GigaDevice Semiconductor Inc.

GD32E53x
Arm® Cortex®-M33 32-bit MCU

<For GD32E533xx, GD32E535xx, GD32E537xx, GD32E53Pxx >

User Manual

Revision 1.0

(Nov. 2026)

Table of Contents

Table of Contents **2**

List of Figures **24**

List of Tables **34**

1. System and memory architecture **38**

1.1. Arm® Cortex®-M33 processor **38**

1.2. System architecture **39**

1.3. Memory map **41**

1.3.1. On-chip SRAM memory 46

1.3.2. On-chip flash memory overview 48

1.4. Boot configuration **48**

1.5. System configuration registers **50**

1.5.1. Configuration register 0 (SYSCFG_CFG0) 50

1.5.2. Configuration register 1 (SYSCFG_CFG1) 50

1.5.3. Lockup control register (SYSCFG_LKCTL) 51

1.5.4. Bus timeout register (SYSCFG_BUSTO) 52

1.5.5. SRAM ECC error status register 0 (SYSCFG_ECCS0) 53

1.5.6. FPU interrupt enable register (SYSCFG_FPUINTEN) 54

1.5.7. SRAM write protection register 0 (SYSCFG_SRAMWP0) 55

1.5.8. SRAM ECC error status register 1 (SYSCFG_ECCS1) 56

1.5.9. SRAM ECC configuration register 0 (SYSCFG_ECCCFG0) 57

1.5.10. ECC configuration register 1 (SYSCFG_ECCCFG1) 58

1.5.11. SRAM ECC configuration register 2 (SYSCFG_ECCCFG2) 59

1.5.12. SRAM ECC configuration register 3 (SYSCFG_ECCCFG3) 60

1.5.13. Bus timeout status register (SYSCFG_BUSTOSTAT) 60

1.5.14. SRAM write protection register1 (SYSCFG_SRAMWP1) 61

1.5.15. Timer input selection register (SYSCFG_TIMERCISEL) 61

1.5.16. SRAM ECC configuration register 4 (SYSCFG_ECCCFG4) 63

1.5.17. SRAM ECC configuration register5 (SYSCFG_ECCCFG5) 64

1.5.18. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=0, 7, 19) 64

1.5.19. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=0, 7, 19) 66

1.5.20. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=0, 7, 19) 67

1.5.21. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=1, 2, 3, 4) 68

1.5.22. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=1, 2, 3, 4) 70

1.5.23. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=1, 2, 3, 4) 71

1.5.24. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=9, 10, 12, 13) 72

1.5.25. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=9, 10, 12, 13) 74

1.5.26. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=9, 10, 12, 13) 74

1.5.27. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=14, 40~44) 75
1.5.28. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=14, 40~44) 77
1.5.29. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=14, 40~44) 78
1.6. Device electronic signature 79
1.6.1. Memory density information 79
1.6.2. Unique device ID (96 bits) 79
2. Flash memory controller (FMC) 81
2.1. Overview 81
2.2. Characteristics 81
2.3. Function overview 81
2.3.1. Flash memory architecture 81
2.3.2. Error checking and correcting (ECC) 82
2.3.3. Read operations 83
2.3.4. Unlock the FMC_CTL register and FMC_OBCTL register 85
2.3.5. Page erase 85
2.3.6. Mass erase 87
2.3.7. Main flash programming 88
2.3.8. OTP programming 90
2.3.9. Option bytes 94
2.3.10. Erase/program protection(Write Protection) 99
2.3.11. Security protection 100
2.3.12. Secure user area (SCR) 101
2.4. Register definition 102
2.4.1. Wait state register (FMC_WS) 102
2.4.2. Unlock key register (FMC_KEY) 103
2.4.3. Option byte unlock key register (FMC_OBKEY) 103
2.4.4. Status register (FMC_STAT) 103
2.4.5. Control register (FMC_CTL) 105
2.4.6. Address register (FMC_ADDR) 106
2.4.7. ECC control register(FMC_ECCCTL) 107
2.4.8. Offset region register (FMC_OFRG) 107
2.4.9. Offset value register (FMC_OFVR) 108
2.4.10. SCR control register (FMC_SCRCTL) 108
2.4.11. Option byte control register (FMC_OBCTL) 109
2.4.12. SCR write lock register (FMC_SCRWLK) 111
2.4.13. Erase/Program protection register (FMC_WPx) (x = 0…3) 111
2.4.14. OTP DATA1 low block lock register (FMC_OTPDATA1_LLK) 112
2.4.15. OTP DATA1 high block lock register (FMC_OTPDATA1_HLK) 112
2.4.16. OTP DATA2 low block lock register (FMC_OTPDATA2_LLK) 113
2.4.17. OTP DATA2 high block lock register (FMC_OTPDATA2_HLK) 113
2.4.18. OTP DATA2 low read control register (FMC_OTPDATA2_LR_CTL) 114

2.4.19. OTP DATA2 high read control register (FMC_OTPDATA2_HR_CTL) 114
2.4.20. ECC control and status register (FMC_ECCCS) 115
2.4.21. ECC address register(FMC_ECCADDR) 116
2.4.22. Product ID register 0 (FMC_PID0) 116
2.4.23. Product ID register 1 (FMC_PID1) 117
3. Instruction cache (ICACHE) 118
3.1. Introduction 118
3.2. Characteristics 118
3.3. Function overview 118
3.3.1. ICACHE initialization 119
3.3.2. Paired master cache 119
3.3.3. ICACHE TAG memory 120
3.3.4. Address remapping 120
3.3.5. Cacheable and uncacheable access 122
3.3.6. ICACHE performance monitoring 123
3.3.7. ICACHE interrupts 124
3.4. Register definition 125
3.4.1. Control register (ICACHE_CTL) 125
3.4.2. Status register (ICACHE_STAT) 126
3.4.3. Interrupt enable register (ICACHE_INTEN) 126
3.4.4. Flag clear register (ICACHE_FC) 127
3.4.5. Hit monitor counter register (ICACHE_HMC) 127
3.4.6. Miss monitor counter register (ICACHE_MMC) 128
3.4.7. Configuration register (ICACHE_CFGx) (x = 0...3) 128
4. Data cache (DCACHE) 130
4.1. Introduction 130
4.2. Characteristics 130
4.3. Function overview 130
4.3.1. DCACHE initialization 131
4.3.2. DCACHE TAG memory 132
4.3.3. Cacheable and uncacheable access 133
4.3.4. Cache maintenance operations 136
4.3.5. DCACHE performance monitoring 138
4.4. Register definition 139
4.4.1. Control register (DCACHE_CTL) 139
4.4.2. Status register (DCACHE_STAT) 140
4.4.3. Interrupt enable register (DCACHE_INTEN) 141
4.4.4. Flag clear register (DCACHE_FC) 142
4.4.5. Read-hit monitor counter register (DCACHE_RHMC) 143
4.4.6. Read-miss monitor counter register (DCACHE_RMMC) 143

4.4.7. Write-hit monitor counter register (DCACHE_WHMC) 143
4.4.8. Write-miss monitor counter register (DCACHE_WMMC) 144
4.4.9. Maintenance operation start address register (DCACHE_MOSADDR) 144
4.4.10. Maintenance operation end address register (DCACHE_MOEADDR) 144
5. Power management unit (PMU) 146
5.1. Overview 146
5.2. Characteristics 146
5.3. Function overview 147
5.3.1. Battery backup domain 148
5.3.2. Backup SRAM 149
5.3.3. VDD / VDDA power domain 150
5.3.4. V_{CORE} power domain 154
5.3.5. Power saving modes 156
5.4. Register definition 159
5.4.1. Control register 0 (PMU_CTL0) 159
5.4.2. Control and status register 0 (PMU_CS0) 161
5.4.3. Control register 1 (PMU_CTL1) 164
5.4.4. Control and status register 1 (PMU_CS1) 166
5.4.5. Control register 2 (PMU_CTL2) 166
5.4.6. Control register 3 (PMU_CTL3) 167
5.4.7. Parameter register (PMU_PAR) 169
6. Reset and clock unit (RCU) 170
6.1. Reset control unit (RCTL) 170
6.1.1. Overview 170
6.1.2. Function overview 170
6.2. Clock control unit (CCTL) 171
6.2.1. Overview 171
6.2.2. Characteristics 174
6.2.3. Function overview 174
6.3. Register definition 179
6.3.1. Control register (RCU_CTL) 179
6.3.2. Clock configuration register 0 (RCU_CFG0) 181
6.3.3. Clock interrupt register (RCU_INT) 184
6.3.4. APB2 reset register (RCU_APB2RST) 187
6.3.5. APB1 reset register (RCU_APB1RST) 190
6.3.6. AHB enable register (RCU_AHBEN) 194
6.3.7. APB2 enable register (RCU_APB2EN) 195
6.3.8. APB1 enable register (RCU_APB1EN) 199
6.3.9. Backup domain control register (RCU_BDCTL) 202
6.3.10. Reset source/clock register (RCU_RSTSCK) 204

6.3.11. AHB reset register (RCU_AHBRST) 206
6.3.12. Clock configuration register 1 (RCU_CFG1) 207
6.3.13. Deep-sleep mode voltage register (RCU_DSV) 209
6.3.14. Clock frequency monitor configuration register 0 (RCU_CKFMCFG0) 210
6.3.15. Clock frequency monitor configuration register 1 (RCU_CKFMCFG1) 211
6.3.16. Clock frequency monitor configuration register 2 (RCU_CKFMCFG2) 212
6.3.17. Clock frequency monitor configuration register 3 (RCU_CKFMCFG3) 213
6.3.18. Clock frequency monitor configuration register 4 (RCU_CKFMCFG4) 214
6.3.19. Additional clock control register (RCU_ADDCTL) 215
6.3.20. Additional clock interrupt register (RCU_ADDINT) 216
6.3.21. PLL clock spread spectrum control register (RCU_PLLSSCTL) 217
6.3.22. Clock configuration register 2 (RCU_CFG2) 218
6.3.23. APB1 additional reset register (RCU_ADDAPB1RST) 219
6.3.24. APB1 additional enable register (RCU_ADDAPB1EN) 220
6.3.25. LOCK register (RCU_LOCK) 220

7. Clock trim controller (CTC) 222

7.1. Overview 222

7.2. Characteristics 222

7.3. Function overview 222

7.3.1. REF sync pulse generator 223
7.3.2. CTC trim counter 223
7.3.3. Frequency evaluation and automatically trim process 224
7.3.4. Software program guide 225

7.4. Register definition 227

7.4.1. Control register 0 (CTC_CTL0) 227
7.4.2. Control register 1 (CTC_CTL1) 228
7.4.3. Status register (CTC_STAT) 229
7.4.4. Interrupt clear register (CTC_INTC) 231

8. Interrupt/event controller (EXTI) 233

8.1. Overview 233

8.2. Characteristics 233

8.3. Interrupts function overview 233

8.4. External interrupt and event (EXTI) block diagram 238

8.5. External Interrupt and event function overview 238

8.5.1. Hardware trigger 239
8.5.2. Software trigger 239

8.6. EXTI Register 241

8.6.1. Interrupt enable register (EXTI_INTEN) 241
8.6.2. Event enable register (EXTI_EVEN) 241

8.6.3. Rising edge trigger enable register (EXTI_RTEN) 242
8.6.4. Falling edge trigger enable register (EXTI_FTEN) 242
8.6.5. Software interrupt event register (EXTI_SWIEV) 242
8.6.6. Pending register (EXTI_PD) 243
9. Trigger selection controller (TRIGSEL) 244
9.1. Overview 244
9.2. Characteristics 244
9.3. Function overview 244
9.4. Internal connections 245
9.5. Register definition 254
9.5.1. Trigger selection for EXTOUT register 0 (TRIGSEL_EXTOUT_0) 254
9.5.2. Trigger selection for EXTOUT register 1 (TRIGSEL_EXTOUT_1) 254
9.5.3. Trigger selection for EXTOUT register 2 (TRIGSEL_EXTOUT_2) 255
9.5.4. Trigger selection for EXTOUT register 3 (TRIGSEL_EXTOUT_3) 256
9.5.5. Trigger selection for TIMER0_ITI register (TRIGSEL_TIMER0ITI) 256
9.5.6. Trigger selection for TIMER1_ITI register (TRIGSEL_TIMER1ITI) 257
9.5.7. Trigger selection for TIMER2_ITI register (TRIGSEL_TIMER2ITI) 258
9.5.8. Trigger selection for TIMER3_ITI register (TRIGSEL_TIMER3ITI) 258
9.5.9. Trigger selection for TIMER4_ITI register (TRIGSEL_TIMER4ITI) 259
9.5.10. Trigger selection for TIMER7_ITI register (TRIGSEL_TIMER7ITI) 259
9.5.11. Trigger selection for TIMER14_ITI register (TRIGSEL_TIMER14ITI) 260
9.5.12. Trigger selection for TIMER19_ITI register (TRIGSEL_TIMER19ITI) 261
9.5.13. Trigger selection for TIMER40_ITI register (TRIGSEL_TIMER40ITI) 261
9.5.14. Trigger selection for TIMER41_ITI register (TRIGSEL_TIMER41ITI) 262
9.5.15. Trigger selection for TIMER42_ITI register (TRIGSEL_TIMER42ITI) 262
9.5.16. Trigger selection for TIMER43_ITI register (TRIGSEL_TIMER43ITI) 263
9.5.17. Trigger selection for TIMER44_ITI register (TRIGSEL_TIMER44ITI) 264
9.5.18. Trigger selection for DAC register0 (TRIGSEL_DAC0OUT0) 264
9.5.19. Trigger selection for DAC register1 (TRIGSEL_DAC0OUT1) 265
9.5.20. Trigger selection for ADC0_ROUTRG register (TRIGSEL_ADC0_ROUTRG) 265
9.5.21. Trigger selection for ADC0_INSTRG register (TRIGSEL_ADC0_INSTRG) 266
9.5.22. Trigger selection for ADC1_ROUTRG register (TRIGSEL_ADC1_ROUTRG) 267
9.5.23. Trigger selection for ADC1_INSTRG register (TRIGSEL_ADC1_INSTRG) 267
9.5.24. Trigger selection for ADC2_ROUTRG register (TRIGSEL_ADC2_ROUTRG) 268
9.5.25. Trigger selection for ADC2_INSTRG register (TRIGSEL_ADC2_INSTRG) 269
9.5.26. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN) 269
9.5.27. Trigger selection for TIMER0_CHBRKIN register (TRIGSEL_TIMER0CHBRKIN) 270
9.5.28. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN) 271
9.5.29. Trigger selection for TIMER7_CHBRKIN register (TRIGSEL_TIMER7CHBRKIN) 271
9.5.30. Trigger selection for TIMER19_BRKIN register (TRIGSEL_TIMER19BRKIN) 272
9.5.31. Trigger selection for TIMER19_CHBRKIN register (TRIGSEL_TIMER19CHBRKIN) 273

9.5.32. Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN) 274
9.5.33. Trigger selection for TIMER40_BRKIN register (TRIGSEL_TIMER40BRKIN) 274
9.5.34. Trigger selection for TIMER41_BRKIN register (TRIGSEL_TIMER41BRKIN) 275
9.5.35. Trigger selection for TIMER42_BRKIN register (TRIGSEL_TIMER42BRKIN) 276
9.5.36. Trigger selection for TIMER43_BRKIN register (TRIGSEL_TIMER43BRKIN) 276
9.5.37. Trigger selection for TIMER44_BRKIN register (TRIGSEL_TIMER44BRKIN) 277
9.5.38. Trigger selection for TIMER9_ITI register (TRIGSEL_TIMER9ITI) 277
9.5.39. Trigger selection for TIMER10_ITI register (TRIGSEL_TIMER10ITI) 278
9.5.40. Trigger selection for TIMER12_ITI register (TRIGSEL_TIMER12ITI) 279
9.5.41. Trigger selection for TIMER13_ITI register (TRIGSEL_TIMER13ITI) 280

10. General-purpose and alternate-function I/Os (GPIO and AFIO) 281

10.1. Overview 281

10.2. Characteristics 281

10.3. Function overview 281

10.3.1. GPIO pin configuration 283
10.3.2. External interrupt/event lines 284
10.3.3. Alternate functions (AF) 284
10.3.4. Input configuration 284
10.3.5. Output configuration 284
10.3.6. Analog configuration 285
10.3.7. Alternate function (AF) configuration 285
10.3.8. GPIO locking function 286
10.3.9. GPIO I/O compensation cell 286
10.3.10. Input filtering 287
10.3.11. CLK pins AF remapping 289

10.4. Register definition 291

10.4.1. Port control register (GPIOx_CTL, x = A...E) 291
10.4.2. Port output mode register (GPIOx_OMODE, x = A...E) 293
10.4.3. Port output speed register (GPIOx_OSPD, x = A...E) 294
10.4.4. Port pull-up / pull-down register (GPIOx_PUD, x = A...E) 296
10.4.5. Port input status register (GPIOx_ISTAT, x = A...E) 298
10.4.6. Port output control register (GPIOx_OCTL, x = A...E) 298
10.4.7. Port bit operate register (GPIOx_BOP, x = A...E) 299
10.4.8. Port configuration lock register (GPIOx_LOCK, x = A...E) 299
10.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x = A...E) 300
10.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x = A...E) 301
10.4.11. Bit clear register (GPIOx_BC, x = A...E) 302
10.4.12. Input filtering register (GPIOx_IFL, x=A..E) 303
10.4.13. Input filtering type register (GPIOx_IFTP, x=A…E) 303
10.4.14. EXTI sources selection register 0 (AFIO_EXTISS0) 305
10.4.15. EXTI sources selection register 1 (AFIO_EXTISS1) 306
10.4.16. EXTI sources selection register 2 (AFIO_EXTISS2) 307

10.4.17. EXTI sources selection register 3 (AFIO_EXTISS3) 308
10.4.18. IO compensation control register (AFIO_CPSCTL) 309
10.4.19. PD01 remap register (AFIO_PD01) 310
11. Cyclic redundancy checks management unit (CRC) 311
11.1. Overview 311
11.2. Characteristics 311
11.3. Function overview 312
11.4. Register definition 313
11.4.1. Data register (CRC_DATA) 313
11.4.2. Free data register (CRC_FDATA) 313
11.4.3. Control register (CRC_CTL) 314
11.4.4. Initialization data register (CRC_IDATA) 314
11.4.5. Polynomial register (CRC_POLY) 315
12. True random number generator (TRNG) 316
12.1. Overview 316
12.2. Characteristics 316
12.3. Function overview 316
12.3.1. Operation flow 317
12.3.2. Error flags 317
12.4. Register definition 318
12.4.1. Control register (TRNG_CTL) 318
12.4.2. Status register (TRNG_STAT) 318
12.4.3. Data register (TRNG_DATA) 319
13. Cryptographic Acceleration Unit (CAU) 321
13.1. Overview 321
13.2. Characteristics 321
13.3. CAU data type and initialization vectors 321
13.3.1. Data type 321
13.4. Cryptographic acceleration processor 323
13.4.1. AES cryptographic acceleration processor 323
13.5. Operating modes 325
13.6. CAU DMA interface 326
13.7. CAU interrupts 326
13.8. Register definition 328
13.8.1. Control register (CAU_CTL) 328
13.8.2. Status register 0 (CAU_STAT0) 329

13.8.3. Data input register (CAU_DI) 330
13.8.4. Data output register (CAU_DO) 330
13.8.5. DMA enable register (CAU_DMAEN) 331
13.8.6. Interrupt enable register (CAU_INTEN) 331
13.8.7. Status register 1 (CAU_STAT1) 332
13.8.8. Interrupt flag register (CAU_INTF) 332
13.8.9. Key registers (CAU_KEY0..3(H / L)) 333
13.8.10. Error flag register (CAU_ERROR_FLAG) 336
14. Hash Acceleration Unit (HAU) 337
14.1. Overview 337
14.2. Characteristics 337
14.3. HAU data type 337
14.4. HAU core 339
14.4.1. Automatic data padding 339
14.4.2. Digest computing 340
14.4.3. Hash mode 341
14.5. HAU interrupt 341
14.5.1. Input FIFO interrupt 341
14.5.2. Calculation completion interrupt 341
14.6. Register definition 342
14.6.1. Control register (HAU_CTL) 342
14.6.2. Data input register (HAU_DI) 343
14.6.3. Configuration register (HAU_CFG) 343
14.6.4. Data output register (HAU_DO0..7) 344
14.6.5. Interrupt enable register (HAU_INTEN) 347
14.6.6. Status and flag register (HAU_STAT) 347
15. Direct memory access controller (DMA) 349
15.1. Overview 349
15.2. Characteristics 349
15.3. Block diagram 350
15.4. Function overview 350
15.4.1. DMA operation 350
15.4.2. Link list descriptor 353
15.4.3. Data Transfer mode 353
15.4.4. Repeat area access 354
15.4.5. Peripheral handshake 355
15.4.6. Arbitration 355
15.4.7. Address generation 356
15.4.8. Circular mode 356

15.4.9. Memory to memory mode 356
15.4.10. Channel configuration 356
15.4.11. Interrupt 357
15.4.12. DMA request mapping 358
15.5. Register definition 359
15.5.1. Interrupt flag register 0 (DMA_INTF0) 359
15.5.2. Interrupt flag register 1 (DMA_INTF1) 360
15.5.3. Interrupt flag clear register 0 (DMA_INTC0) 361
15.5.4. Interrupt flag clear register 1 (DMA_INTC1) 361
15.5.5. Channel x control register 0 (DMA_CHxCTL0) 362
15.5.6. Channel x counter register (DMA_CHxCNT) 364
15.5.7. Channel x peripheral base address register (DMA_CHxPADDR) 365
15.5.8. Channel x memory base address register (DMA_CHxMADDR) 365
15.5.9. Channel 0 peripheral address offset register (DMA_CH0PAOFS) 366
15.5.10. Channel 0 memory address offset register (DMA_CH0MAOFS) 366
15.5.11. Channel x link-list address register (DMA_CHxLADDR) 367
15.5.12. Channel x block cnounter register (DMA_CHxBCNT) 367
15.5.13. Channel x peripheral block offset register (DMA_CHxPABOFS) 367
15.5.14. Channel x memory block offset register (DMA_CHxMABOFS) 368
15.5.15. Channel x control1 register (DMA_CHxCTL1) 368
16. DMA request multiplexer (DMAMUX) 371
16.1. Overview 371
16.2. Characteristics 371
16.3. Block diagram 372
16.4. Function overview 372
16.4.1. DMAMUX signals 373
16.4.2. DMAMUX request multiplexer 373
16.4.3. DMAMUX request generator 376
16.4.4. Channel configurations 376
16.4.5. Interrupt 377
16.4.6. DMAMUX mapping 377
16.5. Register definition 383
16.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG) 383
16.5.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF) 384
16.5.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC) 384
16.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG) 385
16.5.5. Request generator interrupt flag register (DMAMUX_RG_INTF) 386
16.5.6. Request generator interrupt flag clear register (DMAMUX_RG_INTC) 386
17. Debug (DBG) 388
17.1. Introduction 388

17.2. JTAG/SW function description 388
17.2.1. Switch JTAG or SW interface 388
17.2.2. Pin assignment 388
17.2.3. JTAG daisy chained structure 389
17.2.4. Secure JTAG 389
17.2.5. Debug reset 391
17.2.6. JEDEC-106 ID code 391
17.3. Debug hold function description 391
17.3.1. Debug support for power saving modes 391
17.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT, CAN and RTC 391
17.4. DBG registers 393
17.4.1. ID code register (DBG_ID) 393
17.4.2. Control register 0 (DBG_CTL0) 393
17.4.3. Control register 1 (DBG_CTL1) 396
17.4.4. Control register 2 (DBG_CTL2) 398
18. Analog-to-digital converter (ADC) 399
18.1. Overview 399
18.2. Characteristics 399
18.3. Pins and internal signals 400
18.4. Function overview 401
18.4.1. ADC clock 401
18.4.2. ADC enable 401
18.4.3. Routine and inserted sequence 401
18.4.4. Operation modes 402
18.4.5. Inserted sequence management 406
18.4.6. Convert data latch 407
18.4.7. Conversion result threshold monitor 408
18.4.8. Data storage mode 409
18.4.9. Sample time configuration 410
18.4.10. External trigger configuration 410
18.4.11. DMA request 412
18.4.12. Overflow detection 412
18.4.13. ADC internal channels 412
18.4.14. Battery voltage monitoring 413
18.4.15. Programmable resolution (DRES) 413
18.4.16. On-chip hardware oversampling 414
18.5. ADC sync mode 416
18.5.1. Free mode 417
18.5.2. Routine parallel mode 417
18.5.3. Inserted parallel mode 418

18.5.4. Routine follow-up fast mode 419
18.5.5. Routine follow-up slow mode 419
18.5.6. Inserted trigger rotation mode 420
18.5.7. Combined routine parallel & inserted parallel mode 421
18.5.8. Combined routine parallel & inserted trigger rotation mode 421
18.5.9. Combined inserted parallel & routine follow-up mode 422
18.6. ADC interrupts 423
18.7. Register definition 424
18.7.1. Status register (ADC_STAT) 424
18.7.2. Control register 0 (ADC_CTL0) 426
18.7.3. Control register 1 (ADC_CTL1) 429
18.7.4. Sample time register 0 (ADC_SAMPT0) 433
18.7.5. Sample time register 1 (ADC_SAMPT1) 433
18.7.6. Inserted channel data offset register x (ADC_IOFFx) (x = 0..3) 434
18.7.7. Watchdog 0 high threshold register (ADC_WD0HT) 435
18.7.8. Watchdog 0 low threshold register (ADC_WD0LT) 435
18.7.9. Routine sequence register 0 (ADC_RSQ0) 436
18.7.10. Routine sequence register 1 (ADC_RSQ1) 436
18.7.11. Routine sequence register 2 (ADC_RSQ2) 437
18.7.12. Inserted sequence register (ADC_ISQ) 437
18.7.13. Latch data register x (ADC_LDATAx) (x= 0..3) 438
18.7.14. Routine data register (ADC_RDATA) 439
18.7.15. Inserted data register (ADC_IDATA) 439
18.7.16. Latch data control register(ADC_LDCTL) 439
18.7.17. Watchdog 1 high threshold register (ADC_WD1HT) 441
18.7.18. Watchdog 1 low threshold register (ADC_WD1LT) 441
18.7.19. Watchdog 2 high threshold register (ADC_WD2HT) 441
18.7.20. Watchdog 2 low threshold register (ADC_WD2LT) 442
18.7.21. Watchdog 1 Channel Selection Register (ADC_WD1SR) 442
18.7.22. Watchdog 2 Channel Selection Register (ADC_WD2SR) 443
18.7.23. Oversample control register (ADC_OVSAMPCTL) 444
18.7.24. Sample time register 2 (ADC_SAMPT2) 445
19. Digital-to-analog converter (DAC) 447
19.1. Overview 447
19.2. Characteristics 447
19.3. Function overview 449
19.3.1. DAC enable 449
19.3.2. DAC output buffer 449
19.3.3. DAC data configuration 449
19.3.4. DAC trigger 449
19.3.5. DAC conversion 449

19.3.6. DAC noise wave 450
19.3.7. DAC output voltage 451
19.3.8. DMA request 451
19.3.9. DAC concurrent conversion 451
19.3.10. DAC output buffer calibration 451
19.4. Register definition 454
19.4.1. DACx control register 0 (DAC_CTL0) 454
19.4.2. DACx software trigger register (DAC_SWT) 457
19.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH) 457
19.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH) 458
19.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH) 458
19.4.6. DACx_OUT1 12-bit right-aligned data holding register (DAC_OUT1_R12DH) 459
19.4.7. DACx_OUT1 12-bit left-aligned data holding register (DAC_OUT1_L12DH) 459
19.4.8. DACx_OUT1 8-bit right-aligned data holding register (DAC_OUT1_R8DH) 459
19.4.9. DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH) 460
19.4.10. DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH) 460
19.4.11. DACx concurrent mode 8-bit right-aligned data holding register (DACC_R8DH) 461
19.4.12. DACx_OUT0 data output register (DAC_OUT0_DO) 461
19.4.13. DACx_OUT1 data output register (DAC_OUT1_DO) 462
19.4.14. DACx_OUT0 status register 0 (DAC_STAT0) 462
19.4.15. DACx calibration register (DAC_CALR) 463
20. Comparator (CMP) 466
20.1. Overview 466
20.2. Characteristic 466
20.3. Function overview 466
20.3.1. CMP clock and reset 467
20.3.2. CMP I/O configuration 467
20.3.3. CMP hysteresis 468
20.3.4. CMP register write protection 469
20.3.5. CMP output blanking 469
20.3.6. CMP digital filter 470
20.3.7. CMP voltage scaler function 471
20.3.8. CMP interrupt 471
20.4. Register definition 472
20.4.1. Status register (CMP_STAT) 472
20.4.2. Interrupt flag clear register (CMP_IFC) 473
20.4.3. CMP0 control/status register (CMP0_CS) 473
20.4.4. CMP1 control/status register (CMP1_CS) 475
21. Watchdog timer (WDGT) 478
21.1. Free watchdog timer (FWDGT) 478

21.1.1. Overview 478
21.1.2. Characteristics 478
21.1.3. Function overview 478
21.1.4. Register definition 481
21.2. Window watchdog timer (WWDGT) 485
21.2.1. Overview 485
21.2.2. Characteristics 485
21.2.3. Function overview 485
21.2.4. Register definition 488
22. Real time clock (RTC) 490
22.1. Overview 490
22.2. Characteristics 490
22.3. Function overview 491
22.3.1. Block diagram 491
22.3.2. Clock source and prescalers 491
22.3.3. Shadow registers introduction 492
22.3.4. Configurable and field maskable alarm 492
22.3.5. RTC initialization and configuration 493
22.3.6. Configurable periodic auto-wakeup counter 494
22.3.7. Calendar reading 495
22.3.8. Resetting the RTC 496
22.3.9. RTC shift function 496
22.3.10. RTC reference clock detection 497
22.3.11. RTC smooth digital calibration 498
22.3.12. RTC high-resolution calibration 500
22.3.13. Time-stamp function 502
22.3.14. Tamper detection 502
22.3.15. Calibration clock output 504
22.3.16. Alarm output 504
22.3.17. RTC pin configuration 504
22.3.18. RTC power saving mode management 505
22.3.19. RTC interrupts 505
22.4. Register definition 507
22.4.1. Time register (RTC_TIME) 507
22.4.2. Date register (RTC_DATE) 507
22.4.3. Control register (RTC_CTL) 508
22.4.4. Status register (RTC_STAT) 511
22.4.5. Prescaler register (RTC_PSC) 514
22.4.6. Wakeup timer register (RTC_WUT) 514
22.4.7. Alarm 0 time and date register (RTC_ALRM0TD) 515
22.4.8. Alarm 1 time and date register (RTC_ALRM1TD) 516

