interrupt 1: Enable IN endpoint-n interrupt Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 3 for IN endpoint 3.

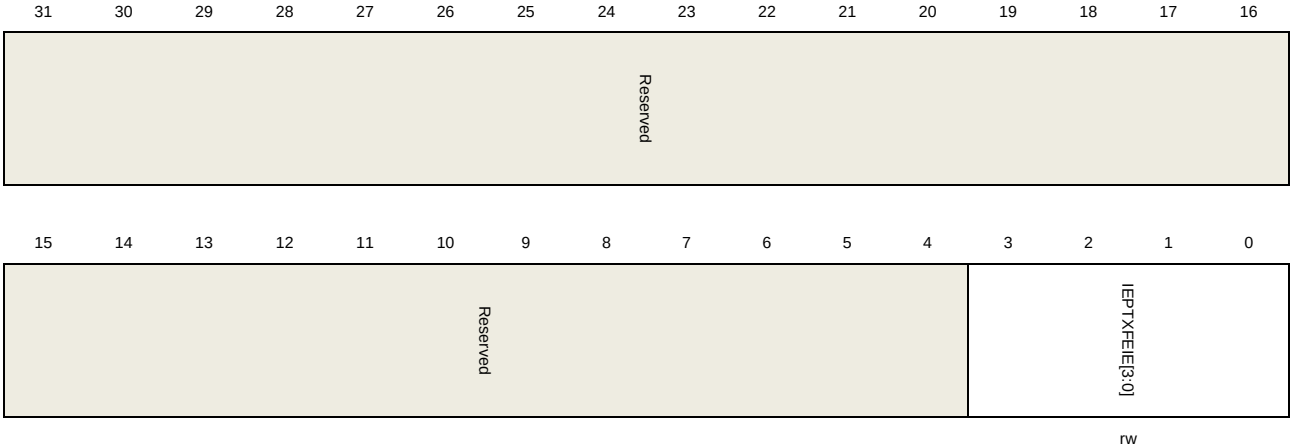
Device IN endpoint FIFO empty interrupt enable register (USBFS_DIEPFEINTEN)

Address offset: 0x0834

Reset value: 0x0000 0000

This register contains the enable bits for the Tx FIFO empty interrupts of IN endpoints.

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



Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	IEPTXFEIE[3:0]	IN endpoint Tx FIFO empty interrupt enable bits This field controls whether the TXFE bits in USBFS_DIEPxINTF registers are able to generate an endpoint interrupt bit in USBFS_DAEPINT register. Bit 0 for IN endpoint 0, bit 3 for IN endpoint 3

0: Disable FIFO empty interrupt

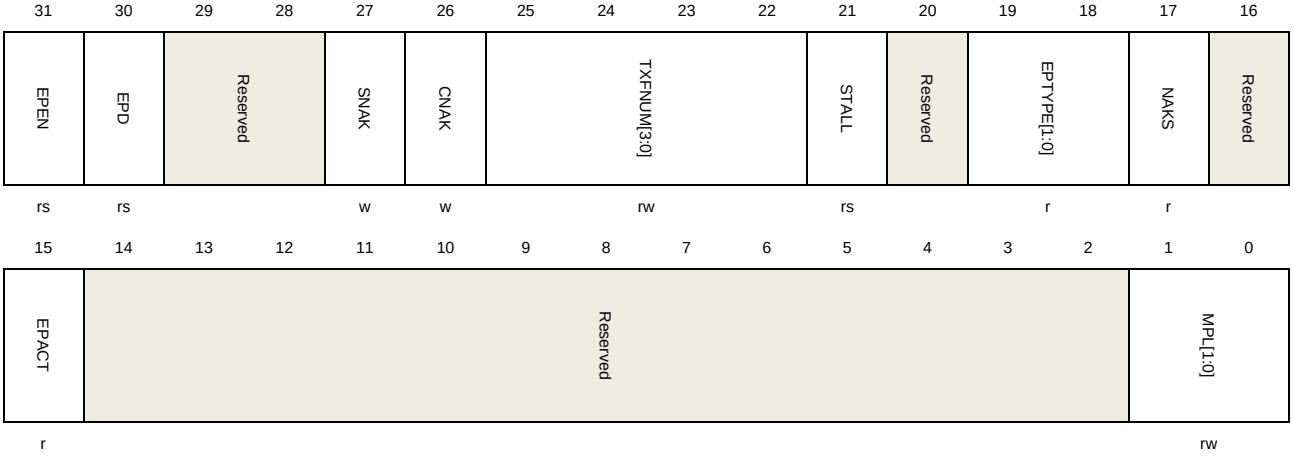
1: Enable FIFO empty interrupt

Device IN endpoint 0 control register (USBFS_DIEP0CTL)

Address offset: 0x0900

Reset value: 0x0000 8000

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



Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should following the operation guide to disable or enable an endpoint.
29:28	Reserved	Must be kept at reset value.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.
25:22	TXFNUM[3:0]	Tx FIFO number Defines the Tx FIFO number of IN endpoint 0.
21	STALL	STALL handshake Software can set this bit to make USBFS sends STALL handshake when receiving IN token. USBFS will clear this bit after a SETUP token is received on the corresponding OUT endpoint 0. This bit has a higher priority than NAKS bit in this

		register and GINS bit in USBFS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect.
20	Reserved	Must be kept at reset value.
19:18	EPTYPE[1:0]	Endpoint type This field is fixed to '00' for control endpoint.
17	NAKS	NAK status This bit controls the NAK status of USBFS when both STALL bit in this register and GINS bit in USBFS_DCTL register are cleared: 0: USBFS sends data or handshake packets according to the status of the endpoint's Tx FIFO. 1: USBFS always sends NAK handshake to the IN token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	Reserved	Must be kept at reset value.
15	EPACT	Endpoint active This field is fixed to '1' for endpoint 0.
14:2	Reserved	Must be kept at reset value.
1:0	MPL[1:0]	Maximum packet length This field defines the maximum packet length for a control data packet. As described in USB 2.0 protocol, there are 4 kinds of length for control transfers: 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes

Device IN endpoint-x control register (USBFS_DIEPxCTL) (x = 1..3, where x = endpoint_number)

Address offset: 0x0900 + (endpoint_number × 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
EPEN	EPD	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM[3:0]				STALL	Reserved	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w	rw				rw/rs		rw		r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

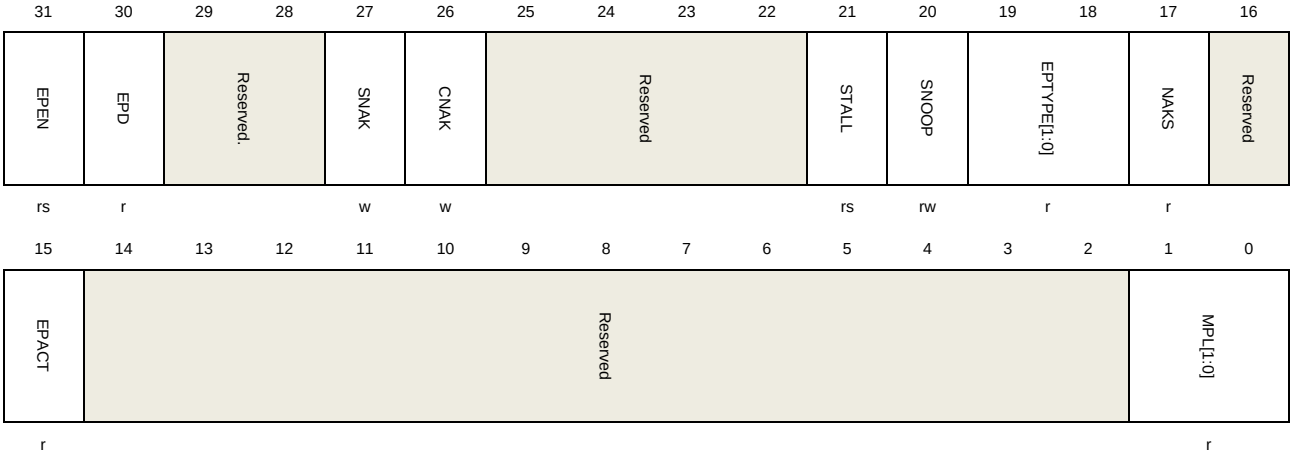
Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should following the operation guide to disable or enable an endpoint.
29	SODDFRM	Set odd frame (For isochronous IN endpoints) This bit has effect only if this is an isochronous IN endpoint. Software sets this bit to set EOFRM bit in this register.
	SD1PID	Set DATA1 PID (For interrupt/bulk IN endpoints) Software sets this bit to set DPID bit in this register.
28	SEVENFRM	Set even frame (For isochronous IN endpoints) Software sets this bit to clear EOFRM bit in this register.
	SD0PID	Set DATA0 PID (For interrupt/bulk IN endpoints) Software sets this bit to clear DPID bit in this register.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.
25:22	TXFNUM[3:0]	Tx FIFO number Defines the Tx FIFO number of this IN endpoint.
21	STALL	STALL handshake Software can set this bit to make USBFS sends STALL handshake when receiving IN token. This bit has a higher priority than NAKS bit in this register and GINS bit in USBFS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect. For control IN endpoint: Only USBFS can clear this bit when a SETUP token is received on the corresponding OUT endpoint. Software is not able to clear it. For interrupt or bulk IN endpoint:

		Only software can clear this bit
20	Reserved	Must be kept at reset value.
19:18	EPTYPE[1:0]	Endpoint type This field defines the transfer type of this endpoint: 00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	NAKS	NAK status This bit controls the NAK status of USBFS when both STALL bit in this register and GINS bit in USBFS_DCTL register are are cleared: 0: USBFS sends data or handshake packets according to the status of the endpoint's Tx FIFO. 1: USBFS always sends NAK handshake to the IN token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	EOFRM	Even/odd frame (For isochronous IN endpoints) For isochronous transfers, software can use this bit to control that USBFS only sends data packets for IN tokens in even or odd frames. If the parity of the current frame number doesn't match with this bit, USBFS only responses with a zero-length packet. 0: Only sends data in even frames 1: Only sends data in odd frames
	DPID	Endpoint data PID (For interrupt/bulk IN endpoints) There is a data PID toggle scheme in interrupt or bulk transfer. Set SD0PID to set this bit before a transfer starts and USBFS maintains this bit during transfers according to the data toggle scheme described in USB protocol. 0: Data packet's PID is DATA0 1: Data packet's PID is DATA1
15	EPACT	Endpoint active This bit controls whether this endpoint is active. If an endpoint is not active, it ignores all tokens and doesn't make any response.
14:11	Reserved	Must be kept at reset value.
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

Device OUT endpoint 0 control register (USBFS_DOEP0CTL)