22.4.9. Write protection key register (RTC_WPK) 517
22.4.10. Sub second register (RTC_SS) 517
22.4.11. Shift function control register (RTC_SHIFTCTL) 518
22.4.12. Time of time-stamp 0 rising register (RTC_TTS0R) 518
22.4.13. Date of time-stamp 0 rising register (RTC_DTS0R) 519
22.4.14. Sub second of time-stamp 0 rising register (RTC_SSTS0R) 520
22.4.15. High resolution frequency compensation register (RTC_HRFC) 520
22.4.16. Tamper register (RTC_TAMP) 521
22.4.17. Alarm 0 sub second register (RTC_ALRM0SS) 523
22.4.18. Alarm 1 sub second register (RTC_ALRM1SS) 524
22.4.19. Configuration register (RTC_CFG) 525
22.4.20. Backup registers (RTC_BKPx) (x=0..3) 526
22.4.21. Time of time-stamp 0 falling register (RTC_TTS0F) 526
22.4.22. Date of time-stamp 0 falling register (RTC_DTS0F) 527
22.4.23. Sub second of time-stamp 0 falling register (RTC_SSTS0F) 527
22.4.24. Time of time-stamp 1 rising register (RTC_TTS1R) 528
22.4.25. Date of time-stamp 1 rising register (RTC_DTS1R) 529
22.4.26. Sub second of time-stamp 1 rising register (RTC_SSTS1R) 529
22.4.27. Time of time-stamp 1 falling register (RTC_TTS1F) 530
22.4.28. Date of time-stamp 1 falling register (RTC_DTS1F) 531
22.4.29. Sub second of time-stamp 1 falling register (RTC_SSTS1F) 531
22.4.30. High resolution clock frequence calibration register(RTC_HRCFC) 532
22.4.31. High-resolution calibration register(RTC_HRCR) 532

23. TIMER (TIMERx) 534

23.1. Advanced timer (TIMERx, x=0, 7, 19) 536
23.1.1. Overview 536
23.1.2. Characteristics 536
23.1.3. Block diagram 537
23.1.4. Function overview 537
23.1.5. Registers definition (TIMERx, x=0, 7, 19) 580

23.2. General level0 timer (TIMERx, x=1,2,3,4) 626
23.2.1. Overview 626
23.2.2. Characteristics 626
23.2.3. Block diagram 626
23.2.4. Function overview 627
23.2.5. Registers definition (TIMERx, x=1,2,3,4) 647

23.3. General level2 timer (TIMERx, x=9, 10, 12, 13) 672
23.3.1. Overview 672
23.3.2. Characteristics 672
23.3.3. Block diagram 672
23.3.4. Function overview 673
23.3.5. TIMERx registers(x=9, 10, 12, 13) 683

23.4. General level3 timer (TIMERx, x=14, 40, 41, 42, 43, 44) 695
23.4.1. Overview 695
23.4.2. Characteristics 695
23.4.3. Block diagram 695
23.4.4. Function overview 696
23.4.5. Register definition (TIMERx, x=14,40,41,42,43,44) 725
23.5. Basic timer (TIMERx, x=5,6,50,51) 760
23.5.1. Overview 760
23.5.2. Characteristics 760
23.5.3. Block diagram 760
23.5.4. Function overview 760
23.5.5. Registers definition (TIMERx, x=5,6,50,51) 764
24. Universal synchronous / asynchronous receiver / transmitter (USART) 769
24.1. Universal synchronous / asynchronous receiver / transmitter (USARTx/UARTx, x=0..4) 769
24.1.1. Overview 769
24.1.2. Characteristics 769
24.1.3. Function overview 770
24.1.4. Register definition 787
24.2. Universal synchronous / asynchronous receiver / transmitter (USARTx, x=5) 803
24.2.1. Overview 803
24.2.2. Characteristics 803
24.2.3. Function overview 805
24.2.4. Register defintion 823
25. Inter-integrated circuit interface (I2C) 846
25.1. Overview 846
25.2. Characteristics 846
25.3. Function overview 846
25.3.1. Clock requirements 847
25.3.2. I2C communication flow 848
25.3.3. Noise filter 851
25.3.4. I2C timings configuration 851
25.3.5. I2C reset 853
25.3.6. Data transfer 853
25.3.7. I2C slave mode 855
25.3.8. I2C master mode 860
25.3.9. SMBus support 865
25.3.10. SMBus mode 868
25.3.11. Wakeup from Deep-sleep mode 869
25.3.12. Use DMA for data transfer 870

25.3.13. I2C error and interrupts 870
25.3.14. I2C debug mode 871
25.4. Register definition 872
25.4.1. Control register 0 (I2C_CTL0) 872
25.4.2. Control register 1 (I2C_CTL1) 874
25.4.3. Slave address register 0 (I2C_SADDR0) 876
25.4.4. Slave address register 1 (I2C_SADDR1) 877
25.4.5. Timing register (I2C_TIMING) 878
25.4.6. Timeout register (I2C_TIMEOUT) 879
25.4.7. Status register (I2C_STAT) 880
25.4.8. Status clear register (I2C_STATC) 883
25.4.9. PEC register (I2C_PEC) 884
25.4.10. Receive data register (I2C_RDATA) 884
25.4.11. Transmit data register (I2C_TDATA) 885
25.4.12. Control register 2 (I2C_CTL2) 885
26. Serial peripheral interface/Inter-IC sound (SPI/I2S) 887
26.1. Overview 887
26.2. Characteristics 887
26.2.1. SPI characteristics 887
26.2.2. I2S characteristics 887
26.3. SPI function overview 888
26.3.1. SPI block diagram 888
26.3.2. SPI signal description 888
26.3.3. SPI clock timing and data format 889
26.3.4. NSS function 890
26.3.5. SPI operation modes 892
26.3.6. DMA function 900
26.3.7. CRC function 900
26.3.8. SPI interrupts 901
26.4. I2S function overview 903
26.4.1. I2S block diagram 903
26.4.2. I2S signal description 903
26.4.3. I2S audio standards 904
26.4.4. I2S clock 912
26.4.5. Operation 913
26.4.6. DMA function 917
26.4.7. I2S interrupts 917
26.5. Register definition 920
26.5.1. Control register 0 (SPI_CTL0) 920
26.5.2. Control register 1 (SPI_CTL1) 922
26.5.3. Status register (SPI_STAT) 923

26.5.4. Data register (SPI_DATA) 924
26.5.5. CRC polynomial register (SPI_CRCPOLY) 925
26.5.6. RX CRC register (SPI_RCRC) 926
26.5.7. TX CRC register (SPI_TCRC) 926
26.5.8. I2S control register (SPI_I2SCTL) 927
26.5.9. I2S clock prescaler register (SPI_I2SPSC) 928
26.5.10. Quad-SPI mode control register (SPI_QCTL) 929

27. Octal-SPI interface(OSPI) 931

27.1. Overview 931

27.2. Characteristics 931

27.3. Function overview 931

27.3.1. OSPI block diagram 931
27.3.2. OSPI regular command format 935
27.3.3. HyperBusTM mode 939
27.3.4. Wrap support 942
27.3.5. Chip select boundary 942
27.3.6. Refresh support 943

27.4. Operating modes 943

27.4.1. Indirect mode 943
27.4.2. Status polling mode 944
27.4.3. Memory-mapped mode 945

27.5. OSPI configuration 945

27.5.1. OSPI system configuration 945
27.5.2. OSPI device configuration 946
27.5.3. OSPI regular command configuration 946
27.5.4. OSPI HyperBusTM mode configuration 947

27.6. Data sampling shift 948

27.7. OSPI send instruction only once 948

27.8. Busy 948

27.9. Error management 949

27.10. OSPI interrupt 949

27.11. Register definition 950

27.11.1. Control register (OSPI_CTL) 950
27.11.2. Device configuration register 0 (OSPI_DCFG0) 953
27.11.3. Device configuration register 1 (OSPI_DCFG1) 955
27.11.4. Device configuration register 2 (OSPI_DCFG2) 956
27.11.5. Device configuration register 3 (OSPI_DCFG3) 956
27.11.6. Status register (OSPI_STAT) 957
27.11.7. Status clear register (OSPI_STATC) 958

27.11.8. Data length register (OSPI_DTLEN) 958
27.11.9. Address register(OSPI_ADDR) 959
27.11.10. Data register (OSPI_DATA) 960
27.11.11. Status mask register (OSPI_STATMK) 960
27.11.12. Status match register(OSPI_STATMATCH) 961
27.11.13. Interval register (OSPI_INTERVAL) 961
27.11.14. Transfer configuration0 register (OSPI_TCFG0) 962
27.11.15. Timing configuration0 register (OSPI_TIMCFG0) 964
27.11.16. Instruction0 register (OSPI_INS0) 965
27.11.17. Alternate bytes0 register (OSPI_ALTE0) 965
27.11.18. Low power timeout register (OSPI_LPOUT) 966
27.11.19. Wrap transfer configuration0 register (OSPI_WPTCFG0) 966
27.11.20. Wrap timing configuration0 register (OSPI_WPTIMCFG0) 968
27.11.21. Wrap instruction0 register (OSPI_WPINS0) 969
27.11.22. Wrap alternate byte0 register (OSPI_WPALTE0) 969
27.11.23. Write transfer configuration0 register (OSPI_WTCFG0) 970
27.11.24. Write timing configuration0 register (OSPI_WTIMCFG0) 972
27.11.25. Write instruction0 register (OSPI_WINS0) 972
27.11.26. Write alternate byte0 register (OSPI_WALTE0) 973
27.11.27. HyperBusTM latency configuration register (OSPI_HBLCFG) 973
27.11.28. Transfer configuration1 register (OSPI_TCFG1) 974
27.11.29. Timing configuration1 register (OSPI_TIMCFG1) 976
27.11.30. Instruction1 register (OSPI_INS1) 976
27.11.31. Alternate1 bytes register (OSPI_ALTE1) 977
27.11.32. Wrap transfer configuration1 register (OSPI_WPTCFG1) 977
27.11.33. Wrap timing configuration1 register (OSPI_WPTIMCFG1) 979
27.11.34. Wrap instruction1 register (OSPI_WPINS1) 980
27.11.35. Wrap alternate1 byte register (OSPI_WPALTE1) 980
27.11.36. Write transfer configuration1 register (OSPI_WTCFG1) 981
27.11.37. Write timing configuration1 register (OSPI_WTIMCFG1) 983
27.11.38. Write instruction1 register (OSPI_WINS1) 983
27.11.39. Write alternate1 byte register (OSPI_WALTE1) 984

28. External memory controller (EXMC) 985

28.1. Overview 985

28.2. Characteristics 985

28.3. Function overview 985

28.3.1. Block diagram 985
28.3.2. Basic regulation of EXMC access 986
28.3.3. External device address mapping 987
28.3.4. NOR/PSRAM controller 990
28.3.5. NAND Flash controller 999

28.4. Registers definition 1004

28.4.1. NOR/PSRAM controller register 1004
28.4.2. NAND Flash controller register 1007
29. Controller area network (CAN) 1012
29.1. Overview 1012
29.2. Characteristics 1012
29.3. Function overview 1013
29.3.1. Working mode 1013
29.3.2. Communication modes 1014
29.3.3. Data transmission 1015
29.3.4. Data reception 1017
29.3.5. Filtering function 1019
29.3.6. Time-triggered communication 1022
29.3.7. Communication parameters 1023
29.3.8. CAN FD operation 1024
29.3.9. Transmitter Delay Compensation 1025
29.3.10. Error flags 1026
29.3.11. CAN interrupts 1027
29.4. CAN registers 1029
29.4.1. Control register (CAN_CTL) 1029
29.4.2. Status register (CAN_STAT) 1030
29.4.3. Transmit status register (CAN_TSTAT) 1032
29.4.4. Receive message FIFO0 register (CAN_RFIFO0) 1035
29.4.5. Receive message FIFO1 register (CAN_RFIFO1) 1036
29.4.6. Interrupt enable register (CAN_INTEN) 1037
29.4.7. Error register (CAN_ERR) 1038
29.4.8. Bit timing register (CAN_BT) 1039
29.4.9. FD control register (CAN_FDCTL) 1040
29.4.10. FD status register (CAN_FDSTAT) 1041
29.4.11. FD transmitter delay compensation register (CAN_FDTDC) 1042
29.4.12. Date Bit timing register (CAN_DBT) 1042
29.4.13. Transmit mailbox identifier register (CAN_TMIx) (x = 0…2) 1043
29.4.14. Transmit mailbox property register (CAN_TMPx) (x = 0…2) 1044
29.4.15. Transmit mailbox data0 register (CAN_TMDATA0x) (x = 0…2) 1045
29.4.16. Transmit mailbox data1 register (CAN_TMDATA1x) (x = 0...2) 1045
29.4.17. Receive FIFO mailbox identifier register (CAN_RFIFOMIx) (x = 0,1) 1046
29.4.18. Receive FIFO mailbox property register (CAN_RFIFOMPx) (x = 0,1) 1046
29.4.19. Receive FIFO mailbox data0 register (CAN_RFIFOMDATA0x) (x = 0,1) 1047
29.4.20. Receive FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x = 0,1) 1048
29.4.21. Filter control register (CAN_FCTL) 1048
29.4.22. Filter mode configuration register (CAN_FMCFG) 1049
29.4.23. Filter scale configuration register (CAN_FSCFG) 1050
29.4.24. Filter associated FIFO register (CAN_FAFIFO) 1050

29.4.25. Filter working register (CAN_FW) 1051
29.4.26. Filter x data y register (CAN_FxDATAy) (x = 0…27, y = 0,1) 1052
30. Secure digital input/output interface (SDIO) 1053
30.1. Introduction 1053
30.2. Main features 1053
30.3. SDIO bus topology 1053
30.4. SDIO functional description 1056
30.4.1. SDIO adapter 1056
30.4.2. APB2 interface 1060
30.5. Card functional description 1061
30.5.1. Card registers 1061
30.5.2. Commands 1063
30.5.3. Responses 1073
30.5.4. Data packets format 1077
30.5.5. Two status fields of the card 1078
30.6. Programming sequence 1085
30.6.1. Card identification 1085
30.6.2. No data commands 1087
30.6.3. Single block or multiple block write 1087
30.6.4. Single block or multiple block read 1088
30.6.5. Stream write and stream read (MMC only) 1089
30.6.6. Erase 1091
30.6.7. Bus width selection 1092
30.6.8. Protection management 1092
30.6.9. Card Lock/Unlock operation 1093
30.7. Specific operations 1095
30.7.1. SD I/O specific operations 1095
30.7.2. CE-ATA specific operations 1099
30.8. SDIO registers 1101
30.8.1. Power control register (SDIO_PWRCTL) 1101
30.8.2. Clock control register (SDIO_CLKCTL) 1101
30.8.3. Command argument register (SDIO_CMDAGMT) 1103
30.8.4. Command control register (SDIO_CMDCTL) 1103
30.8.5. Command index response register (SDIO_RSPCMDIDX) 1105
30.8.6. Response register (SDIO_RESPx x=0..3) 1105
30.8.7. Data timeout register (SDIO_DATATO) 1106
30.8.8. Data length register (SDIO_DATALEN) 1106
30.8.9. Data control register (SDIO_DATACTL) 1107
30.8.10. Data counter register (SDIO_DATACNT) 1108
30.8.11. Status register (SDIO_STAT) 1109

30.8.12. Interrupt clear register (SDIO_INTC) 1110
30.8.13. Interrupt enable register (SDIO_INTEN) 1111
30.8.14. FIFO counter register (SDIO_FIFOCNT) 1114
30.8.15. FIFO data register (SDIO_FIFO) 1114
31. Universal serial bus full-speed interface (USBFS) 1116
31.1. Overview 1116
31.2. Characteristics 1116
31.3. Block diagram 1117
31.4. Signal description 1117
31.5. Function overview 1117
31.5.1. USBFS clocks and working modes 1117
31.5.2. USB host function 1118
31.5.3. USB device function 1121
31.5.4. Data FIFO 1122
31.5.5. Operation guide 1124
31.6. Interrupts 1128
31.7. Register definition 1130
31.7.1. Global control and status registers 1130
31.7.2. Host control and status registers 1148
31.7.3. Device control and status registers 1159
31.7.4. Power and clock control register (USBFS_PWRCLKCTL) 1181
32. Appendix 1183
32.1. List of abbreviations used in register 1183
32.2. List of terms 1184
32.3. Available peripherals 1184
33. Revision history 1185

List of Figures

Figure 1-1. The structure of the Cortex®-M33 processor 39
Figure 1-2. GD32E53x series system architecture 41
Figure 1-3. ECC decoder 47
Figure 2-1. Process of page erase operation 87
Figure 2-2. Process of mass erase operation 88
Figure 2-3. Process of four-word (128-bit) program operation 90
Figure 3-1. ICACHE block diagram 119
Figure 3-2. ICACHE remapping address 121
Figure 4-1. DCACHE block diagram 130
Figure 4-2. DCACHE block diagram 131
Figure 4-3. cache line 132
Figure 4-4. cache way 133
Figure 5-1. Power supply overview 147
Figure 5-2. V_{CORE} domain supply overview 148
Figure 5-3. Waveform of the POR/PDR 150
Figure 5-4. Waveform of the LVD threshold 151
Figure 5-5. Waveform of the OVD threshold 152
Figure 5-6. Waveform of the BOR 152
Figure 5-7. Waveform of the VAUVD threshold 153
Figure 5-8. Waveform of the VAOVD threshold 154
Figure 5-9. waveform of VOVD 155
Figure 5-10. waveform of VUVD 155
Figure 6-1. The system reset circuit 171
Figure 6-2. GD32E533 Clock tree 172
Figure 6-3. GD32E535 and GD32E53P Clock tree 173
Figure 6-4. HXTAL clock source 175
Figure 6-5. HXTAL clock source in bypass mode 175
Figure 7-1. CTC overview 223
Figure 7-2. CTC trim counter 224
Figure 8-1. Block diagram of EXTI 238
Figure 9-1. TRIGSEL main composition example 245
Figure 10-1. Basic structure of a general-pupose I/O 283
Figure 10-2. Basic structure of Input configuration 284
Figure 10-3. Basic structure of Output configuration 285
Figure 10-4. Basic structure of Analog configuration 285
Figure 10-5. Basic structure of Alternate function configuration 286
Figure 10-6. Filtering using the sampling window 288
Figure 10-7. Input filtering clock cycle 289

Figure 11-1. Block diagram of CRC calculation unit 311
Figure 12-1. TRNG block diagram 316
Figure 13-1. DATAM No swapping and Half-word swapping 322
Figure 13-2. DATAM Byte swapping and Bit swapping 322
Figure 13-3. CAU diagram 323
Figure 13-4. AES ECB encryption 324
Figure 13-5. AES ECB decryption 325
Figure 14-1. DATAM No swapping and Half-word swapping 338
Figure 14-2. DATAM Byte swapping and Bit swapping 338
Figure 14-3. HAU block diagram 339
Figure 15-1. Block diagram of DMA 350
Figure 15-2. an example of repeat address access in block transfer mode 354
Figure 15-3. Handshake mechanism 355
Figure 15-4. DMA interrupt logic 358
Figure 16-1. Block diagram of DMAMUX 372
Figure 16-2. Synchronization mode 374
Figure 16-3. Event generation 375
Figure 18-1. ADC module block diagram 401
Figure 18-2. Single operation mode 402
Figure 18-3. Continuous operation mode 403
Figure 18-4. Scan operation mode, continuous disable 404
Figure 18-5. Scan operation mode, continuous enable 405
Figure 18-6. Discontinuous operation mode 405
Figure 18-7. Auto-insertion, CNT = 1 406
Figure 18-8. Triggered insertion 407
Figure 18-9. Data latch 407
Figure 18-10. Sequence data from ADC_LDATAx 408
Figure 18-11. Data storage mode of 12-bit resolution 409
Figure 18-12. Data storage mode of 6-bit resolution 410
Figure 18-13. Routine trigger occurs during routine conversion (ITGRECEN = 0, RTGRECEN = 0) 411
Figure 18-14. Routine trigger occurs during routine conversion (RTGRECEN = 1) 411
Figure 18-15. 20-bit to 16-bit result truncation 415
Figure 18-16. Numerical example with 5-bits shift and rounding 415
Figure 18-17. ADC sync block diagram 417
Figure 18-18. Routine parallel mode on 10 channels 418
Figure 18-19. Inserted parallel mode on 4 channels 418
Figure 18-20. Routine follow-up fast mode (the CTN bit of ADCs are set) 419
Figure 18-21. Routine follow-up slow mode 420
Figure 18-22. Inserted trigger rotation: DISIC=0, IL=1 420
Figure 18-23. Inserted trigger rotation: DISIC=1, IL=1 421
Figure 18-24. Routine parallel & inserted trigger rotation mode: SYNCM = 4'b0010 422
Figure 18-25. Trigger occurs during inserted conversion: SYNCM = 4'b0010 422
Figure 18-26. Follow-up single channel with inserted sequence CH1, CH2 423

Figure 19-1. DAC block diagram 448
Figure 19-2. DAC LFSR algorithm 450
Figure 19-3. DAC triangle noise wave 451
Figure 20-1. CMP block diagram 467
Figure 20-2. CMP hysteresis 469
Figure 20-3. The CMP outputs signal blanking 470
Figure 20-4. CMP noise filter and interrupt operation 470
Figure 21-1. Free watchdog block diagram 479
Figure 21-2. Window watchdog timer block diagram 485
Figure 21-3. Window watchdog timing diagram 486
Figure 22-1. Block diagram of RTC 491
Figure 23-1. Advanced timer block diagram 537
Figure 23-2. Normal mode, internal clock divided by 1 538
Figure 23-3. Counter timing diagram with prescaler division change from 1 to 2 539
Figure 23-4. Timing diagram of up counting mode, PSC=0/2 540
Figure 23-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing 540
Figure 23-6. Timing diagram of down counting mode, PSC=0/2 541
Figure 23-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing 542
Figure 23-8. Timing diagram of center-aligned counting mode 543
Figure 23-9. Repetition counter timing diagram of center-aligned counting mode 544
Figure 23-10. Repetition counter timing diagram of up counting mode 544
Figure 23-11. Repetition counter timing diagram of down counting mode 545
Figure 23-12. Input capture logic for channel 0 545
Figure 23-13. Output compare logic 547
Figure 23-14. Output-compare in three modes 548
Figure 23-15. Timing diagram of EAPWM 549
Figure 23-16. Timing diagram of CAPWM 549
Figure 23-17. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 551
Figure 23-18. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 552
Figure 23-19. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 552
Figure 23-20. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 553
Figure 23-21. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 553
Figure 23-22. Four Channels outputs in Composite PWM mode 554
Figure 23-23. Channel x output PWM with(CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL) 555
Figure 23-24. Channel x output PWM with(CHxVAL = CHxCOMVAL_ADD) 557
Figure 23-25. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 557
Figure 23-26. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00) 558
Figure 23-27. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00) 559
Figure 23-28. Complementary output with dead time insertion 564
Figure 23-29. Complementary output with different dead time(DTDIFEN=1) 565
Figure 23-30. BREAK0 function logic diagram 565
Figure 23-31. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 566

Figure 23-32. Channel break function logic diagram 567
Figure 23-33. Example of counter operation in quadrature decoder interface mode 568
Figure 23-34. Example of quadrature decoder interface mode with CI0FE0 polarity inverted 568
Figure 23-35. Hall sensor is used for BLDC motor 570
Figure 23-36. Hall sensor timing between two timers 570
Figure 23-37. Restart mode 571
Figure 23-38. Pause mode 572
Figure 23-39. Event mode 572
Figure 23-40. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60 573
Figure 23-41. Trigger mode of TIMER0 controlled by enable signal of TIMER2 574
Figure 23-42. Trigger mode of TIMER0 controlled by update signal of TIMER2 575
Figure 23-43. Pause mode of TIMER0 controlled by enable signal of TIMER2 575
Figure 23-44. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2 576
Figure 23-45. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2 577
Figure 23-46. Triggering four timers by trigger out of basic timer 578
Figure 23-47. Counting direction in CINITDIR is 0 or 1 578
Figure 23-48. General Level 0 timer block diagram 627
Figure 23-49. Normal mode, internal clock divided by 1 628
Figure 23-50. Counter timing diagram with prescaler division change from 1 to 2 629
Figure 23-51. Timing chart of up counting mode, PSC=0/2 630
Figure 23-52. Timing chart of up counting, change TIMERx_CAR ongoing 630
Figure 23-53. Timing chart of down counting mode, PSC=0/2 631
Figure 23-54. Timing chart of down counting mode, change TIMERx_CAR ongoing 632
Figure 23-55. Timing chart of center-aligned counting mode 633
Figure 23-56. Input capture logic 634
Figure 23-57. Output compare logic (x=0,1,2,3) 635
Figure 23-58. Output-compare under three modes 636
Figure 23-59. Timing chart of EAPWM 637
Figure 23-60. Timing chart of CAPWM 637
Figure 23-61. Example of counter operation in decoder interface mode 639
Figure 23-62. Example of decoder interface mode with CI0FE0 polarity inverted 639
Figure 23-63. Quadrature decoder signal disconnection detection block diagram 640
Figure 23-64. Example of counter operation in decoder mode 0 / 1 with CH1P=0 641
Figure 23-65. Example of counter operation in decoder mode 2 / 3 (CH0P / CH1P=0) 642
Figure 23-66. Restart mode 644
Figure 23-67. Pause mode 644
Figure 23-68. Event mode 645
Figure 23-69. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60 645
Figure 23-70. General level2 timer block diagram 673
Figure 23-71. Timing chart of internal clock divided by 1 674
Figure 23-72. Timing chart of PSC value change from 0 to 2 674
Figure 23-73. Timing chart of up counting mode, PSC=0/2 675
Figure 23-74. Timing chart of up counting mode, change TIMERx_CAR ongoing 676
Figure 23-75. Channel input capture principle 677

Figure 23-76. Channel output compare principle 678
Figure 23-77. Output-compare under three modes 679
Figure 23-78. Restart mode 681
Figure 23-79. Pause mode 681
Figure 23-80. Event mode 682
Figure 23-81. General level3 timer block diagram 696
Figure 23-82. Normal mode, internal clock divided by 1 697
Figure 23-83. Counter timing diagram with prescaler division change from 1 to 2 698
Figure 23-84. Timing diagram of up counting mode, PSC=0/2 699
Figure 23-85. Timing diagram of up counting mode, change TIMERx_CAR on the go 699
Figure 23-86. Timing diagram of down counting mode, PSC=0/2 700
Figure 23-87. Timing diagram of down counting mode, change TIMERx_CAR ongoing 701
Figure 23-88. Timing diagram of center-aligned counting mode 702
Figure 23-89. Repetition counter timing diagram of center-aligned counting mode 703
Figure 23-90. Repetition counter timing diagram of up counting mode 703
Figure 23-91. Repetition counter timing diagram of down counting mode 704
Figure 23-92. Input capture logic for channel 0 705
Figure 23-93. Input capture logic for multi mode channel 0 705
Figure 23-94. Output compare logic (when MCHxMSEL = 2'00, x=0) 707
Figure 23-95. Output compare logic (when MCHxMSEL = 2'11, x=0) 707
Figure 23-96. Output compare logic (x=1) 707
Figure 23-97. Output-compare in three modes 709
Figure 23-98. Timing diagram of EAPWM 710
Figure 23-99. Timing diagram of CAPWM 710
Figure 23-100. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD) 712
Figure 23-101. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD) 712
Figure 23-102. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD) 713
Figure 23-103. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL 713
Figure 23-104. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD 714
Figure 23-105. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00) 715
Figure 23-106. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00) 715
Figure 23-107. Complementary output with dead-time insertion 720
Figure 23-108. BREAK0 function logic diagram 721
Figure 23-109. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1) 721
Figure 23-110. Restart mode 722
Figure 23-111. Pause mode 723
Figure 23-112. Event mode 723
Figure 23-113. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 724
Figure 23-114. Basic timer block diagram 760
Figure 23-115. Normal mode, internal clock divided by 1 761
Figure 23-116. Counter timing diagram with prescaler division change from 1 to 2 761
Figure 23-117. Timing chart of up counting mode, PSC=0/2 762
Figure 23-118. Timing chart of up counting mode, change TIMERx_CAR ongoing 763

Figure 24-1. USART module block diagram 771
Figure 24-2. USART character frame (8 bits data and 1 stop bit) 771
Figure 24-3. USART transmit procedure 773
Figure 24-4. Receiving a frame bit by oversampling method 774
Figure 24-5. Configuration step when using DMA for USART transmission 776
Figure 24-6. Configuration steps when using DMA for USART reception 777
Figure 24-7. Hardware flow control between two USARTs 777
Figure 24-8. Hardware flow control 778
Figure 24-9. Break frame occurs during idle state 779
Figure 24-10. Break frame occurs during a frame 779
Figure 24-11. Example of USART in synchronous mode 780
Figure 24-12. 8-bit format USART synchronous waveform (CLEN=1) 780
Figure 24-13. IrDA SIR ENDEC module 781
Figure 24-14. IrDA data modulation 781
Figure 24-15. ISO7816-3 frame format 782
Figure 24-16. USART interrupt mapping diagram 785
Figure 24-17. Infrared modulation waveform schematic diagram 786
Figure 24-18. USART module block diagram 805
Figure 24-19. USART character frame (8 bits data and 1 stop bit) 806
Figure 24-20. USART transmit procedure 808
Figure 24-21. Oversampling method of a receive frame bit (OSB=0) 809
Figure 24-22. Configuration step when using DMA for USART transmission 810
Figure 24-23. Configuration step when using DMA for USART reception 811
Figure 24-24. Hardware flow control between two USARTs 811
Figure 24-25. Hardware flow control 812
Figure 24-26. Break frame occurs during idle state 814
Figure 24-27. Break frame occurs during a frame 814
Figure 24-28. Example of USART in synchronous mode 815
Figure 24-29. 8-bit format USART synchronous waveform (CLEN=1) 815
Figure 24-30. IrDA SIR ENDEC module 816
Figure 24-31. IrDA data modulation 816
Figure 24-32. ISO7816-3 frame format 817
Figure 24-33. USART receive FIFO structure 820
Figure 24-34. USART transmitter FIFO structure 820
Figure 24-35. USART interrupt mapping diagram 822
Figure 25-1. I2C module block diagram 847
Figure 25-2. Data validation 848
Figure 25-3. START and STOP signal 849
Figure 25-4. I2C communication flow with 10-bit address (Master Transmit) 849
Figure 25-5. I2C communication flow with 7-bit address (Master Transmit) 850
Figure 25-6. I2C communication flow with 7-bit address (Master Receive) 850
Figure 25-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0) 850
Figure 25-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)

..... 850
Figure 25-9. Data hold time 851
Figure 25-10. Data setup time 852
Figure 25-11. Data transmission 854
Figure 25-12. Data reception 854
Figure 25-13. I2C initialization in slave mode 857
Figure 25-14. Programming model for slave transmitting when SS=0 858
Figure 25-15. Programming model for slave transmitting when SS=1 859
Figure 25-16. Programming model for slave receiving 860
Figure 25-17. I2C initialization in master mode 861
Figure 25-18. Programming model for master transmitting (N<=255) 862
Figure 25-19. Programming model for master transmitting (N>255) 863
Figure 25-20. Programming model for master receiving (N<=255) 864
Figure 25-21. Programming model for master receiving (N>255) 865
Figure 25-22. SMBus Master Transmitter and Slave Receiver communication flow 869
Figure 25-23. SMBus Master Receiver and Slave Transmitter communication flow 869
Figure 26-1. Block diagram of SPI 888
Figure 26-2. SPI timing diagram in normal mode 890
Figure 26-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0) 890
Figure 26-4. A typical Full-duplex connection 893
Figure 26-5. A typical simplex connection (Master: Receive, Slave: Transmit) 893
Figure 26-6. A typical simplex connection (Master: Transmit only, Slave: Receive) 893
Figure 26-7. A typical bidirectional connection 894
Figure 26-8. Timing diagram of TI master mode with discontinuous transfer 896
Figure 26-9. Timing diagram of TI master mode with continuous transfer 896
Figure 26-10. Timing diagram of TI slave mode 896
Figure 26-11. Timing diagram of NSS pulse with continuous transmit 897
Figure 26-12. Timing diagram of quad write operation in Quad-SPI mode 898
Figure 26-13. Timing diagram of quad read operation in Quad-SPI mode 899
Figure 26-14. Block diagram of I2S 903
Figure 26-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 904
Figure 26-16. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 904
Figure 26-17. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 905
Figure 26-18. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 905
Figure 26-19. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 905
Figure 26-20. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 905
Figure 26-21. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 906
Figure 26-22. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 906
Figure 26-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 906
Figure 26-24. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 906
Figure 26-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 906
Figure 26-26. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 907
Figure 26-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 907
Figure 26-28. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 907