Address offset: 0x0B00
Reset value: 0x0000 8000

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



Bits	Fields	Descriptions
31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable This bit is fixed to 0 for OUT endpoint 0.
29:28	Reserved	Must be kept at reset value.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register
25:22	Reserved	Must be kept at reset value.
21	STALL	STALL handshake Set this bit to make USBFS send STALL handshake during an OUT transaction. USBFS will clear this bit after a SETUP token is received on OUT endpoint 0. This bit has a higher priority than NAKS bit in this register, i.e. if both STALL and NAKS bits are set, the STALL bit takes effect.
20	SNOOP	Snoop mode This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBFS doesn’t check the received data packet’s CRC value. 0: Snoop mode disabled 1: Snoop mode enabled
19:18	EPTYPE[1:0]	Endpoint type This field is fixed to ‘00’ for control endpoint.
17	NAKS	NAK status

		This bit controls the NAK status of USBFS when both STALL bit in this register and GONS bit in USBFS_DCTL register are cleared: 0: USBFS sends data or handshake packets according to the status of the endpoint's Rx FIFO. 1: USBFS always sends NAK handshake for the OUT token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	Reserved	Must be kept at reset value.
15	EPACT	Endpoint active This field is fixed to '1' for endpoint 0.
14:2	Reserved	Must be kept at reset value.
1:0	MPL[1:0]	Maximum packet length This is a read-only field, and its value comes from the MPL field of USBFS_DIEP0CTL register: 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes

Device OUT endpoint-x control register (USBFS_DOEPxCTL) (x = 1..3, where x = endpoint_number)

Address offset: 0x0B00 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

The application uses this register to control the operations of each logical OUT endpoint other than OUT endpoint 0.

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

Bits	Fields	Descriptions

31	EPEN	Endpoint enable Set by the application and cleared by USBFS. 0: Endpoint disabled 1: Endpoint enabled Software should follow the operation guide to disable or enable an endpoint.
30	EPD	Endpoint disable Software can set this bit to disable the endpoint. Software should follow the operation guide to disable or enable an endpoint.
29	SODDFRM	Set odd frame (For isochronous OUT endpoints) This bit has effect only if this is an isochronous OUT endpoint. Software sets this bit to set EOFRM bit in this register.
	SD1PID	Set DATA1 PID (For interrupt/bulk OUT endpoints) Software sets this bit to set DPID bit in this register.
28	SEVENFRM	Set even frame (For isochronous OUT endpoints) Software sets this bit to clear EOFRM bit in this register.
	SD0PID	Set DATA0 PID (For interrupt/bulk OUT endpoints) Software sets this bit to clear DPID bit in this register.
27	SNAK	Set NAK Software sets this bit to set NAKS bit in this register.
26	CNAK	Clear NAK Software sets this bit to clear NAKS bit in this register.
25:22	Reserved	Must be kept at reset value.
21	STALL	STALL handshake Software can set this bit to make USBFS sends STALL handshake during an OUT transaction. This bit has a higher priority than NAKS bit in this register and GINAK in USBFS_DCTL register. If both STALL and NAKS bits are set, the STALL bit takes effect. For control OUT endpoint: Only USBFS can clear this bit when a SETUP token is received on the corresponding OUT endpoint. Software is not able to clear it. For interrupt or bulk OUT endpoint: Only software can clear this bit.
20	SNOOP	Snoop mode This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBFS doesn't check the received data packet's CRC value. 0: Snoop mode disabled 1: Snoop mode enabled
19:18	EPTYPE[1:0]	Endpoint type This field defines the transfer type of this endpoint:

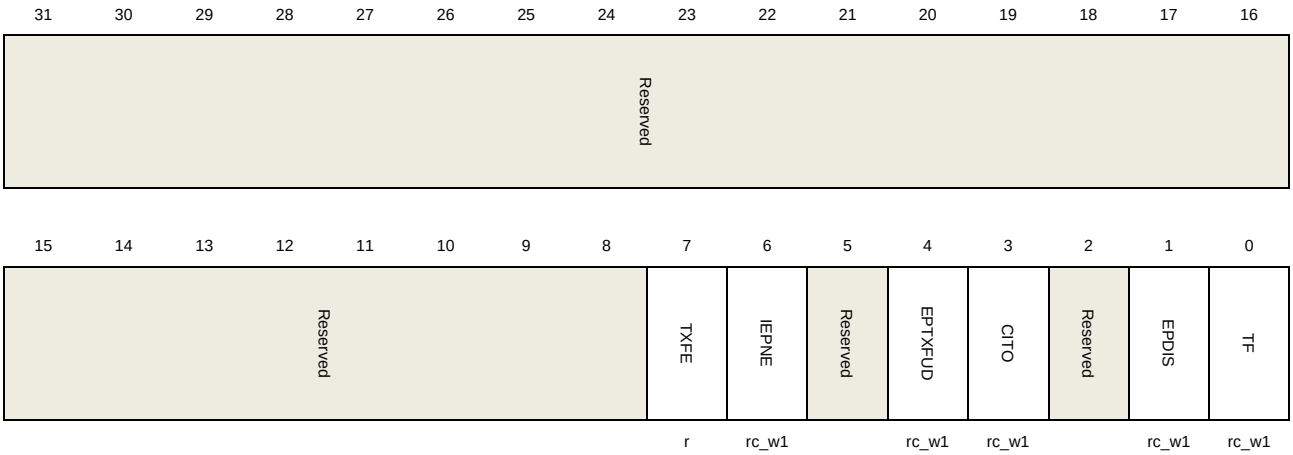
		00: Control 01: Isochronous 10: Bulk 11: Interrupt
17	NAKS	NAK status This bit controls the NAK status of USBFS when both STALL bit in this register and GONS bit in USBFS_DCTL register are cleared: 0: USBFS sends handshake packets according to the status of the endpoint's Rx FIFO. 1: USBFS always sends NAK handshake to the OUT token. This bit is read-only and software should use CNAK and SNAK in this register to control this bit.
16	EOFRM	Even/odd frame (For isochronous OUT endpoints) For isochronous transfers, software can use this bit to control that USBFS only receives data packets in even or odd frames. If the current frame number's parity doesn't match with this bit, USBFS just drops the data packet. 0: Only sends data in even frames 1: Only sends data in odd frames
	DPID	Endpoint data PID (For interrupt/bulk OUT endpoints) These is a data PID toggle scheme in interrupt or bulk transfer. Software should set SD0PID to set this bit before a transfer starts and USBFS maintains this bit during transfers following the data toggle scheme described in USB protocol. 0: Data packet's PID is DATA0 1: Data packet's PID is DATA1
15	EPACT	Endpoint active This bit controls whether this endpoint is active. If an endpoint is not active, it ignores all tokens and doesn't make any response.
14:11	Reserved	Must be kept at reset value.
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

Device IN endpoint-x interrupt flag register (USBFS_DIEPxINTF) (x = 0..3, where x = endpoint_number)

Address offset: 0x0908 + (endpoint_number × 0x20)
Reset value: 0x0000 0080

This register contains the status and events of an IN endpoint, when an IN endpoint interrupt occurs, read this register for the respective endpoint to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1 except the read-only TXFE bit.

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



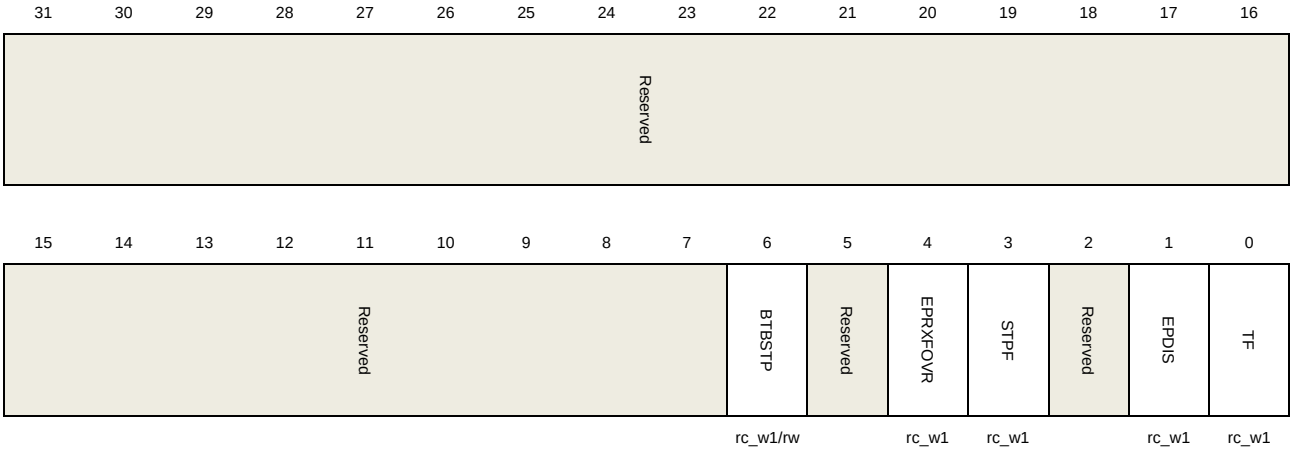
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TXFE	Transmit FIFO empty The Tx FIFO of this IN endpoint has reached the empty threshold value defined by TXFTH field in USBFS_GAHBCS register.
6	IEPNE	IN endpoint NAK effective The setting of SNAK bit in USBFS_DIEPxCTL register takes effect. This bit can be cleared either by writing 1 to it or by setting CNAK bit in USBFS_DIEPxCTL register.
5	Reserved	Must be kept at reset value.
4	EPTXFUD	Endpoint Tx FIFO underrun This flag is triggered if the Tx FIFO has no packet data when an IN token is incoming.
3	CITO	Control IN Timeout interrupt This flag is triggered if the device waiting for a handshake is timeout in a control IN transaction.
2	Reserved	Must be kept at reset value.
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled by the software's request.
0	TF	Transfer finished This flag is triggered when all the IN transactions assigned to this endpoint have been finished.

Device OUT endpoint-x interrupt flag register (USBFS_DOEPxINTF) (x = 0..3, where x = endpoint_number)

Address offset: 0x0B08 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

This register contains the status and events of an OUT endpoint, when an OUT endpoint interrupt occurs, read this register for the respective endpoint to know the source of the interrupt. The flag bits in this register are all set by hardware and cleared by writing 1.