Figure 26-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 907
Figure 26-30. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 907
Figure 26-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 908
Figure 26-32. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 908
Figure 26-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 908
Figure 26-34. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 908
Figure 26-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 909
Figure 26-36. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 909
Figure 26-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 909
Figure 26-38. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 909
Figure 26-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 909
Figure26-40. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 909
Figure 26-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 910
Figure 26-42. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 910
Figure 26-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0) 910
Figure26-44. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1) 910
Figure 26-45. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0) 910
Figure 26-46. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1) 911
Figure 26-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0) 911
Figure 26-48. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1) 911
Figure 26-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0) 911
Figure 26-50. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1) 911
Figure 26-51. Block diagram of I2S clock generator 912
Figure 26-52. I2S initialization sequence 914
Figure 26-53. I2S master reception disabling sequence 916
Figure 27-1 OSPI octal communication mode block diagram 933
Figure 27-2 OSPI quad communication mode block diagram 934
Figure 27-3 OSPI dual CSN communication mode block diagram 934

Figure 27-4 OSPI command format in octal mode 935
Figure 27-5 OSPI command format in quad mode 935
Figure 27-6. DTR read timing using DQS in octal mode 938
Figure 27-7. CSN and SCK behavior 939
Figure 27-8. HyperBus™ read operation with initial latency 940
Figure 27-9. HyperBus™ read operation with additional latency 941
Figure 27-10. Read operation page crossing with latency 941
Figure 27-11. HyperBus™ write operation with initial latency 942
Figure 27-12. HyperBus™ write operation with additional latency 942
Figure 27-13. HyperBus™ write operation with no latency 942
Figure 28-1. The EXMC block diagram 986
Figure 28-2. EXMC memory banks 987
Figure 28-3. Four regions of bank0 address mapping 988
Figure 28-4. NAND address mapping 989
Figure 28-5. Diagram of bank1 common space 989
Figure 28-6. Multiplex mode read access 993
Figure 28-7. Multiplex mode write access 993
Figure 28-8. Read access timing diagram under async-wait signal assertion 995
Figure 28-9. Write access timing diagram under async-wait signal assertion 995
Figure 28-10. Read timing of synchronous multiplexed burst mode 997
Figure 28-11. Write timing of synchronous multiplexed burst mode 998
Figure 28-12. Access timing of common memory space of NAND flash Controller 1001
Figure 28-13. Access to none "NCE don't care" NAND Flash 1002
Figure 29-1. CAN module block diagram 1013
Figure 29-2. Transmission register 1016
Figure 29-3. State of transmit mailbox 1016
Figure 29-4. Reception register 1018
Figure 29-5. 32-bit filter 1019
Figure 29-6. 16-bit filter 1019
Figure 29-7. 32-bit mask mode filter 1020
Figure 29-8. 16-bit mask mode filter 1020
Figure 29-9. 32-bit list mode filter 1020
Figure 29-10. 16-bit list mode filter 1020
Figure 29-11. The bit time 1024
Figure 29-12. Transmitter Delay Measurement 1026
Figure 30-1. SDIO "no response" and "no data" operations 1054
Figure 30-2. SDIO multiple blocks read operation 1055
Figure 30-3. SDIO multiple blocks write operation 1055
Figure 30-4. SDIO sequential read operation 1055
Figure 30-5. SDIO sequential write operation 1055
Figure 30-6. SDIO block diagram 1056
Figure 30-7. Command token format 1063
Figure 30-8. Response Token Format 1074
Figure 30-9. 1-bit data bus width 1077

Figure 30-10. 4-bit data bus width 1078
Figure 30-11. 8-bit data bus width 1078
Figure 30-12. Read wait control by stopping SDIO_CLK 1096
Figure 30-13. Read wait operation using SDIO_DAT[2] 1096
Figure 30-14. Function2 read cycle inserted during function1 multiple read cycle 1097
Figure 30-15. Read Interrupt cycle timing 1098
Figure 30-16. Write interrupt cycle timing 1098
Figure 30-17. Multiple block 4-Bit read interrupt cycle timing 1098
Figure 30-18. Multiple block 4-Bit write interrupt cycle timing 1099
Figure 30-19. The operation for command completion disable signal 1100
Figure 31-1. USBFS block diagram 1117
Figure 31-2. Connection with host or device mode 1118
Figure 31-3. Host mode FIFO access register map 1123
Figure 31-4. Device mode FIFO space in SRAM 1123
Figure 31-5. Device mode FIFO access register map 1124

List of Tables

Table 1-1. The interconnection relationship of the AHB interconnect matrix 39
Table 1-2. Memory map of GD32E53x devices 42
Table 1-3. Boot modes 48
Table 2-1. GD32E53x base address and size for flash memory 82
Table 2-2. The relation between WSCNT and AHB clock frequency(for GD32E533) 83
Table 2-3. The relation between WSCNT and AHB clock frequency(For GD32E535 and GD32E53P) 83
Table 2-4. The relation between WSCNT and AHB clock frequency(For GD32E537) 84
Table 2-5. OTP bytes 90
Table 2-6. Option bytes 94
Table 2-7. WPx_SADDR and WPx_EADDR settings for protected pages 100
Table 2-8. SCR/WP Erase Protection 101
Table 3-1. TAG memory parameters 120
Table 3-2. ICACHE remap region size 121
Table 3-3. ICACHE cache ability for transaction 122
Table 3-4. Memory configuration 122
Table 3-5. ICACHE interrupt 124
Table 4-1. TAG memory parameters 133
Table 4-2. DCACHE cache ability for transaction 134
Table 4-3. Reading 135
Table 4-4. Write Back 136
Table 4-5. Write through 136
Table 5-1. Power saving mode summary 158
Table 6-1. Clock output 0 source select 178
Table 8-1. NVIC exception types in Cortex®-M33 234
Table 8-2. Interrupt vector table 234
Table 8-3. EXTI source 239
Table 9-1. Trigger input bit fields selection 245
Table 9-2. TRIGSEL input and output mapping 249
Table 10-1. GPIO configuration table 282
Table 10-2. OSC32 pins configuration 290
Table 10-3. OSC pins configuration 290
Table 15-1. DMA transfer operation 351
Table 15-2. interrupt events 357
Table 16-1. DMAMUX signals 373
Table 16-2. Interrupt events 377
Table 16-3. Request multiplexer input mapping 378
Table 16-4. Trigger input mapping 381
Table 16-5. Synchronization input mapping 382
Table 17-1. Pin assignment 389

Table 17-2. Debug mode OTP configuration 389
Table 18-1. ADC internal input channels 400
Table 18-2. ADC input pins definition 400
Table 18-3. External trigger modes and trigger type 410
Table 18-4. t_{CONV} timings depending on resolution 414
Table 18-5. Maximum output results for N and M (Grayed values indicates truncation) 415
Table 18-6. ADC sync mode table 416
Table 19-1. DAC I/O description 448
Table 19-2. DAC triggers and outputs summary 448
Table 19-3. Triggers of DAC 449
Table 20-1. CMP inputs and outputs summary 468
Table 21-1. Min/max FWDGT timeout period at 40 kHz (IRC40K) 480
Table 21-2. Min-max timeout value at 100 MHz (f_{PCLK1}) 487
Table 22-1. RTC pin configuration and function 504
Table 22-2. RTC power saving mode management 505
Table 22-3. RTC interrupts control 506
Table 23-1. Timers (TIMERx) are divided into four sorts 534
Table 23-2.The composite PWM pulse width 550
Table 23-3.The asymmetric PWM pulse width 555
Table 23-4. Complementary outputs controlled by parameters 561
Table 23-5. Output behavior of the channel in response to a BREAK0 (the break input is high active) 566
Table 23-6. Counting direction in different quadrature decoder signals 568
Table 23-7. Examples of slave mode 571
Table 23-8. Counting direction in different quadrature decoder signals 639
Table 23-9. the counter operation in decoder mode 1 641
Table 23-10. the counter operation in decoder mode 2 / 3 642
Table 23-11. Examples of slave mode 643
Table 23-12. Examples of slave mode 680
Table 23-13.The Composite PWM pulse width 711
Table 23-14. Complementary outputs controlled by parameters (MCHxMSEL =2'b11) 717
Table 23-15. Slave mode example table 722
Table 24-1. Description of USART important pins 770
Table 24-2. Configuration of stop bits 771
Table 24-3. USART interrupt requests 784
Table 24-4. Description of USART important pins 805
Table 24-5. Configuration of stop bits 806
Table 24-6. USART interrupt requests 821
Table 25-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors) 847
Table 25-2. Data setup time and data hold time 853
Table 25-3. Communication modes to be shut down 854
Table 25-4. I2C error flags 870
Table 25-5. I2C interrupt events 871

Table 26-1. SPI signal description 888
Table 26-2. Quad-SPI signal description 889
Table 26-3. NSS function in slave mode 891
Table 26-4. NSS function in master mode 891
Table 26-5. SPI operation modes 892
Table 26-6. SPI interrupt requests 902
Table 26-7. I2S bitrate calculation formulas 912
Table 26-8. Audio sampling frequency calculation formulas 912
Table 26-9. Direction of I2S interface signals for each operation mode 913
Table 26-10. I2S interrupt 918
Table 27-1. OSPI signal description 932
Table 27-2. Signal line description in HyperBus™ mode 939
Table 27-3 Instruction/address phase description in HyperBus™ mode 939
Table 27-4. OSPI interrupt requests 949
Table 28-1. NOR Flash interface signals description 990
Table 28-2. PSRAM non-muxed signal description 990
Table 28-3. EXMC bank 0 supports all transactions 991
Table 28-4. NOR / PSRAM controller timing parameters 992
Table 28-5. EXMC_timing models 992
Table 28-6. Multiplex mode related registers configuration 993
Table 28-7. Timing configurations of synchronous multiplexed read mode 997
Table 28-8. Timing configurations of synchronous multiplexed write mode 998
Table 28-9. 8-bit or 16-bit NAND interface signal 999
Table 28-10. Bank1 of EXMC support the memory and access mode 999
Table 28-11. NAND Flash programmable parameters 1000
Table 29-1. 32-bit filter number 1020
Table 29-2. Filtering index 1021
Table 29-3. CAN Event / Interrupt flags 1028
Table 30-1. SDIO I/O definitions 1057
Table 30-2. Command format 1063
Table 30-3. Card command classes (CCCs) 1064
Table 30-4. Basic commands (class 0) 1066
Table 30-5. Block-Oriented read commands (class 2) 1067
Table 30-6. Stream read commands (class 1) and stream write commands (class 3) 1068
Table 30-7. Block-Oriented write commands (class 4) 1068
Table 30-8. Erase commands (class 5) 1069
Table 30-9. Block oriented write protection commands (class 6) 1070
Table 30-10. Lock card (class 7) 1070
Table 30-11. Application-specific commands (class 8) 1071
Table 30-12. I/O mode commands (class 9) 1072
Table 30-13. Switch function commands (class 10) 1073
Table 30-14. Response R1 1074
Table 30-15. Response R2 1075
Table 30-16. Response R3 1075

Table 30-17. Response R4 for MMC 1075
Table 30-18. Response R4 for SD I/O 1076
Table 30-19. Response R5 for MMC 1076
Table 30-20. Response R5 for SD I/O 1076
Table 30-21. Response R6 1076
Table 30-22. Response R7 1077
Table 30-23. Card status 1079
Table 30-24. SD status 1081
Table 30-25. Performance move field 1083
Table 30-26. AU_SIZE field 1083
Table 30-27. Maximum AU size 1084
Table 30-28. Erase size field 1084
Table 30-29. Erase timeout field 1085
Table 30-30. Erase offset field 1085
Table 30-31. Lock card data structure 1093
Table 30-32. SDIO_RESPx register at different response type 1105
Table 31-1. USBFS signal description 1117
Table 31-2. USBFS global interrupt 1129
Table 32-1. List of abbreviations used in register 1183
Table 32-2. List of terms 1184
Table 33-1. Revision history 1185

1. System and memory architecture

The GD32E53x 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.

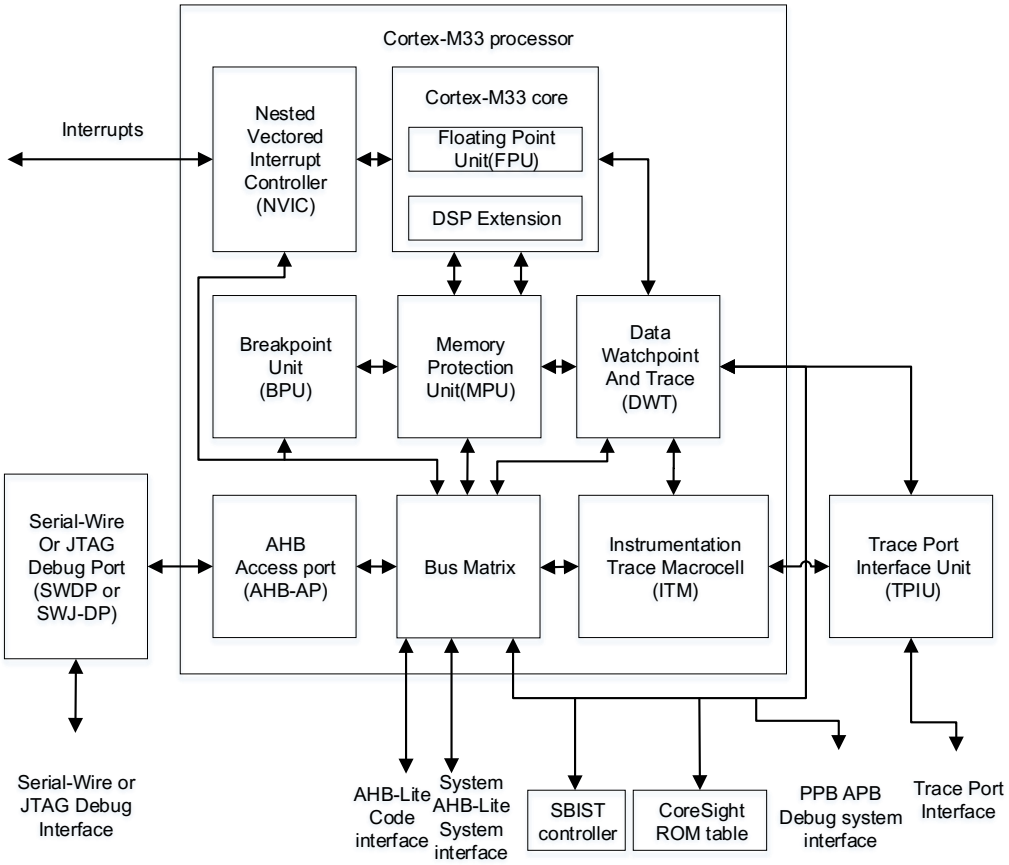
1.1. Arm® Cortex®-M33 processor

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*** 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 GD32E53x 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-F	CBUS-S	SBUS-F	SBUS-S	DMA0	DMA1
FMC					1	1
SRAM0	1		1		1	1
SRAM1	1		1		1	1
AHB			1		1	1

	CBUS-F	CBUS-S	SBUS-F	SBUS-S	DMA0	DMA1
EXMC		1		1	1	1
OSPI		1		1	1	1

As shown in ***Table 1-1***, several masters are connected to the AHB interconnect matrix, including CBUS-F, CBUS-S, SBUS-F, SBUS-S, DMA0 and DMA1 (the suffix -F means Fast and -S means Slow). 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 AHB master ports of the DMA0 and DMA1 controllers respectively.

The GD32E53x series chips are equipped with a master timeout monitoring mechanism to detect system bus transmission timeouts.

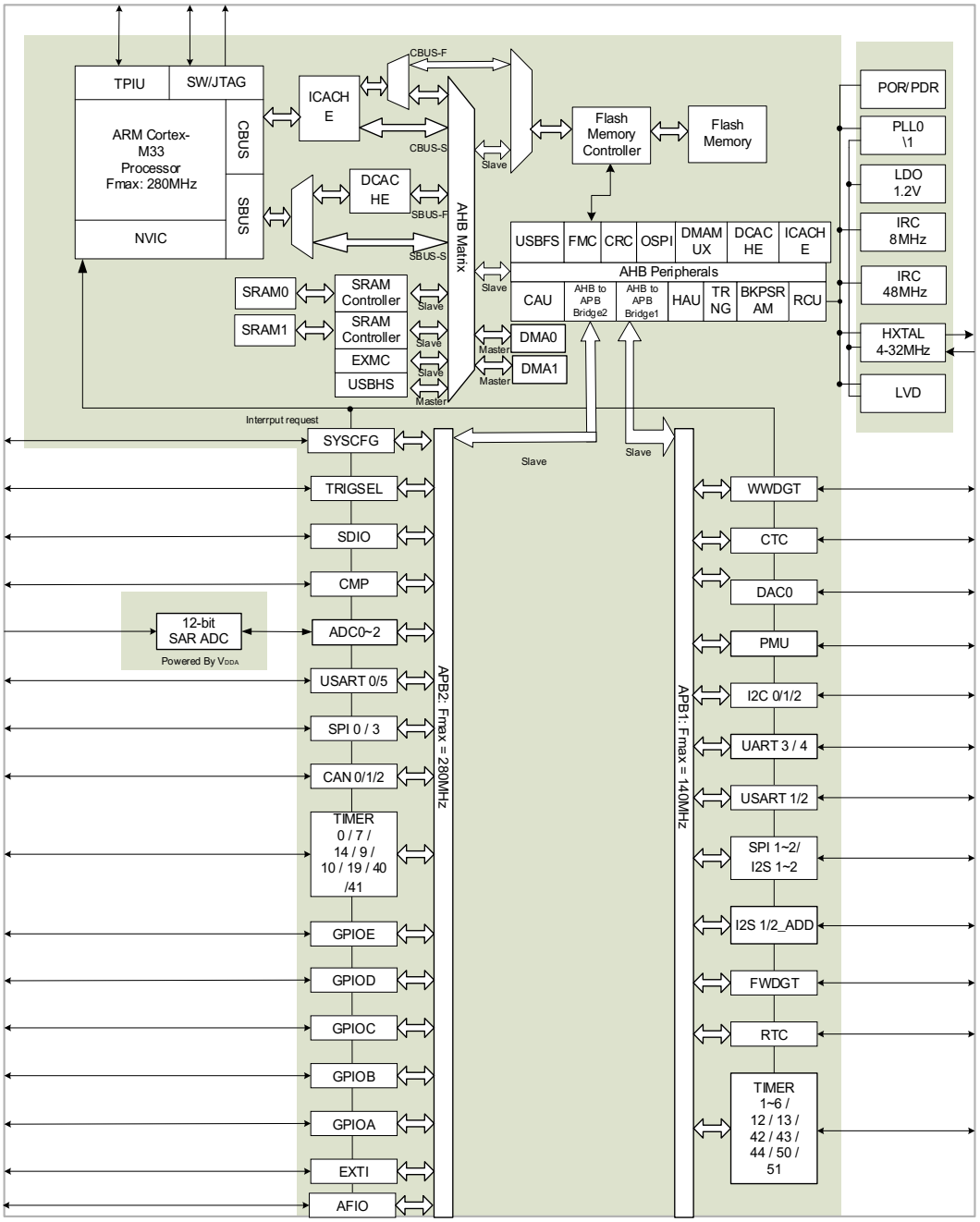
The system monitors timeouts for six AHB master ports (CBUS-F, CBUS-S, SBUS-F, SBUS-S, DMA0, DMA1). When an abnormal access causes prolonged bus access blockage, the hardware automatically generates a bus error and feeds it back to the corresponding master. At the same time, the SYSCFG module sets the corresponding timeout status flag and triggers an NMI interrupt for exception handling.

The timeout monitoring function for the corresponding master can be enabled through the bus timeout register (SYSCFG_BUSTO), and the timeout status flags of each master can be checked through the bus timeout status register (SYSCFG_BUSTOSTAT).

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM0\1, EXMC, OSPI and AHB. 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 126 MHz, APB2 operates at full speed (up to 252MHz depending on the device).

The system architecture of the GD32E53x series devices is shown in ***Figure 1-2***.

Figure 1-2. GD32E53x series system architecture



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 GD32E53x devices*** shows the memory map of the GD32E53x 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 GD32E53x devices

Pre-defined Regions	Bus	Address	Peripherals
External device	AHB3	0xC000 0000 - 0xDFFF FFFF	Reserved
		0xB000 0000 - 0xBFFF FFFF	OSPI(MEM)
		0xA000 1400 - 0xAFFF FFFF	Reserved
		0xA000 1000 - 0xA000 13FF	Reserved
		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/SRAM
Peripheral	AHB1	0x5000 0000 - 0x5003 FFFF	USBFS
		0x4008 0400 - 0x4FFF FFFF	Reserved
		0x4008 0000 - 0x4008 03FF	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	DCACHE
		0x4002 6000 - 0x4002 63FF	ICACHE
		0x4002 5000 - 0x4002 5FFF	OSPI(REG)
		0x4002 4000 - 0x4002 4FFF	BKPSRAM
		0x4002 3C00 - 0x4002 3FFF	TRNG
		0x4002 3800 - 0x4002 3BFF	HAU
		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

Pre-defined Regions	Bus	Address	Peripherals
		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 9C00 - 0x4001 FFFF	Reserved
	APB2	0x4001 9800 - 0x4001 9BFF	SYSCFG
		0x4001 9400 - 0x4001 97FF	Shared APB/CAN2 SRAM
		0x4001 9000 - 0x4001 93FF	TRIGSEL
		0x4001 8D00 - 0x4001 8FFF	Reserved
		0x4001 8C00 - 0x4001 8CFF	Shared APB/CAN SRAM (256 bytes)
		0x4001 8800 - 0x4001 8BFF	Shared APB/CAN SRAM
		0x4001 8400 - 0x4001 87FF	SPI3
		0x4001 8000 - 0x4001 83FF	SDIO
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 7800 - 0x4001 7BFF	Reserved
		0x4001 7400 - 0x4001 77FF	CAN2
		0x4001 7000 - 0x4001 73FF	USART5
		0x4001 6C00 - 0x4001 6FFF	Reserved
		0x4001 6800 - 0x4001 6BFF	Reserved
		0x4001 5C00 - 0x4001 67FF	Reserved
		0x4001 5800 - 0x4001 5BFF	Reserved
		0x4001 5400 - 0x4001 57FF	TIMER10
		0x4001 5000 - 0x4001 53FF	TIMER9
		0x4001 4C00 - 0x4001 4FFF	TIMER40
		0x4001 4800 - 0x4001 4BFF	TIMER19
		0x4001 4400 - 0x4001 47FF	TIMER41
		0x4001 4000 - 0x4001 43FF	TIMER14
		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
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	CAN1

Pre-defined Regions	Bus	Address	Peripherals
		0x4001 1C00 - 0x4001 1FFF	CAN0
		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 F400 - 0x4000 FFFF	Reserved
		0x4000 F000 - 0x4000 F3FF	TIMER51
		0x4000 EC00 - 0x4000 EFFF	TIMER50
		0x4000 E800 - 0x4000 EBFF	TIMER44
		0x4000 E400 - 0x4000 E7FF	TIMER43
		0x4000 E000 - 0x4000 E3FF	TIMER42
		0x4000 D000 - 0x4000 DFFF	Reserved
		0x4000 CC00 - 0x4000 CFFF	CTC1
		0x4000 C800 - 0x4000 CBFF	CTC0
		0x4000 C400 - 0x4000 C7FF	Reserved
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 8000 - 0x4000 BFFF	Reserved
		0x4000 7C00 - 0x4000 7FFF	Reserved
		0x4000 7800 - 0x4000 7BFF	Reserved
		0x4000 7400 - 0x4000 77FF	DAC0
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	Reserved
		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	I2S2_ADD
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	I2S1_ADD
		0x4000 3000 - 0x4000 33FF	FWDGT

Pre-defined Regions	Bus	Address	Peripherals
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	Reserved
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 1C00 - 0x4000 1FFF	TIMER12
		0x4000 1800 - 0x4000 1BFF	Reserved
		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
		0x2002 0000 - 0x2002 FFFF	SRAM1
		0x2000 0000 - 0x2001FFFF	SRAM0
Code	AHB	0x1FFF F850 - 0x1FFF FFFF	Reserved
		0x1FFF F800 - 0x1FFF F84F	Option Bytes
		0x1FFF F000 - 0x1FFF F7FF	Boot loader
		0x1FFF C010 - 0x1FFF EFFF	
		0x1FFF C000 - 0x1FFF C00F	
		0x1FFF B000 - 0x1FFF BFFF	
		0x1FFF 7800 - 0x1FFF AFFF	Reserved
		0x1FFF 6800 - 0x1FFF 77FF	OTP
		0x1FFF 0000 - 0x1FFF 67FF	Reserved
		0x1FFE C010 - 0x1FFE FFFF	External memories
		0x1FFE C000 - 0x1FFE C00F	
		0x1001 0000 - 0x1FFE BFFF	
		0x1000 0000 - 0x1000 FFFF	
		0x0F00 0000 - 0x0FFF FFFF	Reserved
		0x0E03 0000 - 0x0EFF FFFF	Reserved
		0x0E02 0000 - 0x0E02 FFFF	SRAM1
		0x0E00 0000 - 0x0E01 FFFF	SRAM0
		0x083C 0000 - 0x0DFF FFFF	Reserved
		0x0830 0000 - 0x083B FFFF	Reserved
		0x0808 0000 - 0x082F FFFF	Reserved
		0x0800 0000 - 0x0807 FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	Reserved
		0x0010 0000 - 0x002F FFFF	Reserved

Pre-defined Regions	Bus	Address	Peripherals
		0x0008 0000 - 0x000F FFFF	Reserved
		0x0002 0000 - 0x0007 FFFF	External memories
		0x0000 0000 - 0x0001 FFFF	

1.3.1. On-chip SRAM memory

The GD32E53x series of devices contain up to 128 KB of on-chip SRAM0 and 64KB of on-chip SRAM1. It supports byte(8 bits), half-word (16 bits), and word (32 bits) accesses. The 4KB BKPSRAM (Backup SRAM) is used in the backup domain. It can retain its contents even when the VDD power supply is lost or the system enters standby mode.

SRAM write protection

SRAM0 supports write protection for the first 64KB, with a page granularity of 1KB for write protection. Write protection function of SRAM page can be enabled by setting SRAM_PxWP(x=0…31) and SRAM_PxWP(x=32…63) bits in write protection register (SYSCFG_SRAMWP0, SYSCFG_SRAMWP1) and write protection function can be removed/cleared by a system reset only. A write to a write-protected SRAM page generates a HardFault.

ECC

When reading and writing first 64KB SRAM0, it supports 7-bit ECC function. It can correct 1-bit error and detect two-bit errors.

It must be written before reading SRAM0, otherwise it may cause ECC error. Unaligned read operations will be performed in accordance with 32-bit read operations. Unaligned write operations will produce a read-modify-write process. For example, when 16-bit data is written into SRAM0, the other 16 bits of the word are first read out from SRAM0 and combined with the 16-bit data to be written, and then the resulting 32-bit data is written back to SRAM0. Therefore, when initializing SRAM0, it can only be written in a 32-bit width.

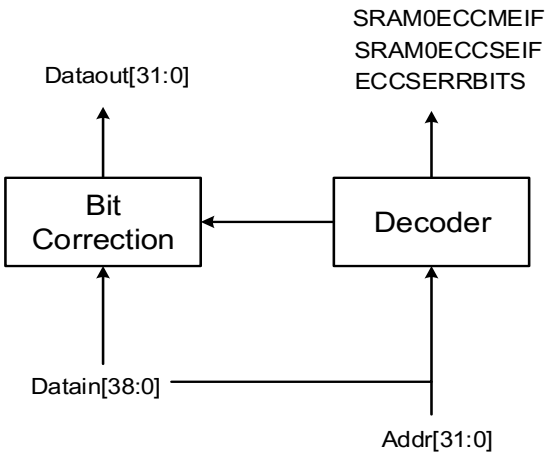
The ECC module is composed of an encoder and a decoder.

Encoder: When performing a SRAM0 write operation, a 7-bit ECC code will be generated and written into the SRAM0 together with the data.

Decoder: When performing a SRAM0 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***:

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 SRAM0ECCSEIF bit in SYSCFG_ECCS1 register will be set. Software can clear it by writing 1.
(2) The SYSCFG_ECCCFG0 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 SRAM0ECCMEIF bit in SYSCFG_ECCS1 register will be set. Software can clear it by writing 1.

(2) The SYSCFG_ECCCFG0 records the address where the multi-bits non-correction error event occurred.

Single bit correction error interrupt

Set the SRAM0ECCSEIE bit in SYSCFG_ECCCFG0 register. When a single-bit error correctable event is detected, SRAM ECC interrupt will be generated.

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

Set the SRAM0ECCMEIE bit in SYSCFG_ECCCFG0 register. When a multi-bits error non-correction event is detected, NMI and SRAM ECC 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 512KB 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 GD32E53x series provide three kinds of boot sources which can be selected by the BOOT0 pin and boot configuration bits BOOTLK, nBOOT1, SWBT0 and nBOOT0 in the User option byte. The details are shown in the following table ***Table 1-3***. When SWBT0=1, the value on the BOOT0 pin is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the boot mode configuration after a power-on reset or a system reset to select the required boot source. When SWBT0=1, once the boot mode configuration have been sampled, they are free and can be used for other purposes.

Table 1-3. Boot modes

Selected boot area	Boot mode configuration				
	BOOTLK	nBOOT1 bit	BOOT0 pin	SWBT0 bit	nBOOT0 bit
Main Flash memory	1	x	x	0	1
System memory	1	1	x	0	0
Embedded SRAM	1	0	x	0	0
Main Flash memory	1	x	0	1	x
System memory	1	1	1	1	x
Embedded SRAM	1	0	1	1	x
Main Flash memory	0	x	x	x	x

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

According to the selected boot source, either the main flash memory (original memory space beginning at 0x0800 0000) or the system memory (original memory space beginning at 0x1FFF B000) is aliased in the boot memory space which begins at the address 0x0000 0000. When the on-chip SRAM whose memory space is beginning at 0x2000 0000 is selected as the boot source, in the application initialization code, you have to relocate the vector table in SRAM using the NVIC exception table and offset register.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory. GD32E53x MCU embedded bootloader supports multiple interfaces to update

the Flash memory. There will be USART0/USART1 port, and standard USB port can be used on GD32E53x line products.

Ability to use the BOOTLK bit to force boot from a unique entry point in the Main Flash memory regardless of other boot mode configurations.

■ Remap configuration

When the boot mode configurations are selected, the software can configure the memory remap function through BIT bits in memory remap configuration register (FMC_OBCTL). Memories including main flash memory, system memory, embedded SRAM can be remapped.