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



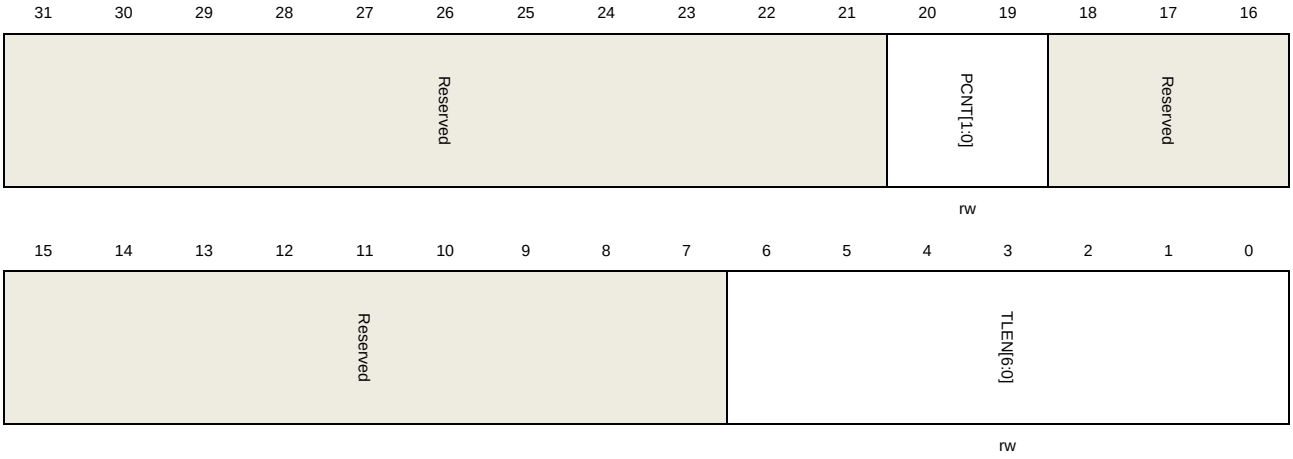
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	BTBSTP	Back-to-back SETUP packets (Only for control OUT endpoint) This flag is triggered when a control out endpoint has received more than 3 back-to-back setup packets.
5	Reserved	Must be kept at reset value.
4	EPRXFOVR	Endpoint Rx FIFO overrun This flag is triggered if the OUT endpoint's Rx FIFO has no enough space for a packet data when an OUT token is incoming. USBFS will drop the incoming OUT data packet and sends a NAK handshake in this case.
3	STPF	SETUP phase finished (Only for control OUT endpoint) This flag is triggered when a setup phase finished, i.e. USBFS receives an IN or OUT token after a setup token.
2	Reserved	Must be kept at reset value.
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled by the software's request.
0	TF	Transfer finished This flag is triggered when all the OUT transactions assigned to this endpoint have been finished.

Device IN endpoint 0 transfer length register (USBFS_DIEP0LEN)

Address offset: 0x0910

Reset value: 0x0000 0000

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



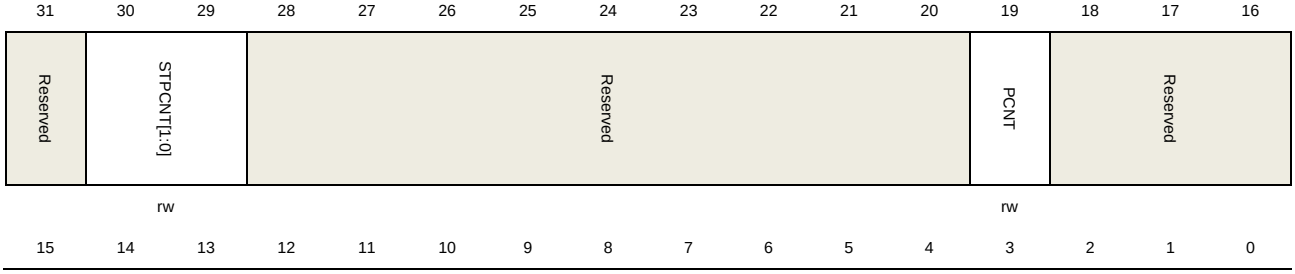
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:19	PCNT[1:0]	Packet count The number of data packets desired to be transmitted in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet transmission.
18:7	Reserved	Must be kept at reset value.
6:0	TLEN[6:0]	Transfer length The total data bytes` number of a transfer. This field is the total data bytes of all the data packets desired to be transmitted in an IN transfer. Program this field before the endpoint is enabled. When software successfully writes a packet into the endpoint's Tx FIFO, this field is decreased by the byte size of the packet.

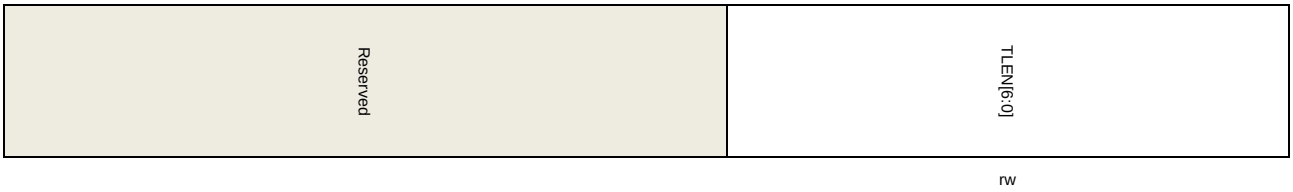
Device OUT endpoint 0 transfer length register (USBFS_DOEP0LEN)

Address offset: 0x0B10

Reset value: 0x0000 0000

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





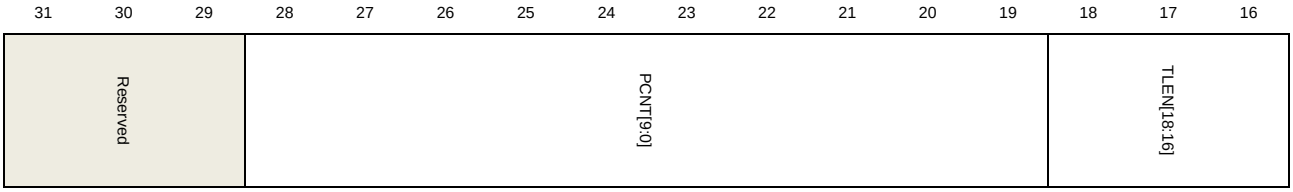
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	STPCNT[1:0]	SETUP packet count This field defines the maximum number of back-to-back SETUP packets this endpoint can accept. Program this field before setup transfers. Each time a back-to-back setup packet is received, USBFS decrease this field by one. When this field reaches zero, the BTBSTP flag in USBFS_DOEP0INTF register will be triggered. 00: 0 packet 01: 1 packet 10: 2 packets 11: 3 packets
28:20	Reserved	Must be kept at reset value.
19	PCNT	Packet count The number of data packets desired to receive in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet reception on bus.
18:7	Reserved	Must be kept at reset value.
6:0	TLEN[6:0]	Transfer length The total data bytes` number of a transfer. This field is the total data bytes of all the data packets desired to receive in an OUT transfer. Program this field before the endpoint is enabled. Each time software reads out a packet from the Rx FIFO, this field is decreased by the byte size of the packet.

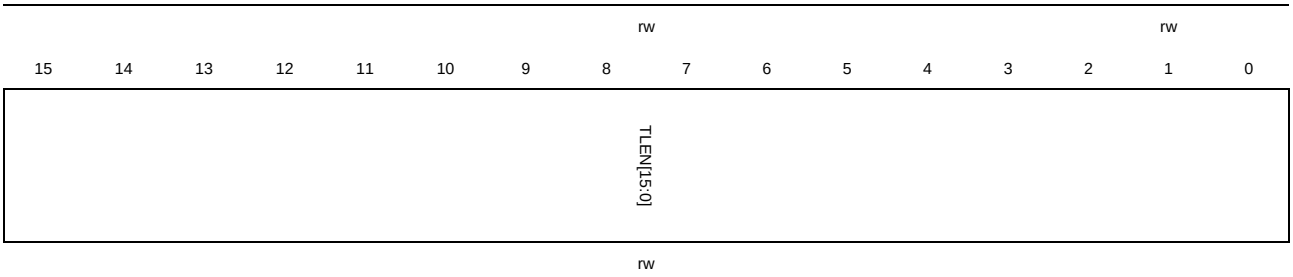
Device IN endpoint-x transfer length register (USBFS_DIEPxLEN) (x = 1..3, where x = endpoint_number)

Address offset: 0x910 + (endpoint_number × 0x20)

Reset value: 0x0000 0000

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



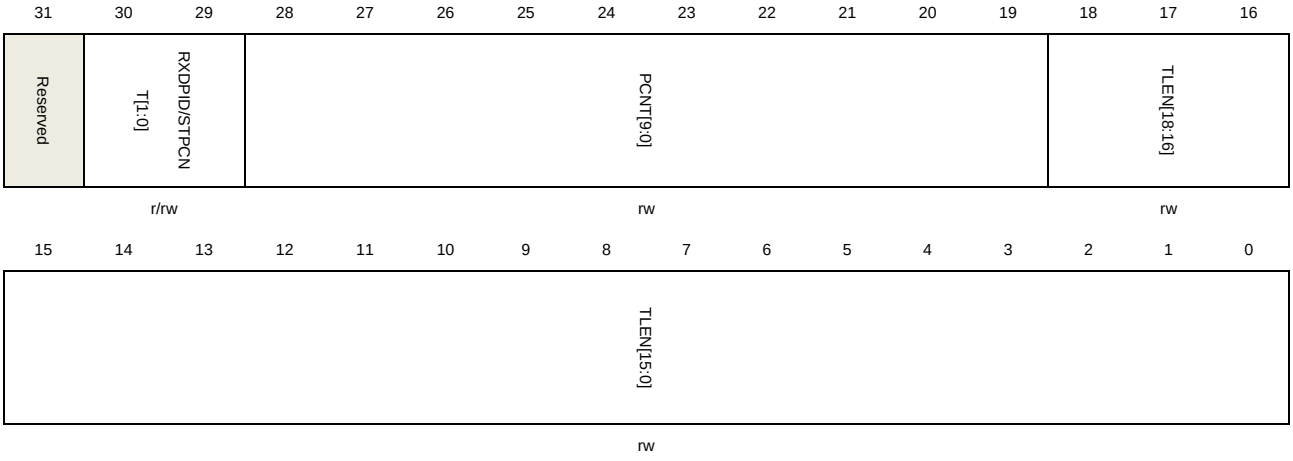


Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:19	PCNT[9:0]	Packet count The number of data packets desired to be transmitted in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet transmission.
18:0	TLEN[18:0]	Transfer length The total data bytes` number of a transfer. This field is the total data bytes of all the data packets desired to be transmitted in an IN transfer. Program this field before the endpoint is enabled. When software successfully writes a packet into the endpoint's Tx FIFO, this field is decreased by the byte size of the packet.