1.5. System configuration registers

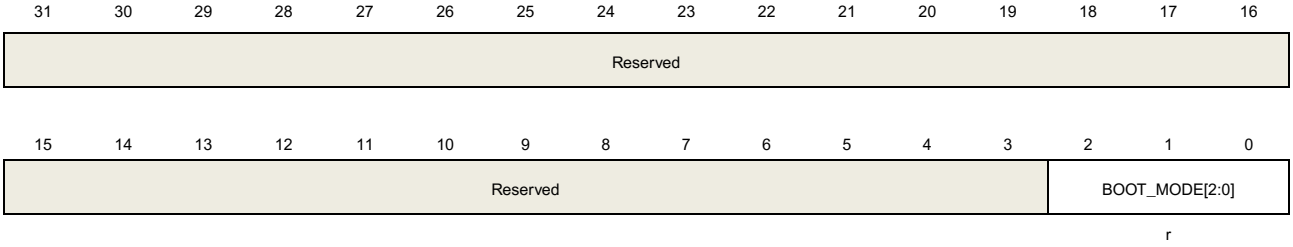
SYSCFG base address: 0x4001 9800

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 pin and the boot configuration bits in the User option byte)

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



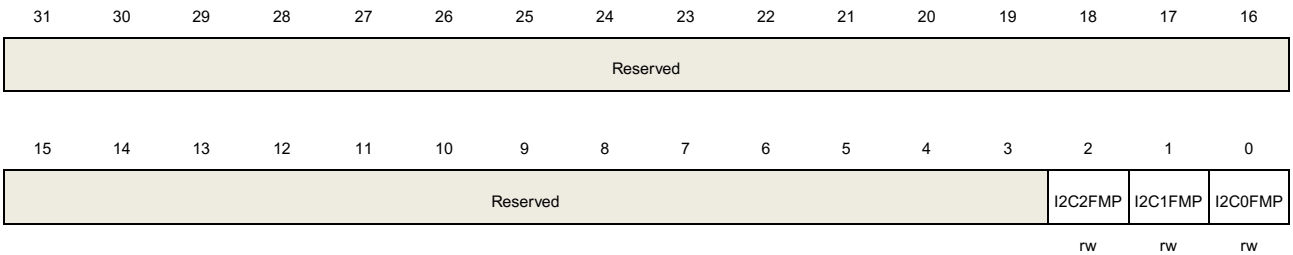
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	BOOT_MODE[2:0]	Boot mode. Refer to ***Table 1-3***

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
31:3	Reserved	Must be kept at reset value.
2	I2C2FMP	I2C2 Fm+ This bit enables Fm+ on I2C2. 0: Fm+ disabled 1: Fm+ enabled

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: 0x18
Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															IWAY1ECCMERRLK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWAY1ECCMERRLK	IWAY0ECCMERRLK	DWAY0ECCMERRLK	SRAM0ECCMERRLK	Reserved				BKPSRAMECCMERRLK	CPU_LOCK	Reserved			PVD_LOCK	Reserved	
rw	rw	rw	rw					rw	rw				rw		

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	IWAY1ECCMERRLK	ICACHE way1 SRAM ECC multi-bits non-correction error lock bit This bit is set by software and cleared by a system reset. 0: ICACHE way1 SRAM ECC multi-bits non-correction error signal disconnected from the break input of TIMER0/7/14/19/40~44 1: ICACHE way1 SRAM ECC multi-bits non-correction error signal connected to the break input of TIMER0/7/14/19/40~44
15	DWAY1ECCMERRLK	DCACHE way1 SRAM ECC multi-bits non-correction error lock bit This bit is set by software and cleared by a system reset. 0: DCACHE way1 SRAM ECC multi-bits non-correction error signal disconnected from the break input of TIMER0/7/14/19/40~44 1: DCACHE way1 SRAM ECC multi-bits non-correction error signal connected to the break input of TIMER0/7/14/19/40~44
14	IWAY0ECCMERRLK	ICACHE way0 SRAM ECC multi-bits non-correction error lock bit This bit is set by software and cleared by a system reset. 0: ICACHE way0 SRAM ECC multi-bits non-correction error signal disconnected from the break input of TIMER0/7/14/19/40~44

		1: ICACHE way0 SRAM ECC multi-bits non-correction error signal connected to the break input of TIMER0/7/14/19/40~44
13	DWAY0ECCMERRLK	DCACHE way0 SRAM ECC multi-bits non-correction error lock bit This bit is set by software and cleared by a system reset. 0: DCACHE way0 SRAM ECC multi-bits non-correction error signal disconnected from the break input of TIMER0/7/14/19/40~44 1: DCACHE way0 SRAM ECC multi-bits non-correction error signal connected to the break input of TIMER0/7/14/19/40~44
12	SRAM0ECCMERRLK	SRAM0 ECC multi-bits non-correction error lockup bit This bit is set by software and cleared by a system reset. 0: SRAM ECC multi-bits non-correction error signal is disconnected from the break input of TIMER0/7/14/19/40~44 1: SRAM ECC multi-bits non-correction error signal is connected to the break input of TIMER0/7/14/19/40~44
11:8	Reserved	Must be kept at reset value.
7	BKPSRAMECCMERRLK	BKPSRAM ECC multi-bits non-correction lockup bit This bit is set by software and cleared by a system reset. 0: BKPSRAM ECC multi-bits non-correction signal disconnected from the break input of TIMER0/7/14/19/40~44 1: BKPSRAM ECC multi-bits non-correction signal connected to the break input of TIMER0/7/14/19/40~44
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 TIMER0/7/14/19/40~44 1: CPU lockup signal is connected to the break input of TIMER0/7/14/19/40~44
5:3	Reserved	Must be kept at reset value.
2	PVD_LOCK	Programmable voltage detector lockup bit. This bit is set by software and cleared by a system reset. 0: PVD signal is disconnected from the break input of TIMER0/7/14/19/40~44 1: PVD signal is connected to the break input of TIMER0/7/14/19/40~44
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	CPUSBUSSTO	CPUSBUSFTO	CPUCBUSSTO	CPUCBUSFTO
										rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	DMA1BUSTO	DMA1 bus timeout enable bit. 0: Disable 1: Enable
4	DMA0BUSTO	DMA0 bus timeout enable bit. 0: Disable 1: Enable
3	CPUSBUSSTO	CPU Sbus-Slow timeout enable bit. 0: Disable 1: Enable
2	CPUSBUSFTO	CPU Sbus-Fast timeout enable bit. 0: Disable 1: Enable
1	CPUCBUSSTO	CPU Cbus-Slow timeout enable bit. 0: Disable 1: Enable
0	CPUCBUSFTO	CPU Cbus-Fast timeout enable bit. 0: Disable 1: Enable

1.5.5. SRAM ECC error status register 0 (SYSCFG_ECCS0)

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
DWAY1ECCERRBITS[5:0]						DWAY0ECCERRBITS[5:0]						IWAY1ECCERRBITS[5:2]			
r						r						r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWAY1ECCERRBITS[1:0]		IWAY0ECCERRBITS[5:0]						Reserved							
r		r													

Bits	Fields	Descriptions

31:26	DWAY1ECCERRBITS[5:0]	Indicates the error bit of the DCACHE way1 Which bit has a DCACHE way1 single-bit correctable error 000000: no error 000001: bit 0 … 100111: bit 38
25:20	DWAY0ECCERRBITS[5:0]	Indicates the error bit of the DCACHE way0 Which bit has a DCACHE way0 single-bit correctable error. 000000: no error 000001: bit 0 … 100111: bit 38
19:14	IWAY1ECCERRBITS[5:0]	Indicates the error bit of the ICACHE way1 Which bit has a ICACHE way1 single-bit correctable error. 000000: no error 000001: bit 0 … 100111: bit 38
13:8	IWAY0ECCERRBITS[5:0]	Indicates the error bit of the ICACHE way0 Which bit has a ICACHE way0 single-bit correctable error. 000000: no error 000001: bit 0 … 100111: bit 38
7:0	Reserved	Must be kept at reset value.

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	IDIE	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: Inexact interrupt disable 1: Inexact interrupt enable
4	IDIE	Input denormal interrupt enable bit 0: Input denormal interrupt disable 1: Input denormal interrupt enable
3	OVFIE	Overflow interrupt enable bit 0: Overflow interrupt disable 1: Overflow interrupt enable
2	UFIE	Underflow interrupt enable bit 0: Underflow interrupt disable 1: Underflow interrupt enable
1	DZIE	Divide by 0 interrupt enable bit 0: Divide by 0 interrupt disable 1: Divide by 0 interrupt enable
0	IOPIE	Invalid operation interrupt enable bit 0: Invalid operation interrupt disable 1: Invalid operation interrupt enable

1.5.7. SRAM write protection register 0 (SYSCFG_SRAMWP0)

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

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 error status register 1 (SYSCFG_ECCS1)

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				DWAY1ECCSEIF	DWAY1ECCMEIF	DWAY0ECCSEIF	DWAY0ECCMEIF	IWAY1ECCSEIF	IWAY1ECCMEIF	IWAY0ECCSEIF	IWAY0ECCMEIF	BKPECCSEIF	BKPECCMEIF	SRAM0ECCSEIF	SRAM0ECCMEIF
				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
31:12	Reserved	Must be kept at reset value.
11	DWAY1ECCSEIF	DCACHE way1 SRAM single-bit correction event flag The software can clear it by writing 1. 0: no DCACHE way1 SRAM single-bit correction event is detected. 1: DCACHE way1 SRAM single-bit correction event is detected.
10	DWAY1ECCMEIF	DCACHE way1 SRAM multi-bits non-correction error flag The software can clear it by writing 1. 0: no DCACHE way1 SRAM non-correction event is detected. 1: DCACHE way1 SRAM non-correction event is detected.
9	DWAY0ECCSEIF	DCACHE way0 SRAM single-bit correction event flag The software can clear it by writing 1. 0: no DCACHE way0 SRAM single-bit correction event is detected. 1: DCACHE way0 SRAM single-bit correction event is detected.
8	DWAY0ECCMEIF	DCACHE way0 SRAM multi-bits non-correction error flag The software can clear it by writing 1. 0: no DCACHE way0 SRAM non-correction event is detected. 1: DCACHE way0 SRAM non-correction event is detected.
7	IWAY1ECCSEIF	ICACHE way1 SRAM single-bit correction event flag The software can clear it by writing 1. 0: no ICACHE way1 SRAM single-bit correction event is detected. 1: ICACHE way1 SRAM single-bit correction event is detected.
6	IWAY1ECCMEIF	ICACHE way1 SRAM multi-bits non-correction error flag The software can clear it by writing 1. 0: no ICACHE way1 SRAM non-correction event is detected. 1: ICACHE way1 SRAM non-correction event is detected.

5	IWAY0ECCSEIF	ICACHE way0 SRAM single-bit correction event flag The software can clear it by writing 1. 0: no ICACHE way0 SRAM single-bit correction event is detected. 1: ICACHE way0 SRAM single-bit correction event is detected.
4	IWAY0ECCMEIF	ICACHE way0 SRAM multi-bits non-correction error flag The software can clear it by writing 1. 0: no ICACHE way0 SRAM non-correction event is detected. 1: ICACHE way0 SRAM non-correction event is detected.
3	BKPECCSEIF	BKPSRAM single-bit correction event flag The software can clear it by writing 1. 0: no BKPSRAM single-bit correction event is detected. 1: BKPSRAM single-bit correction event is detected.
2	BKPECCMEIF	BKPSRAM multi-bits non-correction error flag The software can clear it by writing 1. 0: no BKPSRAM non-correction event is detected. 1: BKPSRAM non-correction event is detected.
1	SRAM0ECCSEIF	SRAM0 single-bit correction error flag The software can clear it by writing 1. 0: no SRAM0 single-bit correction event is detected 1: SRAM0 single-bit correction event is detected
0	SRAM0ECCMEIF	SRAM0 multi-bits non-correction error flag The software can clear it by writing 1. 0: no SRAM0 non-correction event is detected 1: SRAM0 non-correction event is detected

1.5.9. SRAM ECC configuration register 0 (SYSCFG_ECCCFG0)

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
SRAM0ECCEADDR[13:0]														Reserved	
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCSERRBITS[5:0]						Reserved					SRAM0ECCSEIE	SRAM0ECCMEIE	Reserved		FLASHECCMEIE
r											rw	rw			rw

Bits	Fields	Descriptions

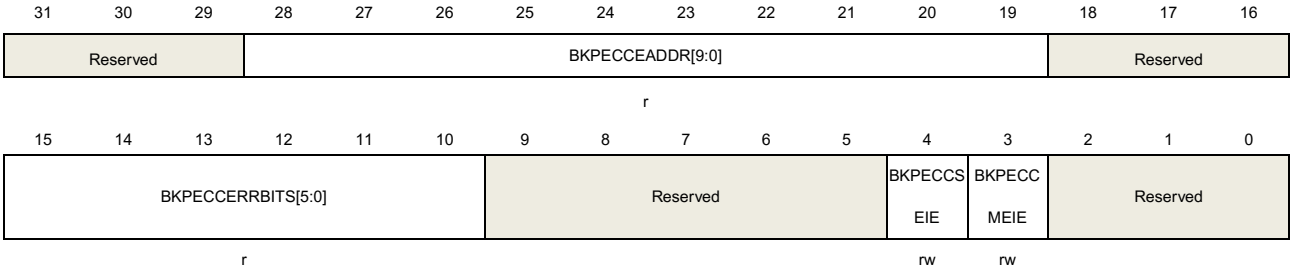
31:18	SRAM0ECCEADDR[13:0]	Record the faulting system address where the last SRAM ECC event on SRAM0 occurred, Physical address = SRAM0 base address (0x2000 0000) + SRAM0ECCEADDR × 4.
17:16	Reserved	Must be kept at reset value.
15:10	ECCSERRBITS[5:0]	Indicates the error bit Which bit has an SRAM0 ECC single-bit correctable error 000000: no error 000001: bit 0 … 100111: bit 38
9:5	Reserved	Must be kept at reset value.
4	SRAM0ECCSEIE	SRAM0 single-bit correction error interrupt enable 0: SRAM single-bit correction error interrupt is disabled. 1: SRAM single-bit correction error interrupt is enabled
3	SRAM0ECCMEIE	SRAM0 multi-bits non-correction error NMI interrupt enable 0: SRAM non-correction error NMI interrupt is disabled. 1: SRAM non-correction error NMI interrupt is enabled.
2:1	Reserved	Must be kept at reset value.
0	FLASHECCMEIE	Flash multi-bits non-correction error NMI interrupt enable 0: Flash non-correction error NMI interrupt is disabled. 1: Flash non-correction error NMI interrupt is enabled.

1.5.10. ECC configuration register 1 (SYSCFG_ECCCFG1)

Address offset: 0x34

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	BKPECCEADDR[9:0]	Indicates the address of the last ECC event on BKPSRAM. BKPSRAM base address = 0x4002 4000

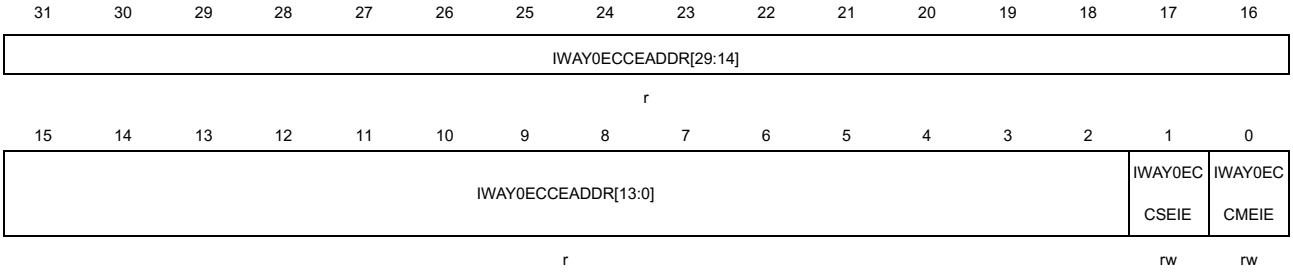
		Physical BKPSRAM ECC word address = BKPSRAM base address + BKPECCEADDR × 4
18:16	Reserved	Must be kept at reset value.
15:10	BKPECCERRBITS[5:0]	Indicates the error bit of the BKPSRAM Which bit has a BKPSRAM ECC single-bit correctable error 000000: no error 000001: bit 0 … 100111: bit 38
9:5	Reserved	Must be kept at reset value.
4	BKPECCSEIE	BKPSRAM single-bit correction error interrupt enable 0: BKPSRAM single-bit correction error interrupt is disabled. 1: BKPSRAM single-bit correction error interrupt is enabled
3	BKPECCMEIE	BKPSRAM multi-bits non-correction error NMI interrupt enable 0: BKPSRAM non-correction error NMI interrupt is disabled. 1: BKPSRAM non-correction error NMI interrupt is enabled.
2:0	Reserved	Must be kept at reset value.

1.5.11. SRAM ECC configuration register 2 (SYSCFG_ECCCFG2)

Address offset: 0x38

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	IWAY0ECCEADDR[29:0]	Indicates the last ECC event on ICACHE way0 SRAM occurred.
1	IWAY0ECCSEIE	ICACHE way0 single-bit correction error interrupt enable 0: ICACHE way0 SRAM single-bit correction error interrupt is disabled. 1: ICACHE way0 SRAM single-bit correction error interrupt is enabled.
0	IWAY0ECCMEIE	ICACHE way0 multi-bits non-correction error NMI interrupt enable 0: ICACHE way0 SRAM non-correction error NMI interrupt is disabled.

1: ICACHE way0 SRAM non-correction error NMI interrupt is enabled.

1.5.12. SRAM ECC configuration register 3 (SYSCFG_ECCCFG3)

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
IWAY1ECCEADDR[29:14]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWAY1ECCEADDR[13:0]														IWAY1ECCSEIE	IWAY1ECCMEIE
r														rw	rw

Bits	Fields	Descriptions
31:2	IWAY1ECCEADDR[29:0]	Indicates the last ECC event on ICACHE way1 SRAM occurred.
1	IWAY1ECCSEIE	ICACHE way1 single-bit correction error interrupt enable 0: ICACHE way1 SRAM single-bit correction error interrupt is disabled. 1: ICACHE way1 SRAM single-bit correction error interrupt is enabled.
0	IWAY1ECCMEIE	ICACHE way1 multi-bits (two bits) non-correction error NMI interrupt enable 0: ICACHE way1 SRAM non-correction error NMI interrupt is disabled. 1: ICACHE way1 SRAM non-correction error NMI interrupt is enabled.

1.5.13. Bus timeout status register (SYSCFG_BUSTOSTAT)

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										DMA1BUSTOF	DMA0BUSTOF	CPUSBUSSTOF	CPUSBUSFTOF	CPUCBUSSTOF	CPUCBUSFTOF
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	DMA1BUSTOF	DMA1 bus timeout flag. 0: DMA1 bus not timeout

		1: DMA1 bus timeout
4	DMA0BUSTOF	DMA0 bus timeout flag. 0: DMA0 bus not timeout 1: DMA0 bus timeout
3	CPUSBUSSTOF	CPU Sbus-Slow timeout flag. 0: CPU Sbus-S not timeout 1: CPU Sbus-S timeout
2	CPUSBUSFTOF	CPU SBUS-Fast timeout flag. 0: CPU Sbus-F not timeout 1: CPU Sbus-F timeout
1	CPUCBUSSTOF	CPU Cbus-Slow timeout flag. 0: CPU Cbus-S not timeout 1: CPU Cbus-S timeout
0	CPUCBUSFTOF	CPU Cbus-Fast timeout flag. 0: CPU Cbus-F not timeout 1: CPU Cbus-F timeout

1.5.14. SRAM write protection register1 (SYSCFG_SRAMWP1)

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
SRAM_P63WP	SRAM_P62WP	SRAM_P61WP	SRAM_P60WP	SRAM_P59WP	SRAM_P58WP	SRAM_P57WP	SRAM_P56WP	SRAM_P55WP	SRAM_P54WP	SRAM_P53WP	SRAM_P52WP	SRAM_P51WP	SRAM_P50WP	SRAM_P49WP	SRAM_P48WP
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
SRAM_P47WP	SRAM_P46WP	SRAM_P45WP	SRAM_P44WP	SRAM_P43WP	SRAM_P42WP	SRAM_P41WP	SRAM_P40WP	SRAM_P39WP	SRAM_P38WP	SRAM_P37WP	SRAM_P36WP	SRAM_P35WP	SRAM_P34WP	SRAM_P33WP	SRAM_P32WP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	SRAM_PxWP (x=32 to 63)	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.

1.5.15. Timer input selection register (SYSCFG_TIMERCISEL)

Address offset: 0x48

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		TIMER42_CI1_SEL[1:0]		Reserved	TIMER42_CI0_SEL[2:0]			Reserved		TIMER41_CI1_SEL[1:0]		Reserved	TIMER41_CI0_SEL[2:0]		
		rw			rw					rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TIMER40_CI1_SEL[1:0]		Reserved	TIMER40_CI0_SEL[2:0]			Reserved		TIMER14_CI1_SEL[1:0]		Reserved	TIMER14_CI0_SEL[2:0]		
		rw			rw					rw			rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:28	TIMER42_CI1_SEL[1:0]	Selects TIMER42_CI1 input 00: TIMER42_CH1 input 01: TIMER4_CH1 input 10: Reserved 11: Reserved
27	Reserved	Must be kept at reset value.
26:24	TIMER42_CI0_SEL[2:0]	Selects TIMER42_CI0 input 000: TIMER42_CH0 input 001: TIMER4_CH0 input 010: Reserved 011: Reserved 100: LXTAL 101: Reserved 110: MCO 111: Reserved
23:22	Reserved	Must be kept at reset value.
21:20	TIMER41_CI1_SEL[1:0]	Selects TIMER41_CI1 input 00: TIMER41_CH1 input 01: TIMER3_CH1 input 10: TIMER4_CH1 input 11: Reserved
19	Reserved	Must be kept at reset value.
18:16	TIMER41_CI0_SEL[2:0]	Selects TIMER41_CI0 input 000: TIMER41_CH0 input 001: TIMER3_CH0 input 010: TIMER4_CH0 input 011: Reserved 100: LXTAL 101: Reserved 110: MCO 111: Reserved
15:14	Reserved	Must be kept at reset value.

13:12	TIMER40_CI1_SEL[1:0]	Selects TIMER40_CI1 input 00: TIMER40_CH1 input 01: TIMER2_CH1 input 10: TIMER3_CH1 input 11: TIMER4_CH1 input
11	Reserved	Must be kept at reset value.
10:8	TIMER40_CI0_SEL[2:0]	Selects TIMER40_CI0 input 000: TIMER40_CH0 input 001: TIMER2_CH0 input 010: TIMER3_CH0 input 011: TIMER4_CH0 input 100: LXTAL 101: Reserved 110: MCO 111: Reserved
7:6	Reserved	Must be kept at reset value.
5:4	TIMER14_CI1_SEL[1:0]	Selects TIMER14_CI1 input 00: TIMER14_CH1 input 01: TIMER1_CH1 input 10: TIMER2_CH1 input 11: TIMER3_CH1 input
3	Reserved	Must be kept at reset value.
2:0	TIMER14_CI0_SEL[2:0]	Selects TIMER14_CI0 input 000: TIMER14_CH0 input 001: TIMER1_CH0 input 010: TIMER2_CH0 input 011: TIMER3_CH0 input 100: LXTAL 101: Reserved 110: MCO 111: Reserved

1.5.16. SRAM ECC configuration register 4 (SYSCFG_ECCCFG4)

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
DWAY0ECCEADDR[29:14]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWAY0ECCEADDR[13:0]														DWAY0ECCSEIE	DWAY0ECCMEIE
r														rw	rw

Bits	Fields	Descriptions
31:2	DWAY0ECCEADDR[29:0]	Indicates the last ECC event on DCACHE way0 SRAM occurred.
1	DWAY0ECCSEIE	DCACHE way0 single-bit correction error interrupt enable 0: DCACHE way0 SRAM single-bit correction error interrupt is disabled. 1: DCACHE way0 SRAM single-bit correction error interrupt is enabled.
0	DWAY0ECCMEIE	DCACHE way0 multi-bits non-correction error NMI interrupt enable 0: DCACHE way0 SRAM non-correction error NMI interrupt is disabled. 1: DCACHE way0 SRAM non-correction error NMI interrupt is enabled.

1.5.17. SRAM ECC configuration register5 (SYSCFG_ECCCFG5)

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
DWAY1ECCEADDR[29:14]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWAY1ECCEADDR[13:0]														DWAY1ECCSEIE	DWAY1ECCMEIE
r														rw	rw

Bits	Fields	Descriptions
31:2	DWAY1ECCEADDR[29:0]	Indicates the last ECC event on DCACHE way1 SRAM occurred.
1	DWAY1ECCSEIE	DCACHE way1 single-bit correction error interrupt enable 0: DCACHE way1 SRAM single-bit correction error interrupt is disabled. 1: DCACHE way1 SRAM single-bit correction error interrupt is enabled.
0	DWAY1ECCMEIE	DCACHE way1 multi-bits (two bits) non-correction error NMI interrupt enable 0: DCACHE way1 SRAM non-correction error NMI interrupt is disabled. 1: DCACHE way1 SRAM non-correction error NMI interrupt is enabled.

1.5.18. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=0, 7, 19)

Address offset: 0x100 for TIMER0
Address offset: 0x13C for TIMER7
Address offset: 0x184 for TIMER19
Reset value: 0x0000 0000

This register has 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				

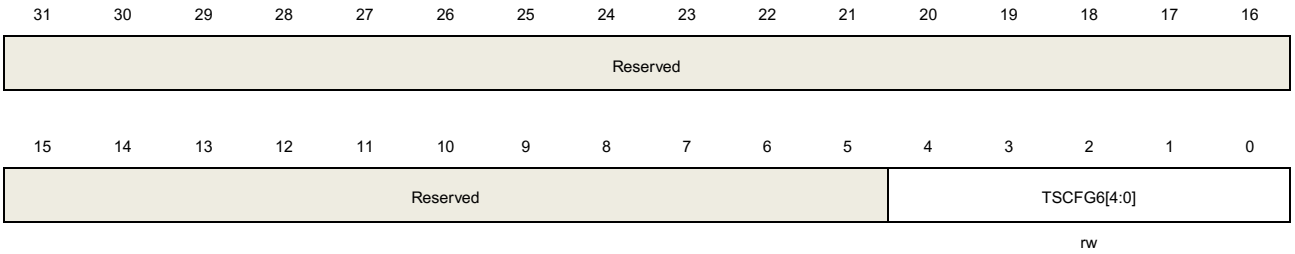
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (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: 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
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.

1.5.19. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=0, 7, 19)

Address offset: 0x104 for TIMER0
Address offset: 0x140 for TIMER7
Address offset: 0x188 for TIMER19
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.
4:0	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

1.5.20. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=0, 7, 19)

Address offset: 0x108 for TIMER0
Address offset: 0x144 for TIMER7
Address offset: 0x18C for TIMER19
Reset value: 0x0000 0000

This register has 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: Reserved 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 **Note:** When TSCFG15[4:0] is used, TSCFGy[4:0](y=0…6) should be zero, otherwise, the ITS trigger cooperates with TSCFGy[4:0], and input source depends on TSCFGy[4:0].
15:0	Reserved	Must be kept at reset value.

1.5.21. 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

This register has 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)

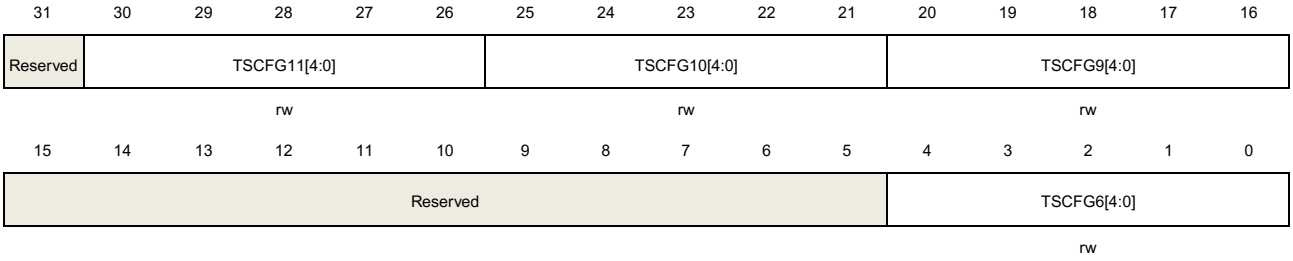
		Others: Reserved
25:21	TSCFG4[4:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 00000: Pause mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (ITI2) 00100: Internal trigger input 3 (ITI3) 00101: 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
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.

1.5.22. 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

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



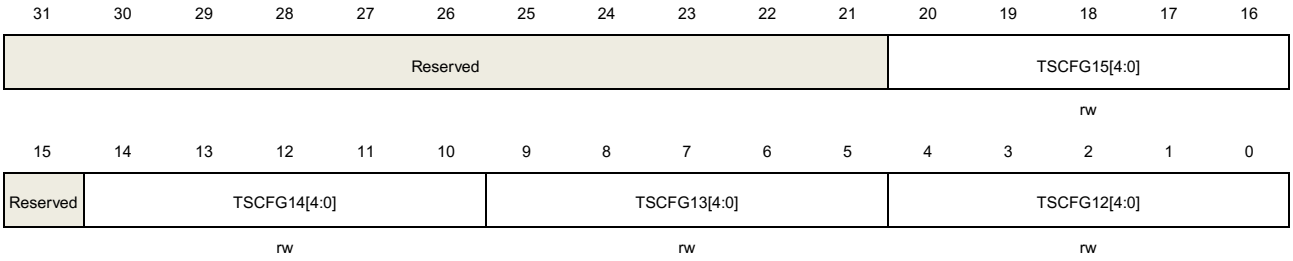
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG11[4:0]	Decoder mode 2 configuration 00000: disable Others: The counter will count on both rising and falling edges of CI0FE0 and CI1FE1 signals. And the counting direction determined by the CH0P and CH1P bits.
25:21	TSCFG10[4:0]	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]	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

1.5.23. 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

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: Reserved 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 **Note:** When TSCFG15[4:0] is used, TSCFGy[4:0](y=0…6, 9…14) should be zero, otherwise, the ITS trigger cooperates with TSCFGy[4:0], and input source depends 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 CI1FE0 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 CI0FE1 level.
4:0	TSCFG12[4:0]	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.

1.5.24. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=9, 10, 12, 13)

Address offset: 0x148 for TIMER9
Address offset: 0x154 for TIMER10
Address offset: 0x160 for TIMER12
Address offset: 0x16C for TIMER13
Reset value: 0x0000 0000

This register has 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.

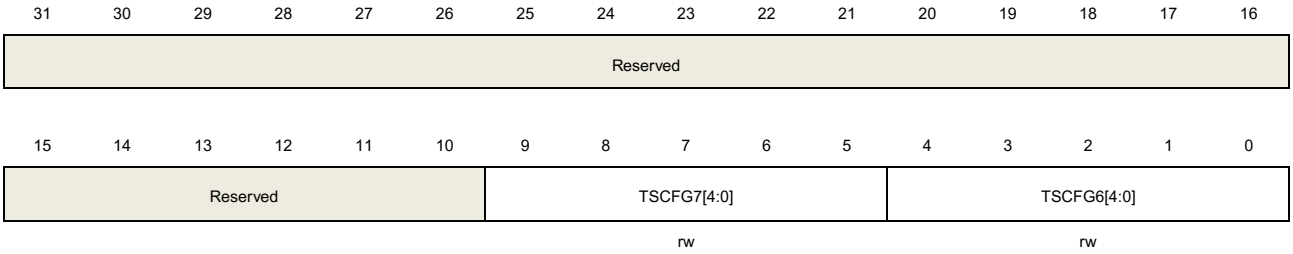
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
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) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 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) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
15	Reserved	Must be kept at reset value.
14:10	TSCFG2[4:0]	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.
9:5	TSCFG1[4:0]	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.
4:0	TSCFG0[4:0]	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.