Device OUT endpoint-x transfer length register (USBFS_DOEPxLEN) (x = 1 .. 3, where x = endpoint_number)

Address offset: 0x0B10 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions

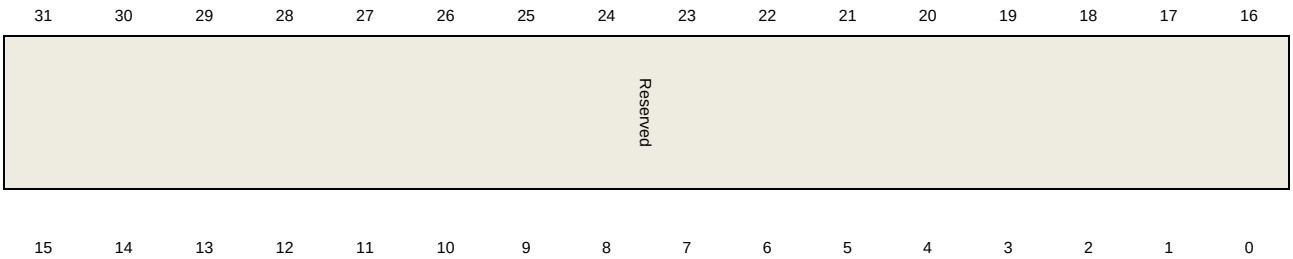
31	Reserved	Must be kept at reset value.
30:29	RXDPID[1:0]	Received data PID (For isochronous OUT endpoints) This field saves the PID of the latest received data packet on this endpoint. 00: DATA0 10: DATA1 Others: Reserved
	STPCNT[1:0]	SETUP packet count (For control OUT Endpoints.) This field defines the maximum number of back-to-back SETUP packets this endpoint can accept. Program this field before setup transfers. Each time a back-to-back setup packet is received, USBFS decrease this field by one. When this field reaches zero, the BTBSTP flag in USBFS_DOEPxINTF register will be triggered. 00: 0 packet 01: 1 packet 10: 2 packets 11: 3 packets
28:19	PCNT[9:0]	Packet count The number of data packets desired to receive in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each successful data packet reception on bus.
18:0	TLEN[18:0]	Transfer length The total data bytes` number of a transfer. This field is the total data bytes of all the data packets desired to receive in an OUT transfer. Program this field before the endpoint is enabled. Each time after software reads out a packet from the RxFIFO, this field is decreased by the byte size of the packet.

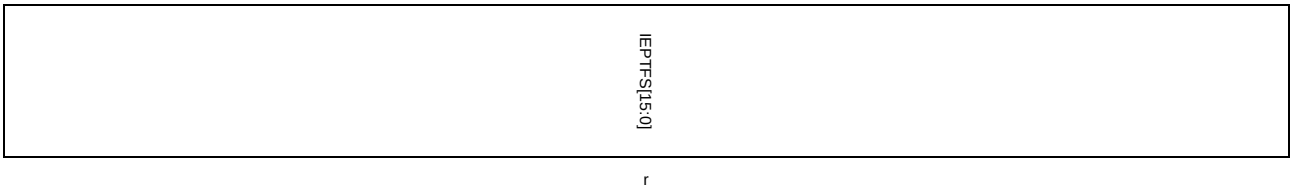
Device IN endpoint-x transmit FIFO status register (USBFS_DIEPxTFSTAT) (x = 0 .. 3, where x = endpoint_number)

Address offset: 0x0918 + (endpoint_number × 0x20)
Reset value: 0x0000 0200

This register contains the information of each endpoint's Tx FIFO.

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





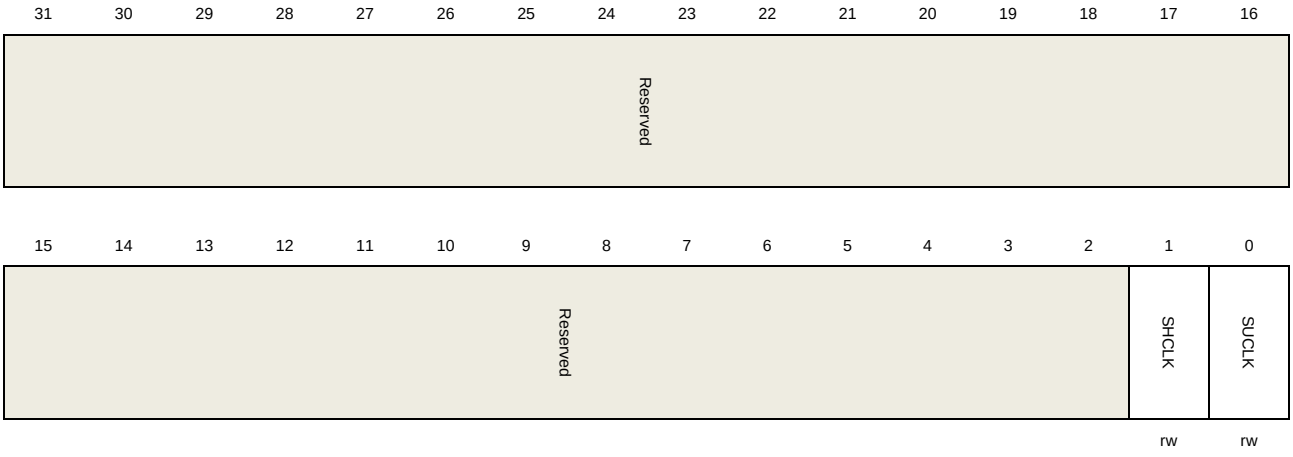
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	IEPTFS[15:0]	IN endpoint's Tx FIFO space remaining IN endpoint's Tx FIFO space remaining in 32-bit words: 0: FIFO is full 1: 1 word available … n: n words available