1.5.25. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=9, 10, 12, 13)

Address offset: 0x14C for TIMER9
Address offset: 0x158 for TIMER10
Address offset: 0x164 for TIMER12
Address offset: 0x170 for TIMER13
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:5	TSCFG7[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) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
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) 00101: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 10011: Internal trigger input 14 (ITI14) Others: Reserved

1.5.26. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=9, 10, 12, 13)

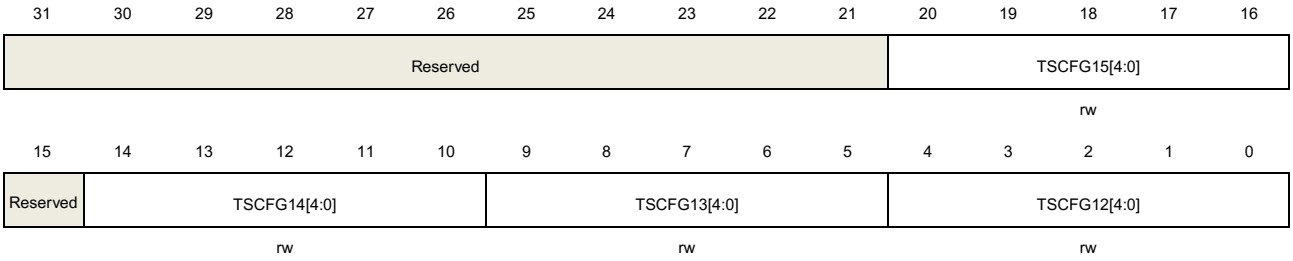
Address offset: 0x150 for TIMER9
Address offset: 0x15C for TIMER10

Address offset: 0x168 for TIMER12

Address offset: 0x174 for TIMER13

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.
20:16	TSCFG15[4:0]	Internal trigger input source configuration 00000: Reserved 00001: Internal trigger input 0 (ITI0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
15	Reserved	Must be kept at reset value.
14:10	TSCFG14[4:0]	Quadrature decoder mode 2 configuration 00000: Disable Others: The counter counts on CI1FE1 edge only, while the direction depends on CI1FE1 level.
9:5	TSCFG13[4:0]	Quadrature decoder mode 1 configuration 00000: Disable Others: The counter counts on CI0FE0 edge only, while the direction depends on CI0FE0 level.
4:0	TSCFG12[4:0]	Decoder mode 2 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.

1.5.27. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0) (x=14, 40~44)

Address offset: 0x178 for TIMER14

Address offset: 0x190 for TIMER40

Address offset: 0x19C for TIMER41

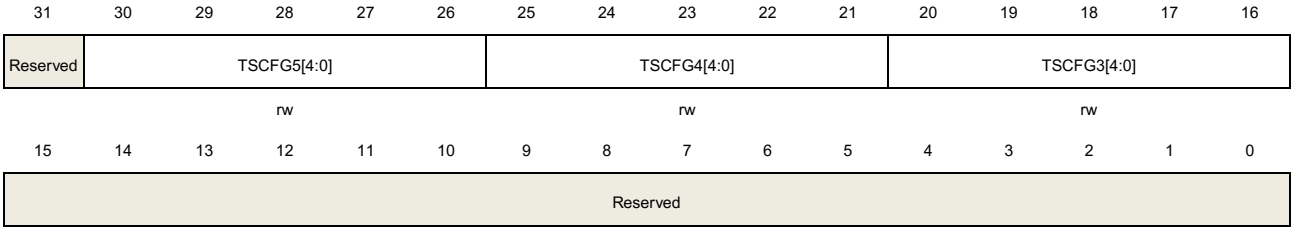
Address offset: 0x1A8 for TIMER42

Address offset: 0x1B4 for TIMER43

Address offset: 0x1C0 for TIMER44

Reset value: 0x0000 0000

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



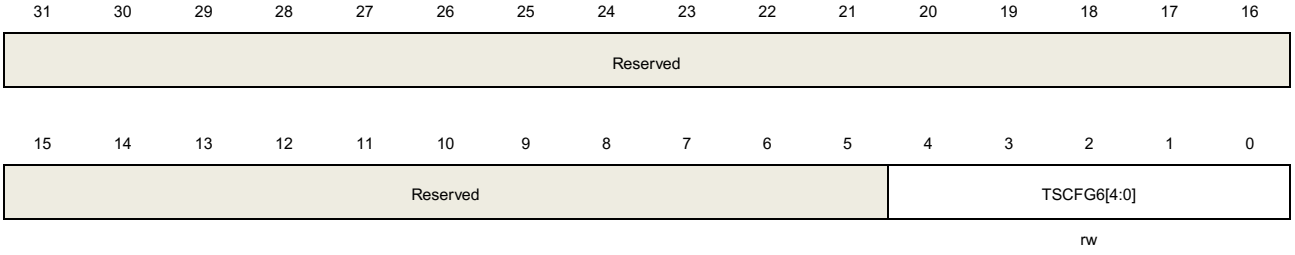
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:26	TSCFG5[4:0]	Event mode configuration A rising edge of the trigger input enables the counter. 00000: Event mode disable 00001: Internal trigger input 0 (ITI0) 00010: Internal trigger input 1 (ITI1) 00011: Internal trigger input 2 (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: The filtered output of multi mode channel 0 input (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: CI0 edge flag (CI0F_ED) 00110: The filtered output of channel 0 input (CI0FE0) 00111: The filtered output of channel 1 input (CI1FE1) 01011: The filtered output of multi mode channel 0 input (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: The filtered output of multi mode channel 0 input (MCI0FEM0) 10011: Internal trigger input 14 (ITI14) Others: Reserved
15:0	Reserved	Must be kept at reset value.

1.5.28. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1) (x=14, 40~44)

Address offset: 0x17C for TIMER14
Address offset: 0x194 for TIMER40
Address offset: 0x1A0 for TIMER41
Address offset: 0x1AC for TIMER42
Address offset: 0x1B8 for TIMER43
Address offset: 0x1C4 for TIMER44
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.
4:0	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: The filtered output of multi mode channel 0 input (MCI0FEM0)
10011: Internal trigger input 14 (ITI14)
Others: Reserved

1.5.29. TIMERx configuration register 2 (SYSCFG_TIMERxCFG2) (x=14, 40~44)

Address offset: 0x180 for TIMER14
Address offset: 0x198 for TIMER40
Address offset: 0x1A4 for TIMER41
Address offset: 0x1B0 for TIMER42
Address offset: 0x1BC for TIMER43
Address offset: 0x1C8 for TIMER44
Reset value: 0x0000 0000

This register has 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: Reserved 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: The filtered output of multi mode channel 0 input (MCI0FEM0) 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.

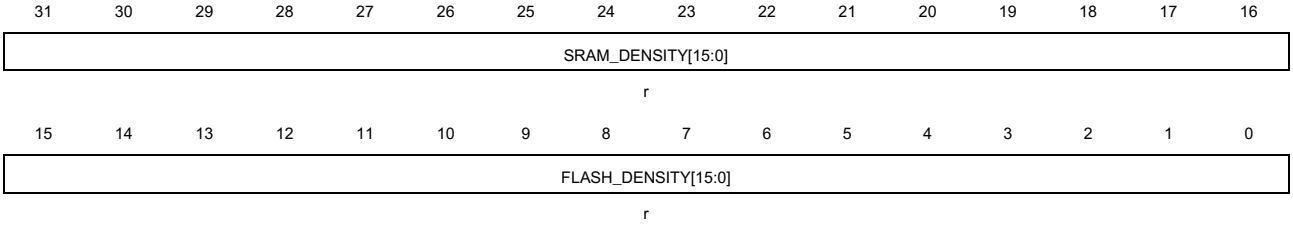
1.6. Device electronic signature

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.

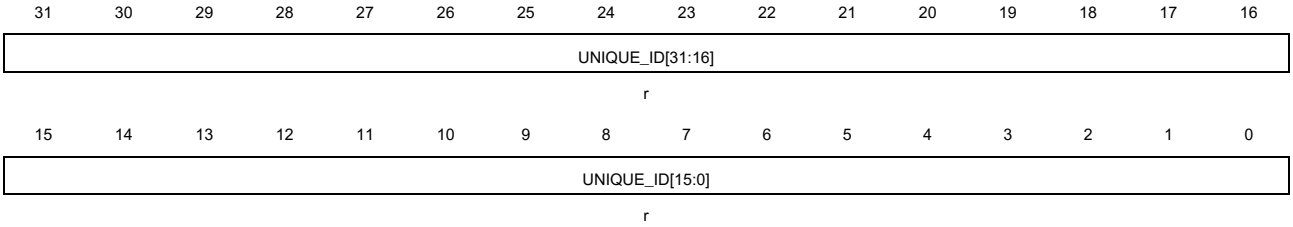


Bits	Fields	Descriptions
31:16	SRAM_DENSITY [15:0]	SRAM density The value indicates the on-chip SRAM density of the device in Kbytes. Example: 0x0008 indicates 8 Kbytes.
15:0	FLASH_DENSITY [15:0]	Flash memory density The value indicates the Flash memory density of the device in Kbytes. Example: 0x0020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



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. Flash memory controller (FMC)

2.1. Overview

The flash memory controller, FMC, provides all the necessary functions for the on-chip flash memory. A small waiting time is needed while the CPU executes instructions stored in the 512KB of flash. It provides page erase, mass erase, and programming operations for flash memory.

2.2. Characteristics

- Up to 512KB of on-chip flash memory for instruction and data, Up to 18KB bootloader. Up to 2KB user OTP. 2KB SEC OTP, 80B option bytes.
- 0~6 wait states (0~5 wait states for GD32E533) within 512KB when CPU executes instructions.
- Flash memory supports 128-bits wide data reading
- Pre-fetch buffer to speed read operations.
- The flash page size is 2KB.
- Four-word programming, page erase and mass erase operation.
- Provides secure user area (SCR).
- ECC with single bit error correction and multi-bits errors detection.
- 2KB OTP(one-time program) block used for user data storage.
- 80B option bytes block for user application requirements.
- Option bytes are uploaded to the option byte control registers when the system is reset.
- Provides secure OTP protection.
- Provides read address offset function to support OTA.
- Page erase / program protection to prevent unexpected operation.

2.3. Function overview

2.3.1. Flash memory architecture

The Flash memory consists of up to 512 KB main Flash organized into 256 pages with 2 KB capacity per page and an 18 KB Information Block for the Boot Loader. The main Flash memory contains a total of up to 256 pages which can be erased individually. The following table shows the base address, size.

The following table shows the details of flash organization.

Table 2-1. GD32E53x base address and size for flash memory

Block	Name	Address range	Size(bytes)
Main flash block	Page 0	0x0800 0000 - 0x0800 07FF	2KB
	Page 1	0x0800 0800 - 0x0800 0FFF	2KB
	Page 2	0x0800 1000 - 0x0800 17FF	2KB
	Page 255	0x0807 F800 - 0x0807 FFFF	2KB
Information block	Boot loader area	0x1FFF B000- 0x1FFF F7FF	18KB
One-time program block	SEC OTP	0x1FFF_6800~0x1FFF_6FFF	2KB
	USER OTP	0x1FFF_7000~0x1FFF_77FF	2KB
Option bytes block	Option bytes	0x1FFF F800 - 0x1FFF F84F	80B

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

2.3.2. Error checking and correcting (ECC)

The ECC mechanism supports:

- One bit error detection and correction
- Two bits errors detection
- Support 128-bit data and 9-bit ECC code for Flash
- Support hardware EDC after ECC feature to check the ECC logic
- 1 bit error status and error address registers

When one error is detected and corrected:

- When ECC error occurs, the ECCCOR bit in FMC_ECCCS register will be set. If the ECCCORIE bit in FMC_ECCCS register is set, an interrupt is generated. The ECCEAR bits in the FMC_ECCADDR register indicate the ECC error address in each space. The ECCCOR bit in FMC_ECCCS register can be cleared by writing 1 to ECCCOR bit.

When multi-bits or Error Detection Code (EDC) are detected:

- When ECC error occurs, the ECCDET bit in FMC_ECCCS register will be set. If the FLASHECCMEIE bit in SYSCFG_ECCCFG0 register is set, an NMI interrupt is generated. The ECCADDR bits notice the ECC error address in each space. And the data return all F. The ECCDET bit in FMC_ECCCS register can be cleared by writing 1 to ECCDET bit.

Note:

1. Data in Flash memory are 128-bits words: 9 bits are added per four-words (128 bits), which the added 9 bits are the ECC code and calculated by hardware and cannot be accessed by user.

2. For a virgin data all 128 bit 1, ECC code stored in flash is 0x1FF.
3. When a new ECC error occurs, only the address of the first error that occurred (single-bit or multi-bit) is recorded.
4. When an ECC error is reported, a new read at the error address may not generate an ECC error if the data is still present in the current buffer / prefetch buffer, even if ECCCOR and ECCDET bits are cleared.
5. Support ECC error injection by setting the ECCINJEN and ECCINJDATA[1:0] bits in the FMC_ECCCS register.

2.3.3. Read operations

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

Wait state added:

The WSCNT bits in the FMC_WS register need to be configured correctly depending on the AHB clock frequency when reading the flash memory. The relationship between WSCNT and AHB clock frequency is shown in the ***Table 2-2. The relation between WSCNT and AHB clock frequency***.

Table 2-2. The relation between WSCNT and AHB clock frequency(for GD32E533)

AHB clock frequency	WSCNT configured
<= 38MHz	0 (0 wait state added)
<= 72MHz	1 (1 wait state added)
<= 114MHz	2 (2 wait state added)
<= 152MHz	3 (3 wait state added)
<= 190MHz	4 (4 wait state added)
<= 200MHz	5 (5 wait state added) (For GD32E533)

Table 2-3. The relation between WSCNT and AHB clock frequency(For GD32E535 and GD32E53P)

AHB clock frequency	WSCNT configured
<= 38MHz	0 (0 wait state added)
<= 72MHz	1 (1 wait state added)
<= 114MHz	2 (2 wait state added)
<= 152MHz	3 (3 wait state added)
<= 190MHz	4 (4 wait state added)
<= 220MHz	5 (5 wait state added)
<= 252MHz	6 (6 wait state added) (For GD32E535 and GD32E53P)

Table 2-4. The relation between WSCNT and AHB clock frequency(For GD32E537)

AHB clock frequency	WSCNT configured
<= 38MHz	0 (0 wait state added)
<= 72MHz	1 (1 wait state added)
<= 114MHz	2 (2 wait state added)
<= 152MHz	3 (3 wait state added)
<= 190MHz	4 (4 wait state added)
<= 220MHz	5 (5 wait state added)
<= 280MHz	6 (6 wait state added) (For GD32E537)

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

Note:

1. If want to increase the AHB clock frequency. First, refer to the ***Table 2-2. The relation between WSCNT and AHB clock frequency*** table between WSCNT bit and AHB clock frequency, configure the WSCNT bits according to the target AHB clock frequency. Then, increase the AHB clock frequency to the target frequency. It is forbidden to increase the AHB clock frequency before configure the WSCNT.

2. If want to decrease the AHB clock frequency. First, decrease the AHB clock frequency to target frequency. Then refer to the ***Table 2-2. The relation between WSCNT and AHB clock frequency*** table between WSCNT bit and AHB clock frequency, configure the WSCNT bits according the target AHB clock frequency. It is forbidden to configure the WSCNT bits before decrease the AHB clock frequency.

Because wait states are added, the read efficiency is very low (such as, add 6 wait state when 252MHz). In order to speed up the read access, there are some functions performed.

Current buffer:

The current buffer is always enabled. Each time data is read from Flash, a 128-bit line is fetched and stored in the current buffer. The CPU only needs 32-bit or 16-bit in each read operation. So in the case of sequential code, the next data can get from current buffer without refetching from Flash.

Pre-fetch buffer:

The pre-fetch buffer is enabled by setting the PFEN bit in the FMC_WS register. The pre-fetch buffer is performed only on the instruction bus (IBUS). In the case of sequential code, when the CPU execute the current buffer data (128-bit), it takes at least 4 clocks for 32-bit operation and at least 8 clocks for 16-bit operation. In this case, pre-fetch the data of next 128-bit (four-word) address from flash memory and store to pre-fetch buffer. So when the CPU finishes the current buffer and needs to execute the next data, the pre-fetch buffer hit.

Read address offset function

The read offset function can be configured to increase the bus initial address by adding an offset and then read it from FMC. After setting the read offset value and area by configuring FMC_OFRG and FMC_OFVR, reading the value of the source address is equivalent to reading the value of the add offset address. This function can be used to support OTA.

Conditions for the read offset function to take effect:

1. The address is within the pages of OF_SPAGE and OF_EPAGE.

2. OF_SPAGE must be beyond the SCR region.

Note:

1. The offset function only supports read operation, not support program operation and erase operation.

2. OF_SPAGE, OF_EPAGE, and OF_VALUE are all in units of page (2KB).

3. To protect SCR region, If the OF_SPAGE offset range overlaps the SCR, the OFERR flag is set and the offset function is disabled.

2.3.4. Unlock the FMC_CTL register and FMC_OBCTL register

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

The FMC_OBCTL register, OBSTART bit in FMC_CTL are protected by FMC_OBKEY register. The unlocking sequence includes two write operations, which are orderly writing 0x45670123 and 0xCDEF89AB to FMC_OBKEY register, then hardware reset the OBLK bit in FMC_CTL register to 0. The software can set OBLK bit to 1 to protect the FMC_OBCTL register, OBSTART bit in FMC_CTL register again.

2.3.5. Page erase

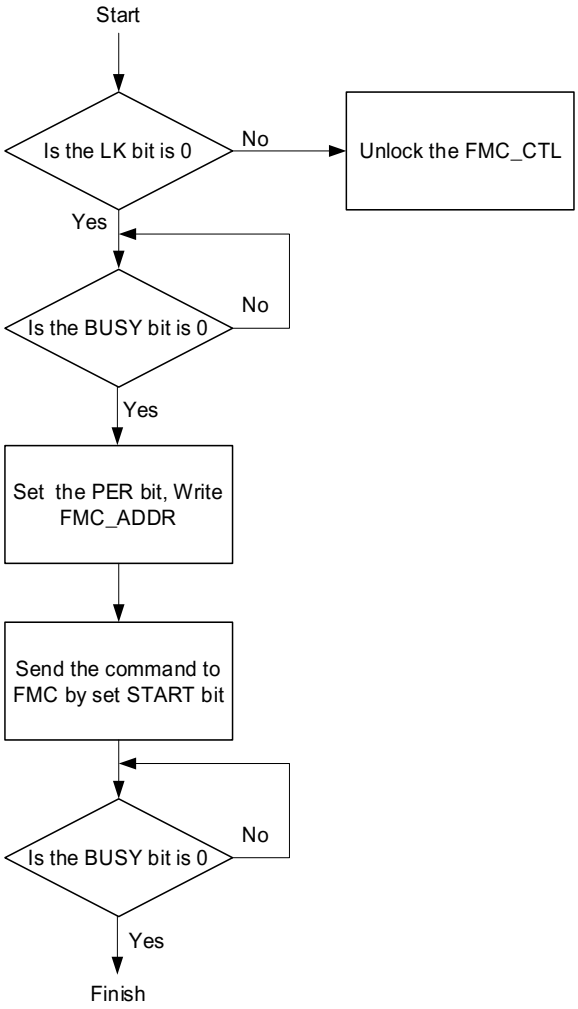
The FMC provides a page erase function which is used to initialize the contents of a main flash memory page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the registers for a page erase operation.

- Unlock the FMC_CTL register if necessary.

- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals 0). Otherwise, wait until the operation has finished.
- Set the PER bit in the FMC_CTL register.
- Write the page absolute address (0x08XX XXXX) into the FMC_ADDR registers.
- Send the page erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the page if required.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that a correct target page address must be confirmed. Otherwise, the software may run out of control if the target erase page is being used to fetch codes or access data. The FMC will not provide any notification when that happens. 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_CTL register is set. The software can check the WPERR bit in the FMC_STAT register to detect this condition in the interrupt handler. The following figure shows the page erase operation flow.

Figure 2-1. Process of page erase operation



2.3.6. Mass erase

The FMC provides a mass erase function which is used to initialize the main flash block contents. This erase can affect entire flash block by setting the MER bit to 1 in the FMC_CTL register. The following steps show the mass erase register access sequence.

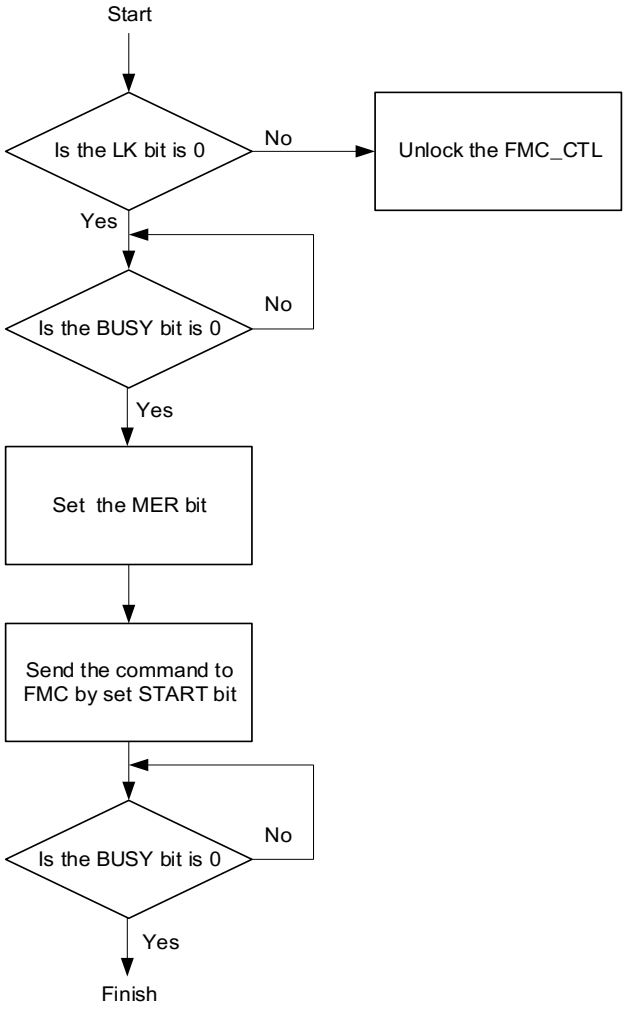
- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals 0). Otherwise, wait until the operation has finished.
- Set the MER bit in the FMC_CTL register if erase entire flash.
- Send the mass erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory if required.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set,

and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register 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 using the debugging tool that accesses the FMC registers directly. Additionally, the mass erase operation will be ignored if any page is erase/program protected. In this condition, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the WPERR bit in the FMC_STAT register to detect this condition in the interrupt handler.

The following figure indicates the mass erase operation flow.

Figure 2-2. Process of mass erase operation



2.3.7. Main flash programming

The FMC provides a 128-bit (four-word) programming function which is used to modify the main flash memory contents.

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

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals 0). Otherwise, wait until the operation has finished.
- Set the PG bit in the FMC_CTL register.
- Write the data to be programmed with desired absolute address (0x08XX XXXX). The DBUS write to form a 128-bit data and then the 128-bit data program to flash memory. The data to be programmed must be aligned to 128 bits (four words).
- Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STAT registers.
- Read and verify the flash memory if required.

When the operation is executed successfully, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that before the programming operation, check that the target address has been erased. If the address has not been erased, the PGERR bit is set when programming the address, except if programming 0x0. Each 128 bits (four words) can be programmed only once after erase and before next erase. Note that the PG bit must be set before the programming operation. Additionally, the program operation will be ignored on erase/program protected pages and the WPERR bit in the FMC_STAT will be set.

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

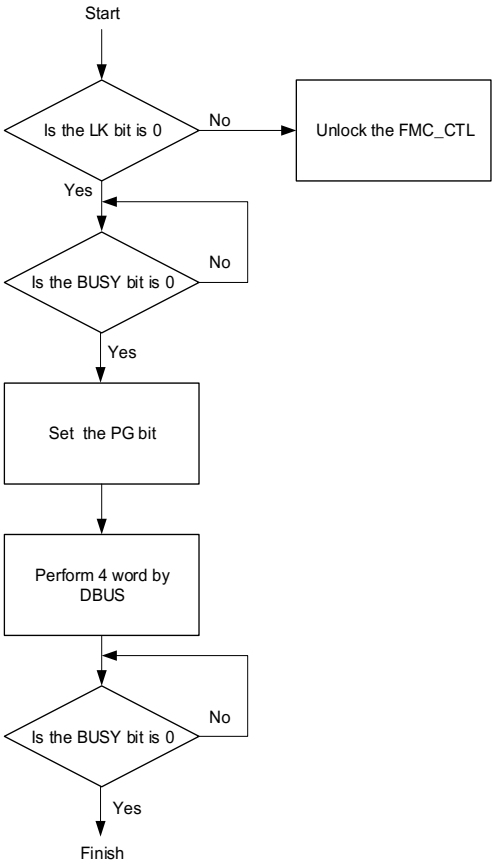
- The program use a byte write (not 32-bit write)
- The programmed size does not match the previous size. Mixing 32-bit and 16-bit writes is not allowed.
- The write is not aligned.
- A write access occurs before the PG bit is set.
- Two or more of PER, MER, PG, and OBSTART are set simultaneously.

Note:

1. If fewer than 128 bits (four words) are written, the data is not programmed to the flash memory and no error is reported.

In these conditions, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the PGERR bit, PGAERR bit, PGSERR bit or WPERR bit in the FMC_STAT register to detect which condition occurred in the interrupt handler.

Figure 2-3. Process of four-word (128-bit) program operation



Note:

1. Reading the flash should be avoided when a program/erase operation is ongoing. And flash memory accesses will fail if the CPU enters the power saving modes.

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

2.3.8. OTP programming

The OTP programming method is the same as the main flash programming. The OTP block can only be programmed once and cannot be erased.

Note: It must ensure the OTP programming sequence completely without any unexpected interrupt, such as system reset or power down. If unexpected interrupt occurs, there is a small probability that the stored data will be corrupted in flash memory.

Table 2-5. OTP bytes

Address	Fields		Descriptions	

		Attributes	Bits	Fields	Descriptions	Factory Value
OTP_SCRCTL						
0x1fff_6800	OTP_SCRCTL	rw	[31:16]	SCR_CTL_LK	OTP Lock Bit. Once locked, it becomes read-only. The lock value is 0xA55A, and any other value indicates the unlocked state.	0xFFFF
			[15:8]	Reserved	-	0x7F
			[7:3]	SCR_BLOCK_CNT[4:0]	Configure the secure user area block count. 0x00: No secure user area 0x01: 1 block secure user area 0x02: 2 blocks secure user area … 0x10: 16 blocks secure user area 0x11-0x1E: 16 blocks secure user area 0x1F: No secure user area	0x1F
			[2:1]	Reserved	-	0x3
			[0]	BOOTLK	Startup Address Lock 1: Select startup address based on pins and option bytes 0: Lock startup from main flash	0x1
0x1fff_6804	Reserved	-	-	-	-	-
0x1fff_6808	Reserved	-	-	-	-	-
0x1fff_680C	Reserved	-	-	-	-	-
0x1fff_6810	DBGPSWD_DATA0[31:0]	w	[31:0]		Secure JTAG Key 0	0xFFFFFFFF
0x1fff_6814	DBGPSWD_DATA1[31:0]	w	[31:0]		Secure JTAG Key 1	0xFFFFFFFF
0x1fff_6818	DBGPSWD_DATA2[31:0]	w	[31:0]		Secure JTAG Key 2	0xFFFFFFFF
0x1fff_681C	DBGPSWD_DATA3[31:0]	w	[31:0]		Secure JTAG Key 3	0xFFFFFFFF
0x1fff_6820	OTP_DBG_CTL	rw	[31:16]	DBG_CTL_LK	Lock the DBG-related fields (DBGPSWD_DATA0-3 and OTP_DBG_CTL). Once locked, they cannot be written. The lock value is	0xFFFF

					0x3CC3, and other values indicate unlocking.	
			[15:3]	Reserved		
			[2 : 1]	NDBG	Configure debugging behavior: 0x: No debugging 10: Secure debugging 11: Normal debugging	0x3
			[0]	JTAGNSW	Switch SW/JTAG debugging 1: Enable SW debugging 0: Enable JTAG debugging	0x1
0x1fff_6824	Reserved	-	-	-	-	-
0x1fff_6828	Reserved	-	-	-	-	-
0x1fff_682C	Reserved	-	-	-	-	-
AES key region						
0x1fff_6830	OTP_KEY_0	w	[31:0]	OTP_KEY_0_0L	AES Key 0	0xFFFFFFFF
0x1fff_6834			[31:0]	OTP_KEY_0_0H		0xFFFFFFFF
0x1fff_6838			[31:0]	OTP_KEY_0_1L		0xFFFFFFFF
0x1fff_683C			[31:0]	OTP_KEY_0_1H		0xFFFFFFFF
0x1fff_6840			[31:0]	OTP_KEY_0_2L		0xFFFFFFFF
0x1fff_6844			[31:0]	OTP_KEY_0_2H		0xFFFFFFFF
0x1fff_6848			[31:0]	OTP_KEY_0_3L		0xFFFFFFFF
0x1fff_684C			[31:0]	OTP_KEY_0_3H		0xFFFFFFFF
0x1fff_6850	OTP_KEY_1	w	[31:0]	OTP_KEY_1_0L	AES Key 1	0xFFFFFFFF
0x1fff_6854			[31:0]	OTP_KEY_1_0H		0xFFFFFFFF
0x1fff_6858			[31:0]	OTP_KEY_1_1L		0xFFFFFFFF
0x1fff_685C			[31:0]	OTP_KEY_1_1H		0xFFFFFFFF
0x1fff_6860			[31:0]	OTP_KEY_1_2L		0xFFFFFFFF
0x1fff_6864			[31:0]	OTP_KEY_1_2H		0xFFFFFFFF

0x1fff_6868			[31:0]	OTP_KEY_1_3L		0xFFFFFFFF
0x1fff_686C			[31:0]	OTP_KEY_1_3H		0xFFFFFFFF
0x1fff_6870	OTP_KEY_2	w	[31:0]	OTP_KEY_2_0L	AES Key 2	0xFFFFFFFF
0x1fff_6874			[31:0]	OTP_KEY_2_0H		0xFFFFFFFF
0x1fff_6878			[31:0]	OTP_KEY_2_1L		0xFFFFFFFF
0x1fff_687C			[31:0]	OTP_KEY_2_1H		0xFFFFFFFF
0x1fff_6880			[31:0]	OTP_KEY_2_2L		0xFFFFFFFF
0x1fff_6884			[31:0]	OTP_KEY_2_2H		0xFFFFFFFF
0x1fff_6888			[31:0]	OTP_KEY_2_3L		0xFFFFFFFF
0x1fff_688C			[31:0]	OTP_KEY_2_3H		0xFFFFFFFF
0x1fff_6890	OTP_KEY_3	w	[31:0]	OTP_KEY_3_0L	AES Key 3	0xFFFFFFFF
0x1fff_6894			[31:0]	OTP_KEY_3_0H		0xFFFFFFFF
0x1fff_6898			[31:0]	OTP_KEY_3_1L		0xFFFFFFFF
0x1fff_689C			[31:0]	OTP_KEY_3_1H		0xFFFFFFFF
0x1fff_68A0			[31:0]	OTP_KEY_3_2L		0xFFFFFFFF
0x1fff_68A4			[31:0]	OTP_KEY_3_2H		0xFFFFFFFF
0x1fff_68A8			[31:0]	OTP_KEY_3_3L		0xFFFFFFFF
0x1fff_68AC			[31:0]	OTP_KEY_3_3H		0xFFFFFFFF
OTP USER DATA region						
0x1fff_68B0-0x1fff_6FFF	USER_DATA0	rw				All F
0x1fff_7000-0x1fff_73FF	USER_DATA1	rw			There are 64 blocks in total, each block contains 16 bytes. The write lock is controlled by the FMC_OTPDATA1_LLK and	All F

					FMC_OTPDATA1_HLK registers, with each bit controlling one block. "0" indicates that the block is write-locked (application: e.g., user status, lifecycle, etc.).	
0x1fff_7400-0x1fff_77FF	USER_DATA2	rw			There are 64 blocks in total, each block contains 16 bytes. The write lock is controlled by the FMC_OTPDATA2_LLK and FMC_OTPDATA2_HLK registers, with each bit controlling one block. "0" indicates that the block is write-locked. The read lock is controlled by the FMC_OTPDATA2_LR_CTL and FMC_OTPDATA2_HR_CTL registers, with each bit controlling one block. "0" indicates that the block is read-locked. (Application: e.g, storing public key hash values)	All F

Forcing boot from Flash memory

To increase the security and establish a chain of trust, the BOOTLK bit of the OTP_SCRCTL allows forcing the system to boot from the Main Flash memory regardless the other boot options.

2.3.9. Option bytes

Option bytes description

The option bytes block is reloaded into the FMC_OBCTL registers after each system reset, then the option bytes take effect. The complement option bytes are the opposite of the option bytes. When the option bytes are reloaded, if the complement option bytes and option bytes do not match, the OBERR bit in the FMC_STAT register will be set, and the option bytes will be set to 0xFF. The OBERR bit will not be set if both the option bytes and its complement bytes are 0xFF.The following table shows the details of the option bytes.

Table 2-6. Option bytes

Address	Name	Description
0x1fff f800	SPC	Option byte security protection value 0xA5 : no security protection any value except 0xA5 or 0xCC : low protection level 0xCC : protection level high
0x1fff f801	SPC_N	SPC complement value

Address	Name	Description
0x1fff f802	USER[7:0]	[7:5]: BOR_TH Level Selection 000: 3.0V rising threshold, 2.9V falling threshold 001: 2.9V rising threshold, 2.8V falling threshold 010: 2.7V rising threshold, 2.6V falling threshold 011: 2.6V rising threshold, 2.5V falling threshold 100: 2.5V rising threshold, 2.4V falling threshold 101: 2.3V rising threshold, 2.2V falling threshold 110: 2.1V rising threshold, 2.0V falling threshold 111: 1.85V rising threshold, 1.75V falling threshold [4]: BOR_EN 0:BOR enable 1: BOR disable [3]: SRAM_ECCEN 0: SRAM ECC disable 1: SRAM ECC enable [2]: nRST_STDBY 0: generate a reset instead of entering standby mode 1: no reset when entering standby mode [1]: nRST_DPSLP 0: generate a reset instead of entering deep-sleep mode 1: no reset when entering deep-sleep mode [0]: nFWDG_HW 0: hardware free watchdog 1: software free watchdog
0x1fff f803	USER_N[7:0]	USER complement value
0x1fff f804	USER[15:8]	[7:5] : BOREF_TH (Flash undervoltage threshold selection) 000: 1.61V rising threshold, 1.57V falling threshold 001: 1.66V rising threshold, 1.62V falling threshold 010: 1.71V rising threshold, 1.67V falling threshold 011: 1.76V rising threshold, 1.72V falling threshold 100: 1.81V rising threshold, 1.77V falling threshold 101~111: 1.86V rising threshold, 1.82V falling threshold [4]: FWDGTSPD_DPSLP 0: FWDGT deep-sleep mode freeze option status bit 1: FWDGT keep running in deep-sleep mode [3]: FWDGTSPD_STANDBY 0: FWDGT Standby mode freeze option status bit 1: FWDGT keep running in Standby mode [2]: nBOOT0 0: BOOT0 signal is 1 1: BOOT0 signal is 0 [1]: nBOOT1

Address	Name	Description
		0: BOOT1 signal is 1 1: BOOT1 signal is 0 Together with the BOOT0 signal, this bit selects boot mode [0]: nSWBT0 [default value 0] 1: BOOT0 signal depends on PB11/BOOT0 pin 0: BOOT0 signal depends on the option bit nBOOT0
0x1fff f805	USER_N[15:8]	USER complement value bit 15 to 8
0x1fff f806	USER[23:16]	[2]: DCACHE ECCEN (The default factory value is 1) 0: disable DCACHE ECC 1: enable DCACHE ECC [1]: ICACHE ECCEN (The default factory value is 1) 0: disable ICACHE ECC 1: enable ICACHE ECC [0]: SRAM_RST 0: Initialize SRAM after power reset 1: Do not initialize SRAM after power reset;
0x1fff f807	USER_N[23:16]	USER complement value bit 23 to 16
0x1fff f808	OB_SCR_WLK[7:0]	SCRWLK[x] indicates SCR data block x is write lock
0x1fff f809	OB_SCR_WLK_N[7:0]	OB_SCR_WLK [7:0] complement value
0x1fff f80a	OB_SCR_WLK[15:8]	SCRWLK[x] indicates SCR data block x is write lock
0x1fff f80b	OB_SCR_WLK_N[15:8]	OB_SCR_WLK[15:8] complement value
0x1fff f80c	Reserved	
0x1fff f80d	Reserved	
0x1fff f80e	Reserved	
0x1fff f80f	Reserved	
0x1fff f810	WP0_SADDR[7:0]	WP0_SADDR contains the first page of Write protect area 0.
0x1fff f811	WP0_SADDR_N[7:0]	WP0_SADDR complement
0x1fff f812	Reserved	
0x1fff f813	Reserved	
0x1fff f814	WP0_EADDR[7:0]	WP0_EADDR contains the last page of Write protect area 0.
0x1fff f815	WP0_EADDR_N[7:0]	WP0_EADDR complement
0x1fff f816	Reserved	
0x1fff f817	Reserved	
0x1fff f818	WP1_SADDR[7:0]	WP1_SADDR contains the first page of Write protect area 1.
0x1fff f819	WP1_SADDR_N[7:0]	WP1_SADDR complement
0x1fff f81A	Reserved	
0x1fff f81B	Reserved	

Address	Name	Description
0x1fff f81C	WP1_EADDR[7:0]	WP1_EADDR contains the last page of Write protect area 1.
0x1fff f81D	WP1_EADDR_N[7:0]	WP1_EADDR complement
0x1fff f81E	Reserved	
0x1fff f81F	Reserved	
0x1fff f820	WP2_SADDR[7:0]	WP2_SADDR contains the first page of Write protect area 2.
0x1fff f821	WP2_SADDR_N[7:0]	WP2_SADDR complement
0x1fff f822	Reserved	
0x1fff f823	Reserved	
0x1fff f824	WP2_EADDR[7:0]	WP2_EADDR contains the last page of Write protect area 2.
0x1fff f825	WP2_EADDR_N[7:0]	WP2_EADDR complement
0x1fff f826	Reserved	
0x1fff f827	Reserved	
0x1fff f828	WP3_SADDR[7:0]	WP3_SADDR contains the first page of Write protect area 3.
0x1fff f829	WP3_SADDR_N[7:0]	WP3_SADDR complement
0x1fff f82A	Reserved	
0x1fff f82B	Reserved	
0x1fff f82C	WP3_EADDR [7:0]	WP3_EADDR contains the last page of Write protect area 3.
0x1fff f82D	WP3_EADDR_N[7:0]	WP3_EADDR complement
0x1fff f82E	Reserved	
0x1fff f82F	Reserved	
0x1fff f830	FMC_OTPDATA1_LK[7:0]	OTP USER_DATA1 block(7-0) lock
0x1fff f831	FMC_OTPDATA1_LK_N[7:0]	FMC_OTPDATA1_LK[7:0] complement
0x1fff f832	FMC_OTPDATA1_LK[15:8]	OTP USER_DATA1 block(15-8) lock
0x1fff f833	FMC_OTPDATA1_LK_N[15:8]	FMC_OTPDATA1_LK[15:8] complement
0x1fff f834	FMC_OTPDATA1_LK[23:16]	OTP USER_DATA1 block(23-16) lock
0x1fff f835	FMC_OTPDATA1_LK_N[23:16]	FMC_OTPDATA1_LK[23:16] complement
0x1fff f836	FMC_OTPDATA1_LK[31:24]	OTP USER_DATA1 block(31-24) lock
0x1fff f837	FMC_OTPDATA1_LK_N[31:24]	FMC_OTPDATA1_LK[31:24] complement

Address	Name	Description
0x1fff f838	FMC_OTPDATA1_LK[39:32]	OTP USER_DATA1 block(39-32) lock
0x1fff f839	FMC_OTPDATA1_LK_N[39:32]	FMC_OTPDATA1_LK[39:32] complement
0x1fff f83a	FMC_OTPDATA1_LK[47:40]	OTP USER_DATA1 block(47-40) lock
0x1fff f83b	FMC_OTPDATA1_LK_N[47:40]	FMC_OTPDATA1_LK[47:40] complement
0x1fff f83c	FMC_OTPDATA1_LK[55:48]	OTP USER_DATA1 block(55-48) lock
0x1fff f83d	FMC_OTPDATA1_LK_N[55:48]	FMC_OTPDATA1_LK[55:48] complement
0x1fff f83e	FMC_OTPDATA1_LK[63:56]	OTP USER_DATA1 block(63-56) lock
0x1fff f83f	FMC_OTPDATA1_LK_N[63:56]	FMC_OTPDATA1_LK[63:56] complement
0x1fff f840	FMC_OTPDATA2_LK[7:0]	OTP USER_DATA2 block(7-0) lock
0x1fff f841	FMC_OTPDATA2_LK_N[7:0]	FMC_OTPDATA2_LK[7:0] complement
0x1fff f842	FMC_OTPDATA2_LK[15:8]	OTP USER_DATA2 block(15-8) lock
0x1fff f843	FMC_OTPDATA2_LK_N[15:8]	FMC_OTPDATA2_LK[15:8] complement
0x1fff f844	FMC_OTPDATA2_LK[23:16]	OTP USER_DATA2 block(23-16) lock
0x1fff f845	FMC_OTPDATA2_LK_N[23:16]	FMC_OTPDATA2_LK[23:16] complement
0x1fff f846	FMC_OTPDATA2_LK[31:24]	OTP USER_DATA2 block(31-24) lock
0x1fff f847	FMC_OTPDATA2_LK_N[31:24]	FMC_OTPDATA2_LK[31:24] complement
0x1fff f848	FMC_OTPDATA2_LK[39:32]	OTP USER_DATA2 block(39-32) lock
0x1fff f849	FMC_OTPDATA2_LK_N[39:32]	FMC_OTPDATA2_LK[39:32] complement
0x1fff f84a	FMC_OTPDATA2_LK[47:40]	OTP USER_DATA2 block(47-40) lock
0x1fff f84b	FMC_OTPDATA2_LK_N[47:40]	FMC_OTPDATA2_LK[47:40] complement
0x1fff f84c	FMC_OTPDATA2_LK[55:48]	OTP USER_DATA2 block(55-48) lock

Address	Name	Description
0x1fff f84d	FMC_OTPDATA2_LK_N[55:48]	FMC_OTPDATA2_LK[55:48] complement
0x1fff f84e	FMC_OTPDATA2_LK[63:56]	OTP USER_DATA2 block(63-56) lock
0x1fff f84f	FMC_OTPDATA2_LK_N[63:56]	FMC_OTPDATA2_LK[63:56] complement

Option bytes modify

The following steps show the modify sequence.

- Unlock the FMC_OBCTL register.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals 0). Otherwise, wait until the operation has finished.
- Unlock the OBLK bit in the FMC_CTL register by writing the right sequence to the FMC_OBKEY register.
- Wait until the OBLK bit is cleared in the FMC_CTL register.
- Write the desired option bytes in the desired option byte registers.
- Set the OBSTART bit in the FMC_CTL register to send the option byte change command.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.

Note:

1. During the modification process, power-off or reset is prohibited; otherwise, it may cause option byte abnormalities, resulting in program failure to run. Code and option bytes need to be reprogrammed.

2.3.10. Erase/program protection(Write 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, then the WPERR bit in the FMC_STAT register will be set by the FMC. If any area is write-protected, a software mass erase operation cannot be performed. The WP area are defined by a start address offset and an end address offset. The page protection function can be individually enabled by configuring the WP address registers: FMC_WPx (x = 0,1,2,3).

Four write protection (WP) areas can be defined with a granularity of 2KB. The WP area is defined by:

[(WPx _EADDR - WPx _SADDR) + 1] x 2Kbytes

From base address + [(WPx_SADDR[7:0]) x 0x800](include) to the address base address + [(WPx _EADDR[7:0] + 1) x 0x800](exclude).

For example, to protect the address from 0x0800 2000 (included) to 0x0800 4FFF (included) the FMC_WPx register must be set as follows:

WPx_SADDR[7:0] = 0x04

WPx_EADDR[7:0] = 0x09

Table 2-7. WPx_SADDR and WPx_EADDR settings for protected pages

FMC_WPx registers value(x=0,1,2,3)	WP area
WPx_SADDR = WPx_EADDR	Page WPx_SADDR is protected.
WPx_SADDR > WPx_EADDR	No WP area.
WPx_SADDR < WPx_EADDR	The pages from WPx_SADDR to WPx_EADDR are protected.

2.3.11. Security protection

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

No protection: when setting the security protection (SPC) byte and its complement value to 0xA55A, no protection performed. The main flash and option bytes block are accessible by all operations.

Protection level low: When the SPC byte and its complementary byte are set to any value other than 0xA55A or 0xCC33, the system enters the low security protection state after reset. If the SPC modification is performed while a debugger remains connected to the JTAG/SWD interface, a power-on reset should be executed instead of a normal system reset. In the low protection level, the main Flash memory can only be accessed through user code. In debug mode, or when booting from SRAM or bootloader mode, any read operation to the main Flash memory block will generate a bus error. In these same modes, if a program or erase operation is attempted on the main Flash block through the FMC registers, the WPERR bit in the FMC_STAT register will be set. Programming Flash through DMA or CPU write operations will not succeed and will not set the WPERR flag. Instead, a bus error will be triggered - DMA will ignore this error, while the CPU-side software can detect it as a HardFault exception. However, in all of these modes, option bytes can still be accessed and modified, which allows disabling the security protection feature. By setting the SPC byte and its complementary value to 0xA55A, the system returns to the unprotected level, and an automatic mass erase operation will be triggered.

Protection level high: when setting SPC byte to 0xCC33, high protection level is enabled. 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.

2.3.12. Secure user area (SCR)

In the main flash block, FMC can define secure user areas which can be executed only once at boot, and never again unless a reset occurs.

Secure user areas can isolate secure code from application non-secure code. Secure user areas can be used to protect a custom secure boot library, firmware update code, or a third party secure library. When the secure user area is enabled, erase operations on the SCR area will set the WPERR bit. When the SCRREN bit corresponding to a data block is 0, the data block is protected against reads and execution. Reading this area will return 0 and trigger a bus error. When the SCRWLK bit corresponding to a data block is 0, performing a write operation on this data block will set the WPERR flag.

The size of the secure user area is defined by the SCR_BLOCK_CNT[4:0] bits of the OTP_SCRCTL. When switching SPC from a low protection level to an unprotected state, all contents except those configured as secure protection areas will be erased.

The secure user area is defined as follows:

- from base address (0x0800 0000) up to (but not including) base address + (SCR_BLOCK_CNT[4:0] * 0x2000). When SCR_BLOCK_CNT[4:0] is 0x10 to 0x1E, it is calculated as 16 blocks, with each block 8KB (0x2000); when it is 0x00 or 0x1F, there is no secure user area.

Table 2-8. SCR/WP Erase Protection

Erase Type	SCR_BLOCK_CNT[4:0]	Write Protection	Behavior/Action	WPERR
Page erase	0x00 or 0x1F(Non-secure User Area)	NO	Page erasable normally	Not set
	0x00 or 0x1F(Non-secure User Area)	YES	Write-protected page not erasable	Set
	0x01-0x1E	-	Protected page cannot be erased	Set
Mass erase	0x00 or 0x1F(Non-secure User Area)	NO	mass erase performed normally	Not set
	0x00 or 0x1F(Non-secure User Area)	YES	mass erase not allowed	Set
	0x01-0x1E	-	mass erase not allowed	Set
Protection level low or Protection level high	0x00 or 0x1F(Non-secure User Area)	NO	mass erase performed normally	Not set
	0x00 or 0x1F(Non-secure User Area)	YES	mass erase not allowed	Not set
	0x01-0x1E	-	Erase all pages except security-protected area	Not set

2.4. Register definition

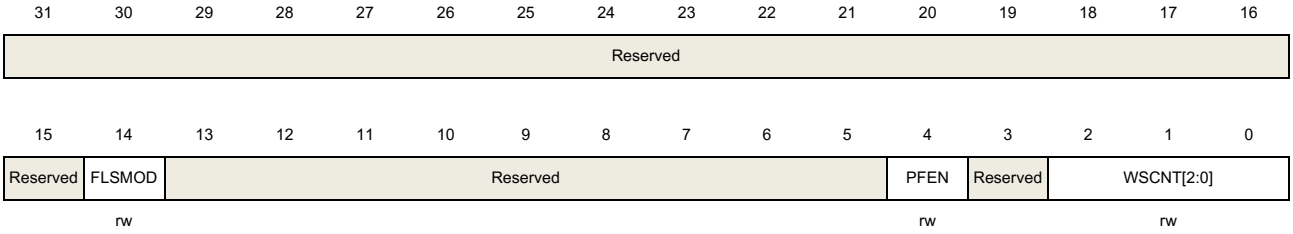
FMC base address: 0x4002 2000

2.4.1. Wait state register (FMC_WS)

Address offset: 0x00

Reset value: 0x0000 0010

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



Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	FLSMOD	Flash goes to sleep mode or power-down mode when MCU enters Deep-sleep mode. 0: Flash is in sleep mode. 1: Flash is in power-down mode.
13:5	Reserved	Must be kept at reset value.
4	PFEN	Pre-fetch enable 0: Pre-fetch disable 1: Pre-fetch enable
3	Reserved	Must be kept at reset value.
2:0	WSCNT[2:0]	Wait state counter These bits are set and reset by software. 000: 0 wait state added 001: 1 wait state added 010: 2 wait state added 011: 3 wait state added 100: 4 wait state added 101: 5 wait state added 110: 6 wait state added 111: 7 wait state added

2.4.2. Unlock key register (FMC_KEY)

Address offset: 0x04

Reset value: 0x0000 0000

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

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_CTL unlock register These bits can only be written by software. Write KEY[31:0] with keys to unlock FMC_CTL register.

2.4.3. 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]	FMC_OBCTL 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 the FMC_OBCTL register.

2.4.4. Status register (FMC_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

FLSPWR	Reserved
r	

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

Bits	Fields	Descriptions
31	FLSPWR	Flash power ready. 0: Flash power not ready 1: Flash power ready
30:9	Reserved	Must be kept at reset value.
8	OFERR	Offset error flag bit When an error occurs with OF_SPAGE in the SCR region, this bit is set by hardware. This bit can be cleared by writing 1. 0: No offset error occurs 1: Offset error occurs
7	Reserved	Must be kept at reset value.
6	OBERR	Option byte read error flag bit. This bit is set by hardware when the option byte and its complement byte do not match, and the option byte set 0xFF.
5	ENDF	End of operation flag bit When the operation is executed successfully, this bit is set by hardware. This bit can be cleared by writing 1.
4	WPERR	Erase / program protection error flag bit. When an erase/program protection error occurs, this bit is set by hardware. This bit can be cleared by writing 1. 0: No write protection error occurs 1: Write protection error occurs
3	PGAERR	Program alignment error flag bit This bit is set by hardware when write data is not aligned. The software can clear it by writing 1.
2	PGERR	Program error flag bit This bit is set by hardware when programming a flash location that is not erased (not all 1s). The software can clear it by writing 1.
1	PGSERR	Program sequence error flag bit. Set by hardware if a write operation to the Flash is performed while not setting PG previously. It is also set by hardware when PGERR, PGAERR or WPERR because of a previous programming error that occurred. This bit can be cleared by writing 1. 0: No program sequence error occurs

		1: Program sequence error occurs
0	BUSY	The flash is busy bit When the operation is in progress, this bit is set to 1. When the operation is end or an error is generated, this bit is cleared to 0.

2.4.5. Control register (FMC_CTL)

Address offset: 0x10

Reset value: 0xC000 0000

This register 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	OBLK	Reserved												OBSTART	START
rw	rw													rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFERRIE	Reserved	OBERRIE	ENDIE	Reserved	ERRIE	Reserved							MER	PER	PG
rw		rw	rw		rw								rw	rw	rw

Bits	Fields	Descriptions
31	LK	FMC_CTL lock bit This bit is cleared by hardware when right sequent written to FMC_KEY register. This bit can be set by software.
30	OBLK	FMC_OBCTL lock bit If this bit is set, all bits about user option in FMC_OBCTL register and so option page are locked. This bit is cleared by hardware when right sequence written to FMC_OBKEY register. This bit can be set by software.
29:18	Reserved	Must be kept at reset value
17	OBSTART	Send option byte change command to FMC bit. This bit is set by software to send option byte change command to FMC only when OBLK is set to 0. This bit is cleared by hardware when the BUSY bit is cleared
16	START	Send erase command to the FMC This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
15	OFERRIE	Offset error interrupt enable 0: No interrupt generated by hardware 1: Offset error interrupt enable
14	Reserved	Must be kept at reset value

13	OBERRIE	Option bytes error interrupt enable 0: No interrupt generated by hardware 1: Option bytes error interrupt enable
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
9:3	Reserved	Must be kept at reset value.
2	MER	Main flash mass erase command bit This bit is set or cleared by software. 0: No effect 1: Main flash mass erase command
1	PER	Main flash page erase command bit This bit is set or cleared by software. 0: No effect 1: Main flash page erase command
0	PG	Main flash program command bit This bit is set or cleared by software. 0: Disabled flash programming 1: Enabled flash programming

2.4.6. Address register (FMC_ADDR)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions

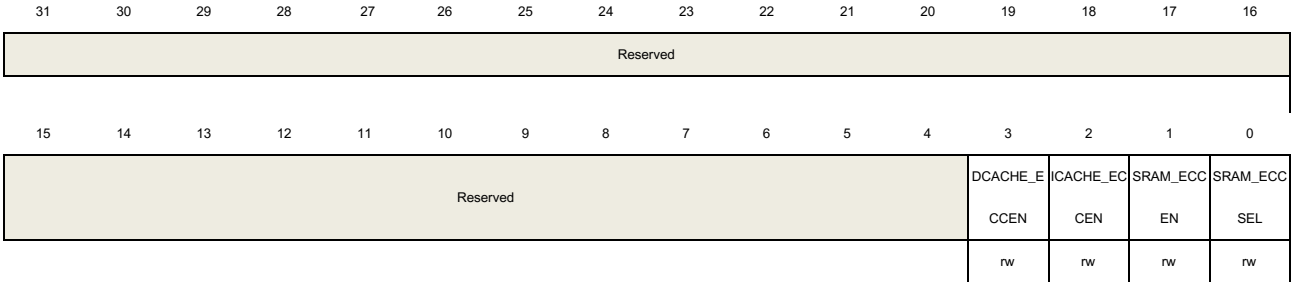
31:0	ADDR[31:0]	Flash erase/program command address bits These bits are configured by software. ADDR bits are the address of flash to be erased/programmed.

2.4.7. ECC control register(FMC_ECCCTL)

Address offset: 0x18

Reset value: 0x0000 0000

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



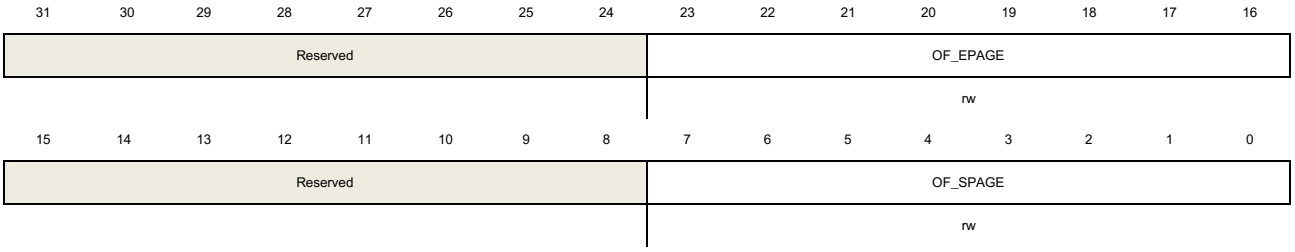
Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	DCACHE_ECCEN	DCACHE ECC enable 0: DCACHE ECC disable 1: DCACHE ECC enable When SRAM_ECCSEL=1, DCACHE_ECCEN controls the enabling of DCACHE ECC.
2	ICACHE_ECCEN	ICACHE ECC enable 0: ICACHE ECC disable 1: ICACHE ECC enable When SRAM_ECCSEL=1, ICACHE_ECCEN controls the enabling of ICACHE ECC.
1	SRAM_ECCEN	SRAM ECC enable 0: SRAM ECC disable 1: SRAM ECC enable When SRAM_ECCSEL=1, SRAM_ECCEN controls the enabling of SRAM ECC.
0	SRAM_ECCSEL	SRAM ECC enable control select 0: Option byte control 1: SRAM ECC is enabled through register bit

2.4.8. Offset region register (FMC_OFRG)

Address offset: 0x1C

Reset value: 0x0000 00FF

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



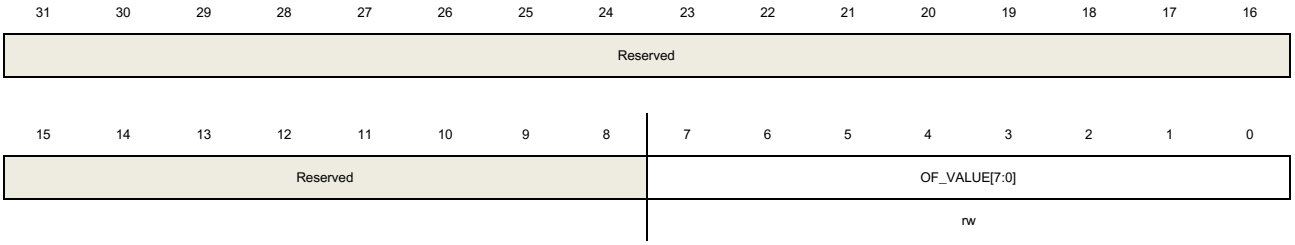
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	OF_EPAGE[7:0]	End page of offset region
15:8	Reserved	Must be kept at reset value
7:0	OF_SPAGE[7:0]	Start page of offset region

2.4.9. Offset value register (FMC_OFVR)

Address offset: 0x20

Reset value: 0x0000 0000

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



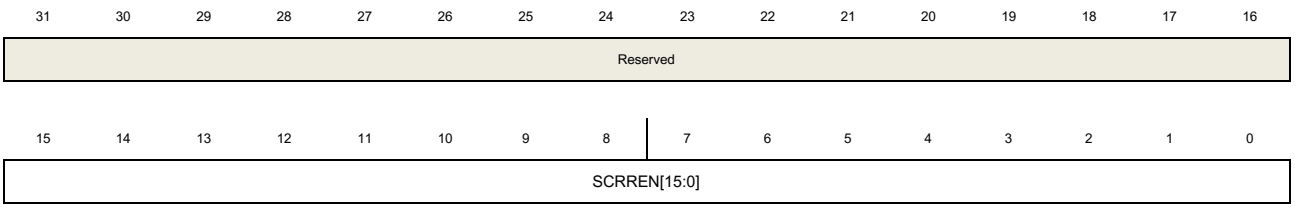
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OF_VALUE[7:0]	Page offset value.

2.4.10. SCR control register (FMC_SCRCTL)

Address offset: 0x24

Reset value: 0x0000 FFFF

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



rc_w0

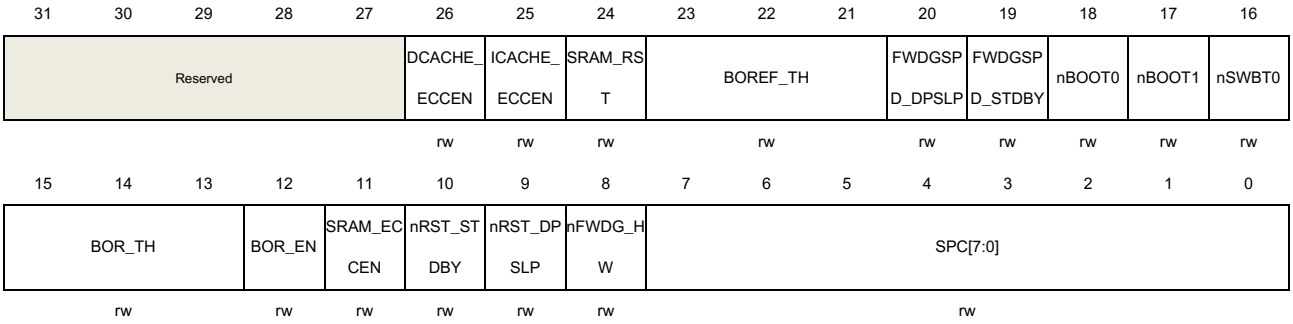
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SCRREN[15:0]	Enable SCR area read SCRREN[n] indicates SCR data block n can be read and execute, n=0..15. Default value is 1, software can write 0, and lock in this reset cycle. The number of valid bits in SCRREN is determined by the value of SCR_BLOCK_CNT. If the secure user area is not enabled, this bit is invalid. 0: data block is not readable 1: data block is readable

2.4.11. Option byte control register (FMC_OBCTL)

Address offset: 0x28

Reset value: 0x0XXX XXXX

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



Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	DCACHE_ECCEN	enable DCACHE ECC 0: disable DCACHE ECC 1: enable DCACHE ECC
25	ICACHE_ECCEN	enable ICACHE ECC 0: disable ICACHE ECC 1: enable ICACHE ECC
24	SRAM_RST	sram initialize 0: Initialize SRAM after power reset 1: Do not initialize SRAM after power reset **Note:** When the Option Bytes reload is manually triggered by setting the OBSTART bit, the chip generates a system reset. If this reload changes the SRAM_RST bit in the Option Bytes from "1" before loading to "0" after loading, the SRAM will perform

		a hardware initialization after the system reset is completed.
23:21	BOREF_TH	BOR eflash threshold 000: 1.61V rising threshold, 1.57V falling threshold 001: 1.66V rising threshold, 1.62V falling threshold 010: 1.71V rising threshold, 1.67V falling threshold 011: 1.76V rising threshold, 1.72V falling threshold 100: 1.81V rising threshold, 1.77V falling threshold 101~111: 1.86V rising threshold, 1.82V falling threshold
20	FWDGSPD_DPSLP	FWDGT suspend option in Deep-sleep mode configuration 0: FWDGT is suspended in system Deep-sleep mode 1: FWDGT is running in system Deep-sleep mode.
19	FWDGSPD_STDBY	FWDGT suspend option in standby mode configuration 0: FWDGT is suspend in system standby mode 1: FWDGT is running in system standby mode
18	nBOOT0	Boot0 configuration bit 0: BOOT0 is 1 1: BOOT0 is 0
17	nBOOT1	Boot1 configuration bit 0: BOOT1 is 1 1: BOOT1 is 0 Together with the BOOT0 pin, this bit selects boot mode
16	nSWBT0	Software BOOT0 disable 0: BOOT0 depends on the option bit nBOOT0 1: BOOT0 depends on PB11/BOOT0 pin
15:13	BOR_TH	BOR_TH Level Selection 000: 3.0V rising threshold, 2.9V falling threshold 001: 2.9V rising threshold, 2.8V falling threshold 010: 2.7V rising threshold, 2.6V falling threshold 011: 2.6V rising threshold, 2.5V falling threshold 100: 2.5V rising threshold, 2.4V falling threshold 101: 2.3V rising threshold, 2.2V falling threshold 110: 2.1V rising threshold, 2.0V falling threshold 111: 1.85V rising threshold, 1.75V falling threshold
12	BOR_EN	BOR_EN 0: BOR enable 1:BOR disable
11	SRAM_ECCEN	SRAM ECC enable 0: SRAM ECC disable

		1: SRAM ECC enable
10	nRST_STDBY	Option byte standby reset value 0: Generates a reset if entering standby mode 1: No reset if entering standby mode
9	nRST_DPSLP	Option byte Deep-sleep reset value 0: Generates a reset if entering Deep-sleep mode 1: No reset if entering Deep-sleep mode
8	nFWDG_HW	Free watchdog configuration bit 0: Hardware free watchdog 1: Software free watchdog
7:0	SPC[7:0]	Store SPC of option bytes block after system reset.