28.7.4. Power and clock control register (USBFS_PWRCLKCTL)

Address offset: 0x0E00

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	SHCLK	Stop HCLK Stop the HCLK to save power. 0: HCLK is not stopped 1: HCLK is stopped
0	SUCLK	Stop the USB clock Stop the USB clock to save power. 0: USB clock is not stopped

1: USB clock is stopped

29. Appendix

29.1. List of abbreviations used in register

Table 29-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read-only (r)	Software can only read this bit.
write-only (w)	Software can only write to this bit.
read/write (rw)	Software can read and write to this bit.
read/write-once (rwo)	Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value.
read/write-1-to-clear (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
read/write-0-to-clear (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
read/write-to-clear (rc_w)	Software can read as well as clear this bit by writing. Writing 0 or 1 has the same effect to this bit.
write-to-clear (c_w)	Software can only clear this bit by writing. Writing 0 or 1 has the same effect to this bit.
write-1-to-clear (c_w1)	Software can only clear this bit by writing 1. Writing 0 has no effect on the bit value.
readable/read-to-clear (rc_r)	Software can read this bit. Reading this bit automatically clears it to '0'. Writing '0' has no effect on the bit value.
readable/settable (rs)	Software can read as well as set this bit to 1. Writing 0 has no effect on the bit value.
readable/read-to-set (rs_r)	Software can read this bit, and set this bit by reading. Writing has no effect on the bit value.
writable/read-to-set (ws_r)	Software can write this bit, and set this bit by reading.
readable/write-to-set (rs_w)	Software can read this bit, and set this bit by writing. Writing 0 or 1 has the same effect to this bit.
write-to-set (s_w)	Software can only set this bit by writing. Writing 0 or 1 has the same effect to this bit.
readable/write-0-to-set (rs_w0)	Software can read this bit, and set this bit by writing 0. Writing 1 has no effect on the bit value.
read-only/write-to-trigger (rt_w)	Software can read this bit. Writing 0 or 1 triggers an event but has no effect on the bit value.
read-only/write-1-to-trigger (rt_w1)	Software can only read to this bit. Writing 1 triggers the event but has no effect on the bit value.
read-only/write-0-to-trigger (rt_w0)	Software can only read to this bit. Writing 0 triggers the event but has no effect on the bit value.

abbreviations for registers	Descriptions
toggle (t)	Software can toggle this bit by writing 1. Writing 0 has no effect.

29.2. List of terms

Table 29-2. 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.

29.3. Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

30. Revision history

Table 30-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Oct.31, 2025
1.1	1. Add ***Figure 8-2. Clock tree (for GD32F502xx)***. 2. Add description for GD32F502xx in ***System architecture*** section. 3. Modify the notice in chapter ***GPIO pin configuration*** section. 4. Modify description in chapter ***OTP block programming*** section. 5. Update ***Table 5-8. OTP3 lock and data***. 6. Update ***Figure 1-2. GD32F50x series system architecture***. 7. Update the description in ***Read operations*** section. 8. Delete title ***Data type*** in ***CAU data type and initialization vectors*** section. 9. Modify description of ALGM[2:0] in ***Control register (CAU_CTL)***. 10. Modify the description of no waiting time area of flash in ***Introduction*** and ***Characteristics*** section. 11. Modify the description in ***JTAG daisy chained structure*** section. 12. Modify the description of NERR in ***USART receiver*** section from “If the three samples of any bit of a frame are not the same, whatever it is a start bit, data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame.” to “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.” 13. Update ***Table 11-1. Trigger input bit fields selection***. 14. Add note for frequencies switching in RCU ***Overview*** section.	Dec.2, 2025
1.2	1. Modify the description of RT[23:0] in USART_RT register. 2. Modify the description in ***chapter 23.3.10***. 3. Modify figures in ***chapter I2C***. 4 Modify register description in ***chapter 28.7.1*** and ***chapter 28.7.3***. 5. Modify the description of ***chapter 25.3.10***. Add note “When SMBus acts as a slave transmitter and uses the PEC function, BYTENUM[7:0] must be configured before detecting the start signal.”	Jan.20, 2026

Revision No.	Description	Date
	6. Modify the description of ***chapter 25.3.8***. Add note "BYTENUM[7:0] can not be configured as 0." 7. Modify the description of BYTENUM[7:0], HEAD10R, ADD10EN, TRDIR and SADDRESS in I2C_CTL1 register. 8. Delete TRACE_MODE[1:0] in DBG_CTL register. 9. Modify the description of ***chapter 14.1.3***. Add note "There must be an interval of 7or more IRC32K clock cycles between two sonsecutive watchdog reload operations." 10. Modify the description of ***chapter 26.3.7***. Add note "For detailed explanations of hard synchronization and resynchronization, please refer to the ISO 11898 standard." 11. Modify ***Table 4-3. Request multiplexer input mapping***. Change TIMER3_CH0 to TIMER2_CH0. 12. Modify the chapter order of DBG and CRC, ADC/DAC and CMP.	
1.3	1. Modify the access attributes of the ***CRC DATA register***. 2. Delete the description stating that ADC_IDATA register cannot be used for DMA transfer ***in chapter 20.5***. 3. Delete "All TIMERx_ITI (x=0,…,4,7, 15,16) can only select trigger input sources from TIMER and CK_OUT. Among them," ***in chapter 11.4***. 4. Modify the description of APB1PSC ***in chapter 8.3.2***. 5. Change CK_SYS to CK_AHB ***in chapter 5.3.12***.	May.11, 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