2.4.12. SCR write lock register (FMC_SCRWLK)

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
SCRWLK[15:0]															
rc_w0															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	SCRWLK[15:0]	Enable SCR write lock SCRWLK[n] indicates SCR data block n is write lock, n=0..15 0: data block lock 1: data block not locked When one block is locked,it is not possible to remove the write lock. The number of valid bits in SCRWLK is determined by the value of SCR_BLOCK_CNT. If the secure user area is not enabled, this bit is invalid.

2.4.13. Erase/Program protection register (FMC_WPx) (x = 0…3)

Address offset: 0x30 + 0x04 * x

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

Reserved	WPx_EADDR[7:0]
	rw

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23:16	WPx_EADDR[7:0]	WP area end page
15:8	Reserved	Must be kept at reset value
7:0	WPx_SADDR[7:0]	WP area start page

2.4.14. OTP DATA1 low block lock register (FMC_OTPDATA1_LLK)

Address offset: 0x44

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
OTPDATA1_LWLK[31:16]															
rc_w0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA1_LWLK[15:0]															
rc_w0															

Bits	Fields	Descriptions
31:0	OTPDATA1_LWLK[31:0]	OTP USER_DATA1 block write lock Block n corresponds to OTP USER_DATA1 128-bits * n LWLK[n] = 0: indicates that all OTP USER_DATA1 128-bits in OTP Block n are locked and attempt to program them results in WPERR. LWLK[n] = 1: indicates that all OTP USER_DATA1 128-bits in OTP Block n are not locked. When one block is locked, it is not possible to remove the write protection.

2.4.15. OTP DATA1 high block lock register (FMC_OTPDATA1_HLK)

Address offset: 0x48

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
OTPDATA1_HWLK[63:48]															

rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA1_HWLK[47:32]															

rc_w0

Bits	Fields	Descriptions
31:0	OTPDATA1_HWLK[63:32]	OTP USER_DATA1 block write lock Block n corresponds to OTP USER_DATA1 128-bits * n HWLK[n] = 0: indicates that all OTP USER_DATA1 128-bits in OTP Block n are locked and attempt to program them results in WPERR HWLK[n] = 1: indicates that all OTP USER_DATA1 128-bits in OTP Block n are not locked. When one block is locked, it is not possible to remove the write protection.

2.4.16. OTP DATA2 low block lock register (FMC_OTPDATA2_LLK)

Address offset: 0x4C

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
OTPDATA2_LWLK[31:16]															

rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_LWLK[15:0]															

rc_w0

Bits	Fields	Descriptions
31:0	OTPDATA2_LWLK[31:0]	OTP USER_DATA2 block write lock Block n corresponds to OTP USER_DATA2 128-bits * n LWLK[n] = 0: indicates that all OTP USER_DATA2 128-bits in OTP Block n are locked and attempt to program them results in WPERR. LWLK[n] = 1: indicates that all OTP USER_DATA2 128-bits in OTP Block n are not locked. When one block is locked, it is not possible to remove the write protection.

2.4.17. OTP DATA2 high block lock register (FMC_OTPDATA2_HLK)

Address offset: 0x50

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
OTPDATA2_HWLK[63:48]															

rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_HWLK[47:32]															

rc_w0

Bits	Fields	Descriptions
31:0	OTPDATA2_HWLK[63:32]	OTP USER_DATA2 block write lock Block n corresponds to OTP USER_DATA2 128-bits * n HWLK[n] = 0: indicates that all OTP USER_DATA2 128-bits in OTP Block n are locked and attempt to program them results in WPERR. HWLK[n] = 1: indicates that all OTP USER_DATA2 128-bits in OTP Block n are not locked. When one block is locked, it is not possible to remove the write protection.

2.4.18. OTP DATA2 low read control register (FMC_OTPDATA2_LR_CTL)

Address offset: 0x54

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
OTPDATA2_LREN[31:16]															

rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_LREN[15:0]															

rc_w0

Bits	Fields	Descriptions
31:0	OTPDATA2_LREN[31:0]	Enable OTP USER_DATA2 block read Block n corresponds to OTP USER_DATA2 128-bits * n LREN[n] = 1: indicates that all OTP USER_DATA2 128-bits in OTP Block n can be read. LREN[n] = 0: indicates that all OTP USER_DATA2 128-bits in OTP Block n cannot be read. Software can write 0, and cannot modify in this reset cycle. Hardware set to 1 after reset.

2.4.19. OTP DATA2 high read control register (FMC_OTPDATA2_HR_CTL)

Address offset: 0x58

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

OTPDATA2_HREN[63:48]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTPDATA2_HREN[47:32]															
rc_w0															

Bits	Fields	Descriptions
31:0	OTPDATA2_HREN[63:32]	Enable OTP USER_DATA2 block read Block n corresponds to OTP USER_DATA2 128-bits * n HREN[n] = 1: indicates that all OTP USER_DATA2 128-bits in OTP Block n can be read. HREN[n] = 0: indicates that all OTP USER_DATA2 128-bits in OTP Block n cannot be read. Software can write 0, and cannot modify in this reset cycle. Hardware set to 1 after reset.

2.4.20. ECC control and status register (FMC_ECCCS)

Address offset: 0x60

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCINJEN	Reserved					ECCINJDATA[1:0]		Reserved							
rw						rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCORIE	Reserved													ECCDET	ECCCOR
rw														rc_w1	rc_w1

Bits	Fields	Descriptions
31	ECCINJEN	ECC fault injection enable 0: ECC fault injection disable 1: ECC fault injection enable
30:26	Reserved	Must be kept at reset value.
25:24	ECCINJDATA[1:0]	When ECCINJEN is set to 1, these bits' field can be used to control the injection of one bit or two bits error fault into the flash memory. 00: No error fault injection 01: One bit error fault injection(bit0) 10: One bit error fault injection(bit1) 11: Two bits error fault injection(bit0 and bit1)

		Note: These bit fields can only be used to inject errors into bit0 and bit1 of flash memory data.
23:16	Reserved	Must be kept at reset value.
15	ECCCORIE	One bit error correct interrupt enable. 0: One bit error correct interrupt disable. 1: One bit error correct interrupt enable.
14:2	Reserved	Must be kept at reset value.
1	ECCDET	Two-bit errors detect flag. This bit set when two bit errors is detected. This bit is cleared by writing 1. 0: Two bits errors are not detected. 1: Two bits errors are detected.
0	ECCCOR	One-bit error detected and correct flag. This bit is cleared by writing 1. 0: No bit error is detected and corrected. 1: One bit error is detected and corrected.

2.4.21. ECC address register(FMC_ECCADDR)

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
ECCEAR[31:16]															
rc_w0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCEAR[15:0]															
rc_w0															

Bits	Fields	Descriptions
31:0	ECCEAR[31:0]	ECC error logic address These bits are cleared by writing 0. If the read address offset function is correctly enabled, the ECCEAR address will be the destination address after the offset.

2.4.22. Product ID register 0 (FMC_PID0)

Address offset: 0x100
Reset value: 0xXXXX XXXX

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PID0[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID0[15:0]															
r															

Bits	Fields	Descriptions
31:0	PID0[31:0]	Product reserved ID code 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.

2.4.23. Product ID register 1 (FMC_PID1)

Address offset: 0x104

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
PID1[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID1[15:0]															
r															

Bits	Fields	Descriptions
31:0	PID1[31:0]	Product reserved ID code These bits are read only by software. These bits are unchanged constant after power on. These bits are one time program when the chip produced.

3. Instruction cache (ICACHE)

3.1. Introduction

The instruction cache (ICACHE) is based on CBUS code bus of Cortex®-M33 processor. It is necessary to improve the performance in fetching instruction and data from both internal and external memories.

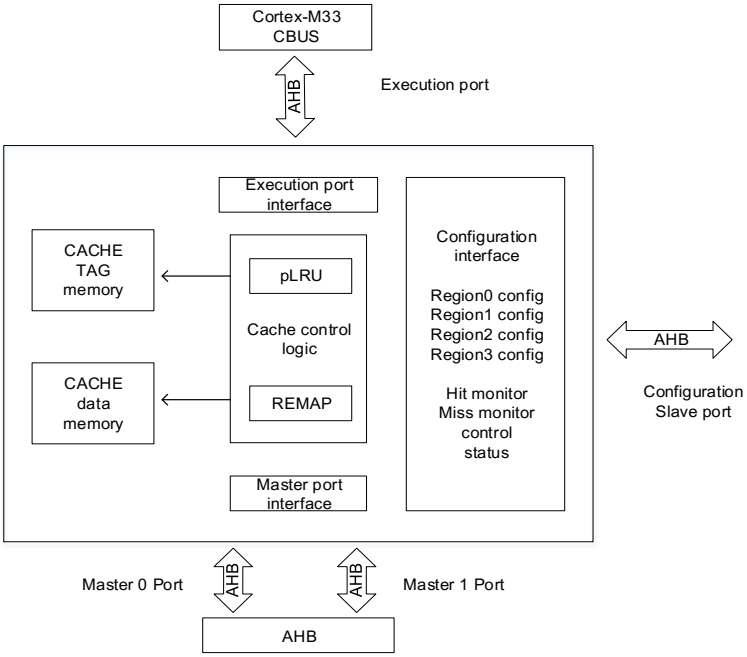
3.2. Characteristics

- Support 8KB cache with 2 ways, 256 cache lines per way and 16B per cache line
- Support fetch address without any wait state if cache hit
- ICACHE interface support two 32-bit slave ports and one 32-bit master port and one 128-bit master port. One slave port is execution port, the other one is for registers access. Master 0 port outputs to fast bus, while master 1 port outputs to slow bus.
- Cache access support hit-under-miss and paired master access feature
- Support memory address remap
- Support pLRU replacement policy and critical-word-first refill policy.
- Support remap region AHB transaction burst type configuration.
- Support two performance counters: 32-bit hit monitor counter and16-bit miss monitor counter.
- Operation management: Cache invalidate and optional interrupt handle.

3.3. Function overview

The purpose of ICACHE is to cache fetched instruction and data, only for read transaction, not for write transaction. Considering error management, in case of detecting an unexpected cacheable write access, set an error flag and optionally trigger an interrupt.

Figure 3-1. ICACHE block diagram



3.3.1. ICACHE initialization

Setting the EN bit of ICACHE_CTL enables the ICACHE function, within the same clock period. While ICACHE is disabled, ICACHE is ignored. ICACHE default state is disabled at boot.

On releasing the reset signal, the cache invalidate operation is automatically started, each TAG is cleared to 0, INVAL bit is automatically cleared, and ICACHE_STAT BUSY flag will be set. When cache invalidate operation is complete, all cache line valid bits will be cleared, and BUSY flag is reset, while END flag is set.

To ensure performance, it is necessary to check if cache invalidate operation completed before enabling the ICACHE. Application must check BUSY and END flag in ICACHE_STAT before enabling the ICACHE. In case that ICACHE is enabled before invalidate operation ended, during this period, on the condition that BUSY flag is set, any cache access is uncacheable and its accessing performance depends on the main memory.

3.3.2. Paired master cache

There is a paired AHB master ports: master0 and master1 port, which support ICACHE to classify the address access to different destination memories. Master0 port (fast bus) accesses internal flash and internal SRAM, and master1 port (slow bus) accesses external flash through the OSPI interface. master0 port is selected for mapped access to internal memories, master 1 is selected for remapped traffic to external memories.

Paired master feature ensures that the processor has a backup method to fetch from

different memory. It is feasible to separate the traffic to internal flash and the traffic to internal SRAM (if it is remapped), so as to decrease the processor stalls when ICACHE is not on hit.

3.3.3. ICACHE TAG memory

The ICACHE TAG memory includes valid bits and address tags that indicate which data are contained in the cache data memory

For each cache line，there is one corresponding valid bit, which is set while the cache line is refilled. Valid bit is reset in several cases, for example, reset ICACHE, disable cache and execute ICACHE invalidate operation.

When input execution port receive cacheable transaction, its AHB address (AHB_ADDR_in, 32bit) is composed of Bits [3:0], Bits [11:4] and Bits [31:12]. Bits [3:0] is the address byte offset, Bits [11:4] is the address cache line index and Bits [31:12] is the tag address. The AHB address indicates the byte offset within a cache line and the cache line index within a way. Furthermore, it is necessary to check whether the requested data is valid in the cache.

ICACHE main parameters of TAG memory for default 2-way set are shown in ***Table 3-1. TAG memory parameters.***

Table 3-1. TAG memory parameters

Parameter	Value
Cache size	8KB
Cache ways number	2
Cache line size	128-bit
Cache lines number	256 per way
Address byte offset size	4-bit
Address way index size	8-bit
TAG address size	20-bit

3.3.4. Address remapping

For external memory regions, it is useful to define an aliased address on which the address remapping transposed to the destination external physical address. The principle of the remapping mechanism is for mapping between input AHB address and specified code region base address.

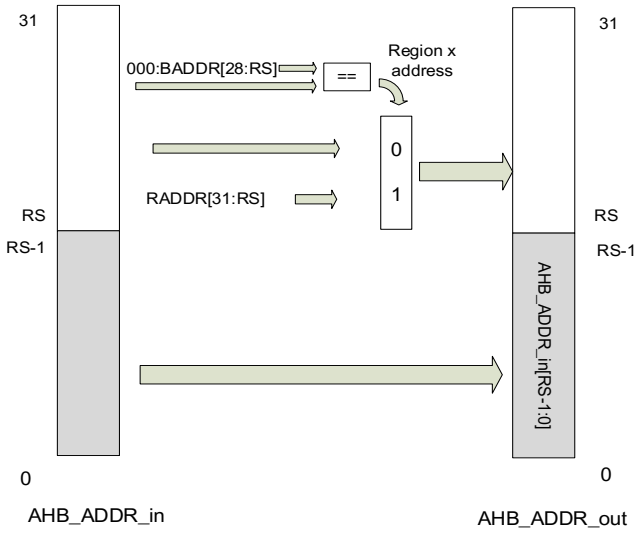
Note: Write operations are not supported for the aliased addresses after remapping.

If AHB_ADDR_in[31:RS] equals 000:BADDR[28:RS], AHB_ADDR_in belongs to region x. BADDR is the code sub-region base address, which is defined in the BADDR field of ICACHE_CFGx. RS defines the number of available bits, it is equal to log2(region size), the minimum is 21, corresponding region size is 2MB, the maximum is 27, corresponding

region size is 128MB.

In case that region x is enabled, the master port output AHB address (AHB_ADDR_out) which consist of MSBs (RADDR[31:RS] field of ICACHE_CFGx) and LSBs (AHB_ADDR_in[RS-1:0]).

Figure 3-2. ICACHE remapping address



When programming BADDR and RADDR field in ICACHE_CFGx, it is notable that if programmed value is larger than expected (MSBs), the unnecessary extra LSBs are ignored.

Table 3-2. ICACHE remap region size

Region size(MB)	Base address(MSBs)	Remap address(MSBs)
2	8	11
4	7	10
8	6	9
16	5	8
32	4	7
64	3	6
128	2	5

The remapping operation is configured by programming the ICACHE_CFGx register, the programming operation is merely done in case that ICACHE is disabled. Once the EN bit in ICACHE_CFGx register is set, even if ICACHE is disabled, or the transaction is uncacheable, the corresponding region x is enabled and then remap operation is processed. The SIZE bit in ICACHE_CFGx register is used to configure remap region size. The size of region is a power of two multiple in 2 Mbytes, the minimum region size is 2 Mbytes and the maximum region size is 128 Mbytes.

Through configuring the OBT bit in ICACHE_CFGx, the type of AHB burst is configured as INCR4.

INCR4 is used for external memories accessed through OSPI interface. INCR4 burst size is equal to cache line size. INCR4 burst start address is equal to cache line boundary aligned address.

3.3.5. Cacheable and uncacheable access

ICACHE also supports a memory region remapping feature which supports making external memory regions cacheable. ICACHE is able to access up to three external memory regions in cacheable way. There is an alias of external memory region address in the code region (address range [0x0000 0000:0x07FF FFFF] or [0x1000 0000:0x1FFF FFFF]), so ICACHE can manage and the CBUS bus is able to route an external memory region (physical address range [0x6000 0000, 0x6FFF FFFF], [0x7000 0000, 0x7FFF FFFF], [0xB000 0000, 0xBFFF FFFF]) through their code alias address.

Depending on memory request AHB transaction memory lookup attribute, which is shown in *Table 3-3. ICACHE cache ability for transaction*, the request to ICACHE is defined as cacheable or uncacheable.

Table 3-3. ICACHE cache ability for transaction

Lookup attribute	Cache ability
1	Cacheable
0	Uncacheable

If there is a memory request for external memory in the code region, ICACHE performs the address remapping first. If the address is required to be aliased, the remapping is performed first, then it is cached. The target physical address does not need further operation. The remapping functionality remains available even if the cache is disabled and traffic is uncacheable.

If there is a memory uncacheable access, ICACHE is ignored, so that the AHB transaction is transmitted directly unchanged to the master output port, while, depending on remapping feature, only transaction address may be changed. The ignored operation has no effect on accessing the target memory.

User is able to configure Cacheable memory regions in the memory protection unit (MPU), which is responsible for the AHB attribute signal generating, the attribute is served for any transaction addressing a given region.

Table 3-4. Memory configuration

Memory	Cacheable (MPU Programming)	Remap (ICACHE_CFGx Programming)
Flash	Yes or No	Not support
SRAM	Not recommended	Not support
External memory (OSPI interface)	Yes or No	Required

If ICACHE receive a cacheable transaction request, there are two possible scenarios,

cache hit and cache miss.

Cache hit: If address is present in its TAG memory and the corresponding cache line is valid, means cache hit, reading from cache and providing to core are in the same period.

However, according to hit-under-miss feature, if address is not present in TAG memory which belongs to the current refill burst, also means cache hit. Even if cache line is refilling, only if the data is available at its master interface, ICACHE is able to fetch the requested data quickly, so as to avoid a miss.

Cache miss: If address is not present in its TAG memory, means cache miss, the data is read from main memory, and refill the cache line. The critical-word-first policy optimize wait periods for the processor, the policy allows the processor to read the requested data first, then the cache line refilling is performed. If an address remap happens, the kind of burst, which is generated on master port, is configured by the OBT bit in its ICACHE_CFGx register.

Considering cache refilling, ICACHE determines the cache line index. In 2-way set associative mode, the line is pointed by the address index in each way, according to pLRU replacement algorithm, the way is selected for use, and the other one is for the next refill. In the 2-way set associative cache mode, if the cache line is selected to be written by the refill data, and the line is already valid, the targeted cache line must be invalidated first.

Comparing to fetching instructions from main memory, processor get lower power consumption in fetching instructions from internal ICACHE. If the cached memory is external, power saving is more obvious.

3.3.6. ICACHE performance monitoring

For analyzing cache performance, ICACHE support two monitors: a 32-bit hit monitor and a 16-bit miss monitor, they are disabled by default

Hit monitor counts the AHB-transactions on ICACHE input port, which do not generate a transaction on output master0 or master1 port. It is necessary to consider the access whose address is present in the TAG memory or in the refill buffer.

Miss monitor counts the AHB-transactions on ICACHE input port, which generate a transaction on output master0 or master1 port. It is necessary to consider the access whose address is not present in the TAG memory and the refill buffer.

Hit and miss monitors can be enabled and reset by software as following:

- Enable / disable the hit monitor through the HMEN bit in ICACHE_CTL.
- Reset the hit monitor by setting the HMRST bit in ICACHE_CTL.
- Enable / disable the miss monitor through the MMEN bit in ICACHE_CTL.
- Reset the miss monitor by setting the MMRST bit in ICACHE_CTL.

3.3.7. ICACHE interrupts

If there is an unsupported cacheable write request detected, an error is generated by setting ERR bit in ICACHE_STAT. If the corresponding interrupt enable bit is set, error interrupt is triggered.

If a cache invalidation operation is finished, an end flag is generated by setting END bit in ICACHE_STAT. If corresponding interrupt enable bit is set, end interrupt is triggered, and then cache is available again.

Table 3-5. ICACHE interrupt

ICACHE	Error	End
Interrupt event	Functional error	Operation end
Event flag	ERR	END
Interrupt enable bit	ERRIE	ENDIE
Interrupt clear bit	ERRC	ENDC

3.4. Register definition

ICACHE base address: 0x4002 6000

3.4.1. Control register (ICACHE_CTL)

Address offset: 0x00

Reset value: 0x0000 000C

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												MMRST	HMRST	MMEN	HMEN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												BSTT	AMSEL	INVAL	EN
												rw	rw	w	rw

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19	MMRST	Miss monitor reset 0: No effect 1: Reset cache miss monitor
18	HMRST	Hit monitor reset 0: No effect 1: Reset cache hit monitor
17	MMEN	Miss monitor enable 0: Switch off cache miss monitor, stopping the monitor and does not reset. 1: Cache miss monitor enabled
16	HMEN	Hit monitor enable 0: Switch off cache hit monitor, stopping the monitor and does not reset. 1: Cache hit monitor enabled
15:4	Reserved	Must be kept at reset value
3	BSTT	Burst type for fast bus 0: WRAP4 1: INCR4
2	AMSEL	Cache set-associativity mode selection Configure set-associativity mode when cache is disabled, software write. 0: No effect 1: 2-way set associative cache (reset value)
1	INVAL	Cache invalidation, set by software and cleared by hardware (BUSY flag is set)

		0: No effect 1: invalidate entire cache
0	EN	Enable 0: Cache disabled 1: Cache enabled

3.4.2. Status register (ICACHE_STAT)

Address offset: 0x04

Reset value: 0x0000 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															

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

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ERR	Cache error flag 0: No error 1: Error occurred during the operation
1	END	Operation end flag 0: Cache busy 1: Invalidate INVAL operation ended
0	BUSY	Busy flag 0: Cache is not executing an invalidate operation 1: Cache is executing a invalidate operation

3.4.3. Interrupt enable register (ICACHE_INTEN)

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													ERRIE	ENDIE	Reserved
													rw	rw	

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ERRIE	Cache error interrupt enable 0: Disable error interrupt 1: Enable error interrupt
1	ENDIE	Cache operation end interrupt enable 0: Disable operation end interrupt 1: Enable operation end interrupt
0	Reserved	Must be kept at reset value

3.4.4. Flag clear register (ICACHE_FC)

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													ERRC	ENDC	Reserved
													w	w	

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2	ERRC	Software clear cache error flag 0: No effect 1: Clears ERR flag in ICACHE_STAT
1	ENDC	Software clear operation end flag 0: No effect 1: Clears END flag in ICACHE_STAT.
0	Reserved	Must be kept at reset value

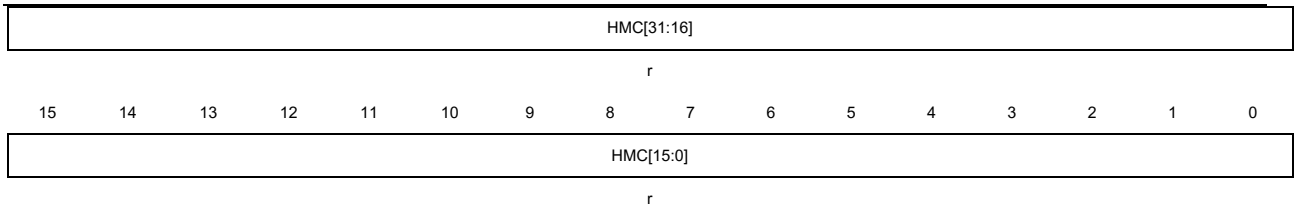
3.4.5. Hit monitor counter register (ICACHE_HMC)

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



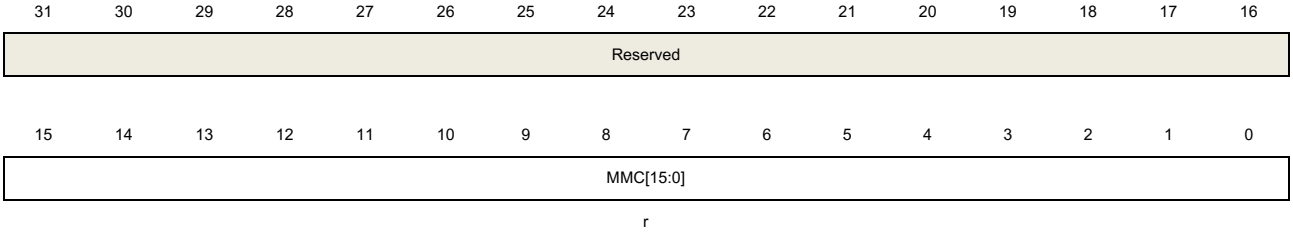
Bits	Fields	Descriptions
31:0	HMC[31:0]	Cache hit monitor counter.

3.4.6. Miss monitor counter register (ICACHE_MMC)

Address offset: 0x14

Reset value: 0x0000 0000

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



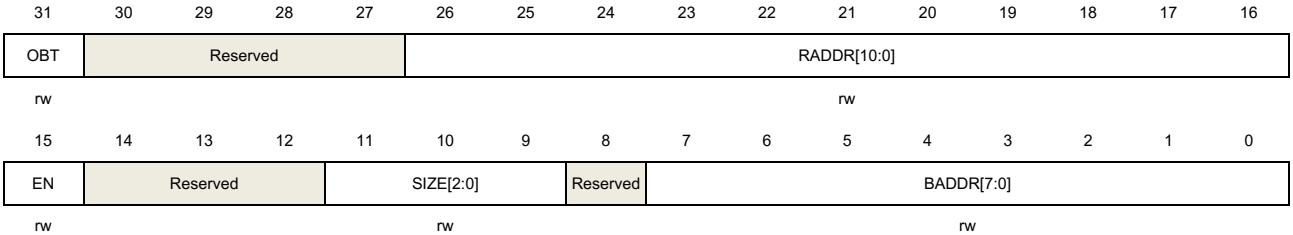
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	MMC[15:0]	Cache miss monitor counter

3.4.7. Configuration register (ICACHE_CFGx) (x = 0...3)

Address offset: 0x20 + 0x4 * x

Reset value: 0x0000 0200

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



Bits	Fields	Descriptions
31	OBT	Output burst type for region x 0: WRAP4 1: INCR4

30:27	Reserved	Must be kept at reset value
26:16	RADDR[10:0]	Region x remapped address, this field replaces the alias address defined by BADDR field.
15	EN	Region x enable 0: Disabled 1: Enabled
14:12	Reserved	Must be kept at reset value
11:9	SIZE[2:0]	Region x size 000: Reserved 001: 2 Mbytes 010: 4 Mbytes 011: 8 Mbytes 100: 16 Mbytes 101: 32 Mbytes 110: 64 Mbytes 111: 128 Mbytes
8	Reserved	Must be kept at reset value
7:0	BADDR[7:0]	Region x base address

4. Data cache (DCACHE)

4.1. Introduction

Data Cache (DCACHE), as an essential component of the Cortex®-M33 processor's S-AHB system bus, aims to enhance the overall performance of external memory data transfers.

DCACHE employs multiple advanced technologies, including hit/miss parallel processing mechanisms and a critical-word-first fill strategy, effectively improving the data access efficiency of external memory.

4.2. Characteristics

- Support 4KB cache with 2 ways, 128 cache lines per way and 16B per cache line
- Support fetch address without any wait state if cache hit
- DCACHE interface support one 32-bit slave ports and one 32-bit master port and one 32-bit slave port. One slave port is execution port, the other one is for registers access. Master port outputs to bus.
- Cache access support hit-under-miss and paired master access feature
- Supprot forceable Write-Through policy
- Supports byte, half-word and word writes
- Support Pseudo-random replacement policy and critical-word-first refill policy.
- Error management and optional interrupt handle.
- Maintenance operations: Cache invalidate, Cache invalidate range, Cache clean range, Cache clean and invalidate range
- Support four performance counters: two hit monitor counters(32-bit), Count read hit and write hit events and two miss monitor counters(16-bit), Count read miss and write miss events.

4.3. Function overview

The Data Cache (DCACHE) serves as an important high-speed buffer layer between the processor and external memory. Its primary function is to cache external memory access data generated by the processor during program execution, including both read and write operations. When the processor needs to access external memory, the DCACHE first checks whether the target data already exists in the cache, thereby avoiding frequent accesses to the slower external memory and improving overall computational efficiency. In addition, the DCACHE can also manage instruction fetch requests that occur in the external address space to further optimize system performance. The DCACHE module is responsible for coordinating and managing all read and write transactions, ensuring data consistency and efficient access.

Figure 4-1. DCACHE block diagram

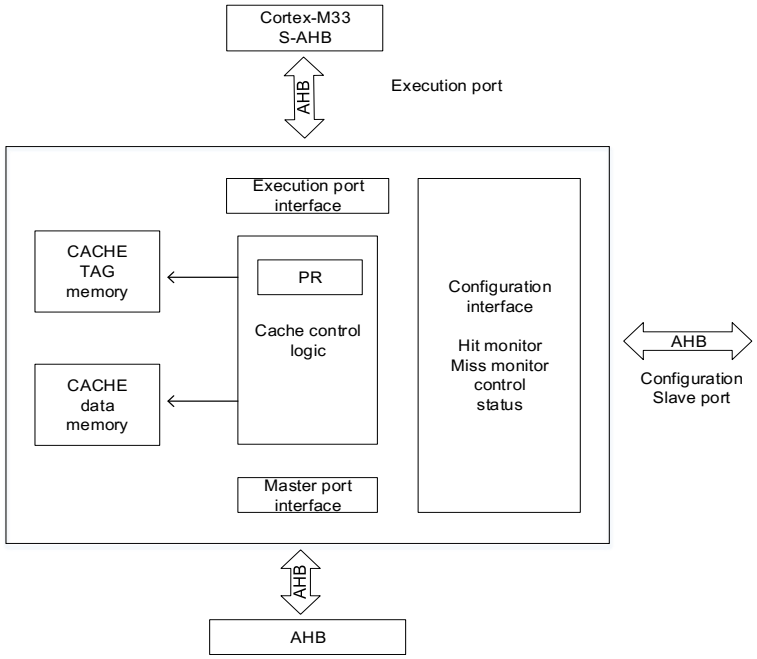
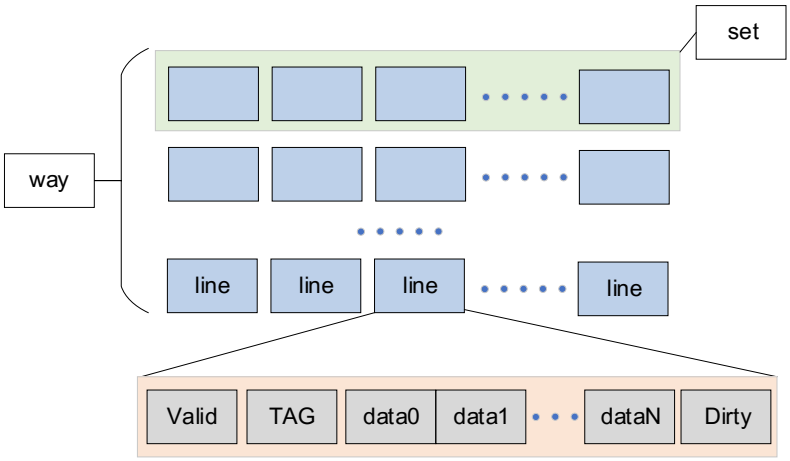


Figure 4-2. DCACHE block diagram



4.3.1. DCACHE initialization

Setting the DCACHEEN bit in the DCACHE_CTL register enables the DCACHE function, which takes effect within the same clock cycle. Disabling the DCACHE means that the cache is bypassed, and the DCACHE is disabled by default upon startup. After the reset signal is released, a cache invalidate operation is automatically initiated. At this time, the valid bit of each cache tag (TAG) is cleared to 0, and the MOBUSYF flag in the DCACHE_STAT register is set. When the cache invalidate operation completes, all valid and dirty bits of the cache lines are cleared, the MOBUSYF flag is reset, and the MOENDF flag is set to indicate that the operation has finished.

To ensure performance, it is necessary to check if cache invalidate operation completed before enabling the DCACHE. Application necessarily check MOBUSYF and MOENDF flag

in DCACHE_STAT before enabling the DCACHE. In case that DCACHE is enabled before invalidate operation ended, during this period, on the condition that MOBUSYF flag is set, any cache access is uncacheable actually and its accessing performance depends on the main memory.

4.3.2. DCACHE TAG memory

The DCACHE TAG memor includes dirty bits,valid bits and address tags bit that indicate which data are contained in the cache data memory.

Figure 4-3. cache line



TAG bits: Store the addresses that need to be matched

Valid bit: For each cache line， there is one corresponding valid bit, which is set while the cache line is refilled. Valid bit is reset in several cases, for example, reset DCACHE, disable Dcache and execute DCACHE invalidate operation.

Dirty bit：Dirty bit indicates whether the data in the cache is consistent with the data in the main memory, to determine whether the main memory needs to be updated. The core condition for resetting the dirty bit is that the cached data has been synchronized with the main memory or the cache line has been discarded.

In the DCACHE, the dirty bit is reset under the following circumstances:

1. Cache line clean operation (Clean)

When a Clean instruction is executed, dirty data is written back to main memory.

After the write-back is completed, the dirty bit is cleared (set to 0), indicating that the cache and main memory are synchronized.

2. Cache line invalidation (Invalidate)

When an Invalidate instruction is executed, the cache line is marked as invalid.

The dirty bit is also cleared, but the dirty data is lost (not written back to main memory).

3. Cache line clean and invalidate (Clean & Invalidate)

The dirty data is first written back to main memory (Clean),

then the cache line is invalidated, and the dirty bit is cleared.

4. Cache line replacement (Eviction)

When a cache miss occurs, a cache line needs to be replaced.

If the dirty bit of the line to be replaced is set to 1, a write-back operation is performed first.

After the write-back is completed, the dirty bit is cleared, and the new data is loaded into the cache line.

5. System reset or cache disable

During a hardware reset, all dirty bits are cleared.

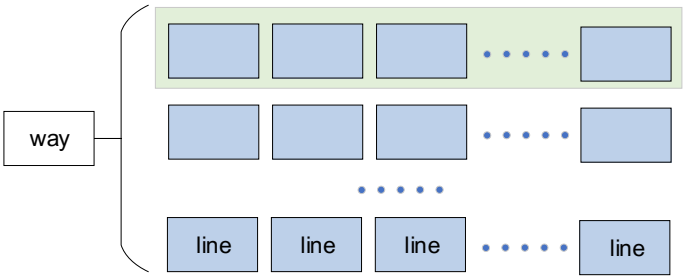
When the DCACHE is disabled, all dirty bits are reset.

6. Write-through mode

In write-through mode, write operations update both the cache and the main memory simultaneously.

The dirty bit remains 0 at all times, and no reset operation is required.

Figure 4-4. cache way



DCACHE main parameters of TAG memory for default 2-way set are shown in ***Table 4-1. TAG memory parameters.***

Table 4-1. TAG memory parameters

Parameter	Value
Cache size	4KB
Cache ways number	2
Cache line size	128-bit
Cache lines number	128 per way
Address byte offset size	4-bit
Address way index size	7-bit
TAG address size	21-bit

4.3.3. Cacheable and uncacheable access

According to the memory lookup attributes of AHB transactions initiated by memory requests, DCache write transactions can be categorized into Write Through and Write Back.In the Write Through mode, all write operations to the DCache are simultaneously performed in the next-level memory.In the Write Back mode, when data is written into the DCache, the corresponding cache line is marked as Dirty, and the data is only written back to the next-level memory when that cache line is replaced.When necessary, the data cache can be forced to operate in the Write-Through mode. Once forced write-through is enabled, the cache will

always use the write-through policy for all cacheable accesses or transfers.To enable this function, set the WTENFP bit to 1 in the DCACHE_CTL control register.

The bufferable memory regions are defined and programmed by the user in the MPU, and the MPU is responsible for generating the AHB attribute signals for the given region based on the addressing of any transaction.*Table 4-2. DCACHE cache ability for transaction*, the request to DCACHE is defined as cacheable or uncacheable.

The bufferable memory regions are defined and programmed by the user in the MPU, and the MPU is responsible for generating the AHB attribute signals for the given region based on the addressing of any transaction.

Table 4-2. DCACHE cache ability for transaction

AHB Lookup attribute	Bufferable attribute	Cache ability
0	X	Non-cacheable (Read/Write Non-cacheable) This type of access bypasses the Data Cache (DCACHE); all read and write operations directly access the main memory.
1	0	Read Cacheable Write-Through Policy When the AHB Lookup attribute is 1 and the bufferable attribute is 0, read operations are cacheable, while write operations use the write-through policy - data is written to both the cache and main memory simultaneously.
1	1	Read Cacheable Write-Back Policy When the AHB Lookup attribute is 1 and the bufferable attribute is 1, read operations are cacheable, while write operations use the write-back policy - data is first written into the cache and later written back to main memory.

When DCACHE receives bufferable transactions from the Core, there are two possible outcomes: cache hit and cache miss.

For read transactions:

Cache hit: The TAG storage contains the target address, and the cache line is valid, resulting in a cache hit. Data is read from the cache and provided to the processor within the same cycle.

However, according to the miss-hit characteristic, if the TAG storage does not contain the target address but belongs to an ongoing fill burst from the main memory, a cache hit can still be achieved. During the cache line fill process, as long as the data is available at its host interface, DCACHE can quickly provide the required data, thereby avoiding a cache miss.

Cache miss: The TAG storage does not contain the target address, resulting in a cache miss. Data is read from the main memory and provided to the processor, followed by cache line filling , whether to fill the cache line is determined by the AHB transaction memory allocate attribute. The critical-word-first strategy can optimize waiting cycles, allowing the processor to first read the requested data and then proceed with filling the cache line containing the requested data.

Table 4-3. Reading

Transaction	TAG	Perform operation
Reading	The TAG memory contains the target address, and the cache line is valid (valid=1).	Data is read from the cache and provided to the processor within the same cycle.
	The TAG memory does not contain the target address, and a fill burst from the main memory is in progress.	Data is fetched from the cache and delivered to the processor within the same cycle.
	The TAG memory does not contain the target address.	Data is fetched from the main memory and delivered to the processor.

For write-back transactions:

Cache hit: The TAG storage contains the target address, and the cache line is valid(valid=1), resulting in a cache hit. Data is written into the cache.

However, according to the miss-hit characteristic, if the TAG storage does not contain the target address but belongs to an ongoing fill burst from the main memory, a cache hit can still be achieved. During the cache line fill process, as long as the data is available at its host interface, DCACHE can quickly write the required data, thereby avoiding a cache miss.

Cache miss: The TAG storage does not contain the target address, resulting in a cache miss. Data is read from the main memory and provided to the processor, followed by cache line filling. The critical-word-first strategy can optimize waiting cycles, allowing the processor to first read the requested data and then proceed with filling the cache line containing the requested data.

Table 4-4. Write Back

Transaction	TAG	Perform operation
Write Back	The TAG memory contains the target address, and the cache line is valid (valid=1).	Data is written to the cache.
	The TAG memory does not contain the target address, and a fill burst from the main memory is in progress.	Perform line allocation: read an entire line of data from main memory to refill the cache. After refilling, write the initial data into the cache line (overwriting the corresponding portion of the newly refilled data).
	The TAG memory does not contain the target address.	Directly write the data into the cache line being refilled, without waiting for the refill to complete or accessing the main memory again.

For write-through transactions:

Cache hit: The TAG storage contains the target address, and the cache line is valid(valid=1), resulting in a cache hit. Data is simultaneously written into the cache and the main memory.

Cache miss: The TAG storage does not contain the target address, resulting in a cache miss. The data coming from the slave port is written only to the main memory.

Table 4-5. Write through

Transaction	TAG	Perform operation
Write through	The TAG memory contains the target address, and the cache line is valid (valid=1).	Data is simultaneously written to the cache and main memory.
	The TAG memory does not contain the target address.	Data received from the port is written only to the main memory.

4.3.4. Cache maintenance operations

DCACHE supports multiple maintenance operations that can be programmed by software through the DCACHE_CTL control register:

DCACHE Invalidate operations are available in two forms:

Full Invalidate: invalidates the entire cache range.

Software sets INVALFULL in DCACHE_CTL to 1 to enable the full invalidate operation. Once

enabled, the system sets the FINVLBUSY flag in DCACHE_STAT to indicate the start of the full invalidate process. During execution, the valid bit of each cache line is cleared to 0, and all dirty bits are reset. Meanwhile, INVALFULL is automatically cleared. After the full invalidate operation completes, the hardware clears the FINVLBUSY flag and sets the FINVLENDF flag in DCACHE_STAT. If the BSYENDIE interrupt enable bit in DCACHE_INTEN is set, a DCACHE interrupt will be triggered.

The full invalidate process is non-interruptible; the cache does not process any requests during execution.

Range Invalidate: invalidates cache lines within a specified address range.

Software first configures the invalidate range (start address MOSADDR, end address MOEADDR), then selects the range invalidate mode 3b010 via MOM, and finally sets MOSTART to 1 to enable range invalidation. The MOBUSYF flag in DCACHE_STAT is set simultaneously. After the invalidate operation starts, MOSTART will be automatically cleared. Upon completion, DCACHE clears MOBUSYF and sets MOENDF to 1. If the MOENDIE interrupt bit is enabled, a DCACHE interrupt will be triggered. During this invalidate range operation, new incoming requests will be pending.

DCACHE Clean operations are available in two forms:

Range Clean:

The purpose of cache cleaning is to ensure data consistency between the main memory and the cache. The clean operation writes back cache lines with dirty = 1 to main memory.

Software first sets the clean range (start address MOSADDR, end address MOEADDR), then selects the range clean mode 3b001 via MOM, and finally sets MOSTART to 1 to enable range cleaning. The MOBUSYF flag in DCACHE_STAT is set simultaneously. After the clean operation starts, MOSTART will be automatically cleared. Upon completion, DCACHE clears MOBUSYF and sets MOENDF to 1. If the MOENDIE interrupt bit is enabled, a DCACHE interrupt will be triggered. During this invalidate range operation, new incoming requests will be pending.

Range Clean and Invalidate:

The clean operation writes back cache lines within the specified range that have dirty = 1 to main memory, while the invalidate operation invalidates cache lines within the same range.

Software first sets the clean-and-invalidate range (start address MOSADDR, end address MOEADDR), then selects the range clean-and-invalidate mode 3b011 via MOM, and finally sets MOSTART to 1 to enable the operation. The MOBUSYF flag in DCACHE_STAT is set simultaneously. After the operation starts, MOSTART will be automatically cleared. Upon completion, DCACHE clears MOBUSYF and sets MOENDF to 1. If the MOENDIE interrupt bit is enabled, a DCACHE interrupt will be triggered. During this invalidate range operation, new incoming requests will be pending.

4.3.5. DCACHE performance monitoring

To analyze performance, DCACHE provides two sets of monitors:

Hit monitors: A 32-bit read hit monitor and a 32-bit write hit monitor;

Miss monitors: A 16-bit read miss monitor and a 16-bit write miss monitor.

By default, the monitors are disabled. The hit or miss monitors can be enabled or disabled via the corresponding bits in DCACHE_CTL through software.

4.4. Register definition

DCACHE base address: 0x4002 6400

4.4.1. Control register (DCACHE_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								WMMRST	WHMRST	WMMEN	WHMEN	RMMRST	RHMRST	RMMEN	RHMEN
								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				STARTMO	MOM[2:0]			Reserved					WTENFP	INVALFULL	DCACHEEN
				rw	rw								rw	w	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value
23	WMMRST	Write-miss monitor reset 0: No effect 1: Reset write-miss monitor
22	WHMRST	Write-hit monitor reset 0: No effect 1: Reset write-hit monitor
21	WMMEN	Write-miss monitor enable 0: Switch off write-miss monitor, but does not reset. 1: write-miss monitor enabled
20	WHMEN	Write-hit monitor enable 0: Switch off write-hit monitor, but does not reset. 1: write-hit monitor enabled
19	RMMRST	Read-miss monitor reset 0: No effect 1: Reset read-miss monitor
18	RHMRST	Read-Hit monitor reset 0: No effect 1: Reset read-hit monitor
17	RMMEN	Read-miss monitor enable

		0: Switch off read-miss monitor, but does not reset. 1: Read-miss monitor enabled
16	RHMEN	Read-hit monitor enable 0: Switch off read-hit monitor, but does not reset. 1: Read-hit monitor enabled
15:12	Reserved	Must be kept at reset value
11	STARTMO	Start cache maintenance operations. software programmable exclusively under conditions: DCACHEEN = 1, MOBUSYF =0, FINVLBUSY =0 and MOM =0b001, 0b010 or 0b011. Hardware-reset upon MOBUSYF flag assertion (throughout cache servicing procedure). Writing 0 produces no effect. 0: maintenance operations completed 1: performing cache maintenance operations
10:8	MOM[2:0]	maintenance operations mode Can be set and cleared by software, only when no maintenance command is ongoing (MOBUSYF = 0). 000: no operation 001: clean range 010: invalidate range 011: clean and invalidate range 100-111: reserved
7:3	Reserved	Must be kept at reset value
2	WTENFP	Write-Through with Enforcement policy 0: disable Write-Through with Enforcement policy 1: enable Write-Through with Enforcement policy
1	INVALFULL	invalidate the full cache set by software and cleared by hardware when the full invalidation operation. 0: No effect 1: invalidate entire cache
0	DCACHEEN	Dcache enable 0: Dcache disabled 1: Dcache enabled

4.4.2. Status register (DCACHE_STAT)

Address offset: 0x04

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											MOENDF	MOBUSYF	ERRF	FINVLENDF	FINVLBUSY
											r	r	r	r	r

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value
4	MOENDF	maintenance operation end flag 0: cache busy or in idle 1: maintenance operation ended
3	MOBUSYF	maintenance operation busy flag 0: maintenance operation not busy 1: maintenance operation busy
2	ERRF	dcache error flag 0: No error 1: Error occurred during the operation
1	FINVLENDF	full invalidate operation end flag 0: Cache busy 1: full invalidate operation ended
0	FINVLBUSY	full invalidate operation busy flag 0: Cache is not executing a full invalidate operation 1: Cache is executing a full invalidate operation

4.4.3. Interrupt enable register (DCACHE_INTEN)

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											MOENDIE	Reserved	ERRIE	BUSYENDIE	Reserved
											rw		rw	rw	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value

4	MOENDIE	maintenance operation end interrupt enable 0: Disable maintenance operation end interrupt 1: Enable maintenance operation end interrupt
3	Reserved	Must be kept at reset value
2	ERRIE	Cache error interrupt enable 0: Disable cache error interrupt 1: Enable cache error interrupt
1	BUSYENDIE	busy end interrupt enable 0: Disable busy end interrupt 1: Enable busy end interrupt
0	Reserved	Must be kept at reset value.

4.4.4. Flag clear register (DCACHE_FC)

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											MOENDFC	Reserved	ERRFC	FINVLENDFC	Reserved
											w		w	w	

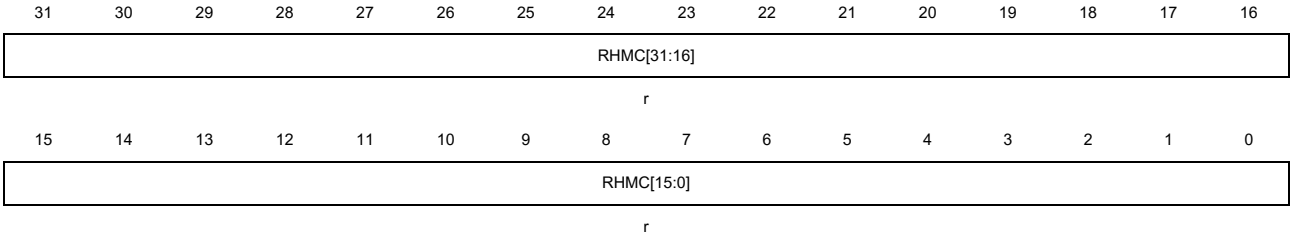
Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value
4	MOENDFC	Software clear maintenance operation end flag 0: No effect 1: Clears MOENDF flag in DCACHE_STAT.
3	Reserved	Must be kept at reset value
2	ERRFC	Software clear cache error flag 0: No effect 1: Clears ERRF flag in DCACHE_STAT
1	FINVLENDFC	Software clear full invalidate operation end flag 0: No effect 1: Clears FINVLENDF flag in DCACHE_STAT.
0	Reserved	Must be kept at reset value

4.4.5. Read-hit monitor counter register (DCACHE_RHMC)

Address offset: 0x10

Reset value: 0x0000 0000

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



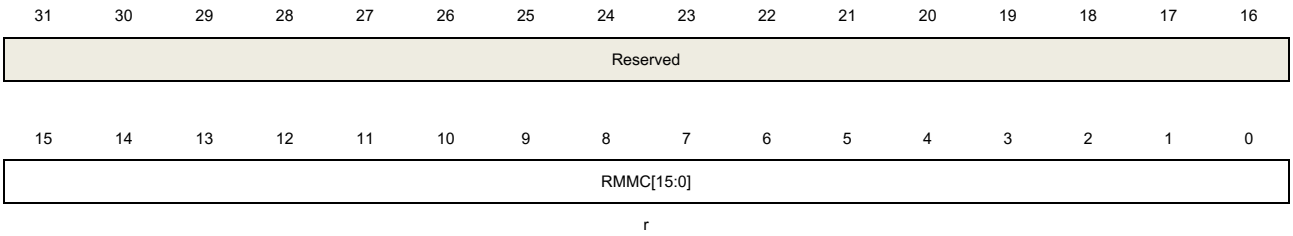
Bits	Fields	Descriptions
31:0	RHMC[31:0]	Read-hit monitor counter.

4.4.6. Read-miss monitor counter register (DCACHE_RMMC)

Address offset: 0x14

Reset value: 0x0000 0000

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



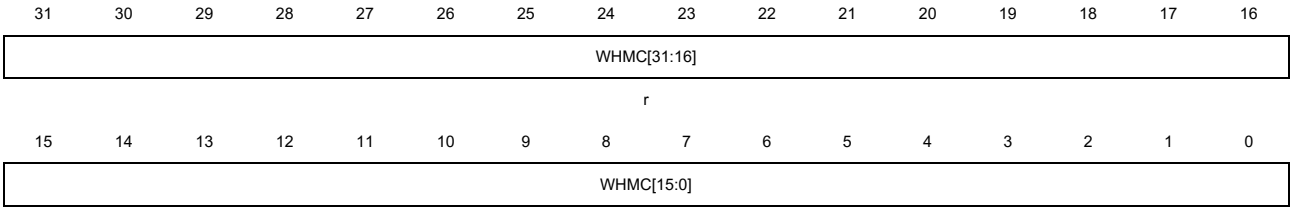
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	RMMC[15:0]	Read-miss monitor counter

4.4.7. Write-hit monitor counter register (DCACHE_WHMC)

Address offset: 0x20

Reset value: 0x0000 0000

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



r

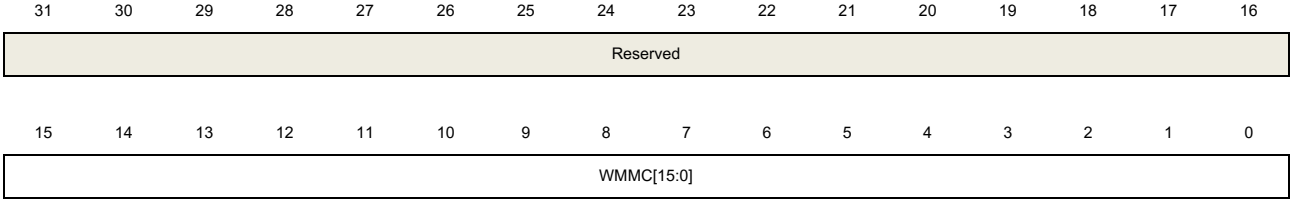
Bits	Fields	Descriptions
31:0	WHMC[31:0]	Write-hit monitor counter.

4.4.8. Write-miss monitor counter register (DCACHE_WMMC)

Address offset: 0x24

Reset value: 0x0000 0000

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



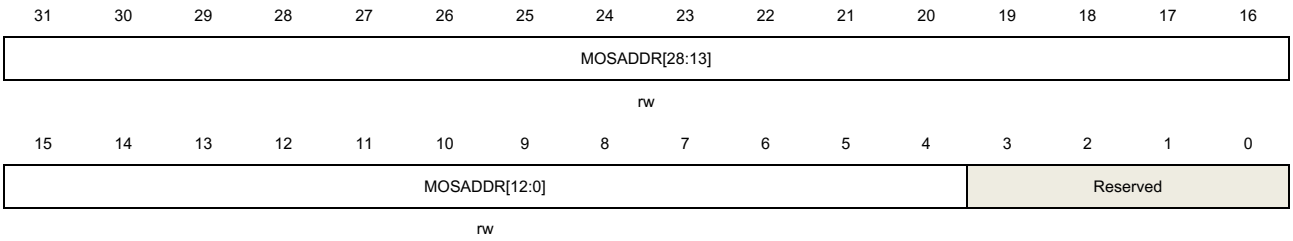
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	WMMC[15:0]	Write-miss monitor counter

4.4.9. Maintenance operation start address register (DCACHE_MOSADDR)

Address offset: 0x28

Reset value: 0x0000 0000

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



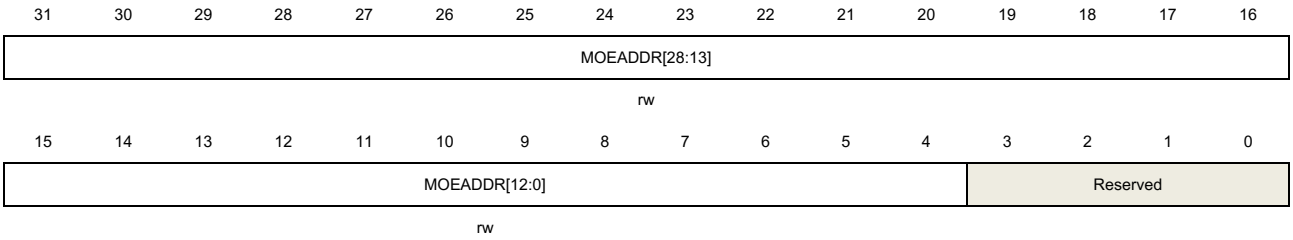
Bits	Fields	Descriptions
31:4	MOSADDR[28:0]	Maintenance operation start address
3:0	Reserved	Must be kept at reset value

4.4.10. Maintenance operation end address register (DCACHE_MOEADDR)

Address offset: 0x2C

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:4	MOEADDR[28:0]	Maintenance operation end address
3:0	Reserved	Must be kept at reset value

5. Power management unit (PMU)

5.1. Overview

The power consumption is regarded as one of the most important issues for the devices of this series. The Power management unit (PMU) provides four types of power saving modes, including Sleep, Deep-sleep, Deep-sleep 1 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 ***Figure 5-1. Power supply overview***. The power of the V_{DD} domain is supplied directly by V_{DD}. An embedded LDO in the V_{DD} / V_{DDA} domain is used to supply the V_{CORE} domain power. A power switch is implemented for the Backup domain. It can be powered from the V_{BAT} voltage when the main V_{DD} supply is shut down.

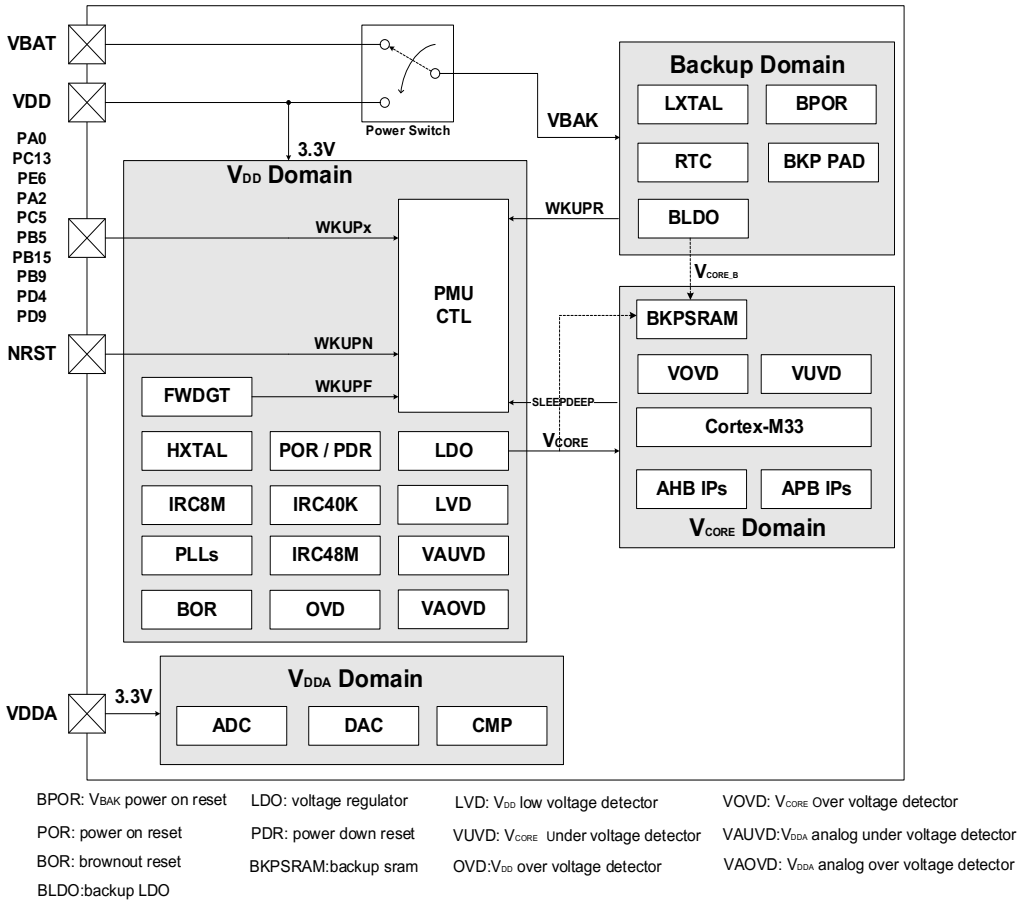
5.2. Characteristics

- Three power domains: V_{BAK}, V_{DD} / V_{DDA} and V_{CORE} power domains.
- Four power saving modes: Sleep, Deep-sleep, Deep-sleep 1 and Standby modes.
- Internal Voltage regulator (LDO) supplies around V_{CORE} voltage source for V_{CORE} domain. and a Backup LDO dedicate to Backup SRAM.
- Low Voltage Detector can issue an interrupt or event when the power is lower than a programmed threshold.
- Battery power (V_{BAT}) for Backup domain when V_{DD} is shut down.
- 4K bytes backup SRAM powered by V_{CORE} which source from V_{DD} or V_{BAK} for data protection of user application data when V_{DD} shut down.
- LDO output voltage select for power saving.
- Power supply supervision: POR / PDR monitor / LVD monitor / OVD monitor / VOVD monitor / VAUVD monitor / VAOVD monitor / VUVD monitor.

5.3. Function overview

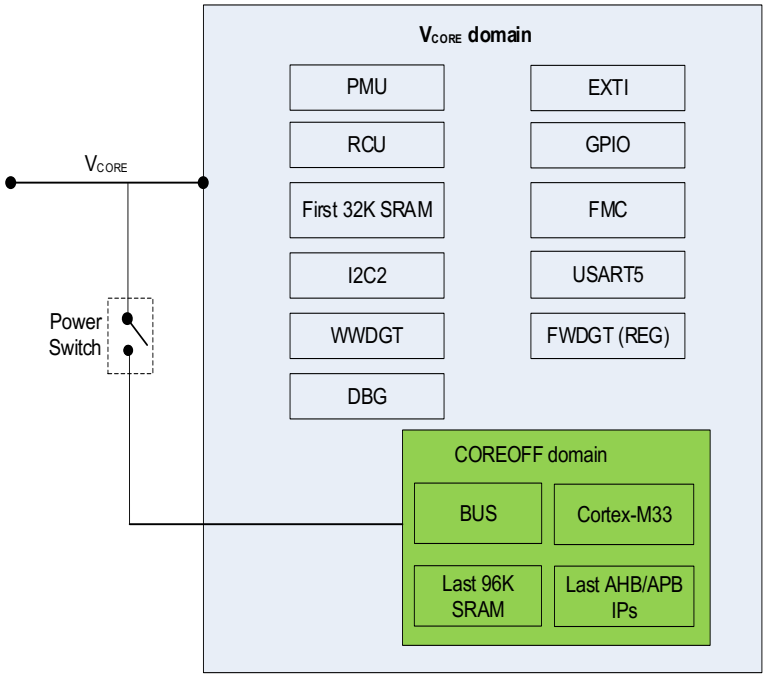
Figure 5-1. Power supply overview provides details on the internal configuration of the PMU and the relevant power domains.

Figure 5-1. Power supply overview



BPOR: V_{BAK} power on reset
POR: power on reset
BOR: brownout reset
BLDO:backup LDO
LDO: voltage regulator
PDR: power down reset
BKPSRAM:backup sram
LVD: V_{DD} low voltage detector
VUVD: V_{CORE} Under voltage detector
OVD:V_{DD} over voltage detector
VOVD: V_{CORE} Over voltage detector
VAUVD:V_{DDA} analog under voltage detector
VAOVD: V_{DDA} analog over voltage detector

Figure 5-2. V_{CORE} domain supply overview



As shown in ***Figure 5-2. VCORE domain supply overview***, the V_{CORE} domain includes two parts:

1. COREOFF domain includes:
   - Last AHB/APB IPs: all V_{CORE} peripherals except the FMC, PMU, RCU, EXTI, GPIO, DBG, FWDGT (registers only), WWDGT, USART5, I2C2, SYSCFG.
   - Last 96K SRAM: SRAM (except the first 32K).
   - Cortex®-M33 CPU.
   - BUS: consists of an AHB interconnect matrix, one AHB bus and two APB buses.
2. Deep-sleep power on modules: FMC, PMU, RCU, EXTI, GPIO, DBG, FWDGT (only registers), WWDGT, USART5, I2C2, SYSCFG, BKPSRAM, VOVD, VUVD and the first 32K SRAM.

The power switch is used for power supply control of power saving modes.

5.3.1. Battery backup domain

The Backup domain is powered by the V_{DD} or the battery power source (V_{BAT}) selected by the internal power switch, and the VBAK pin which drives Backup Domain, power supply for RTC unit, LXTAL oscillator, BPOR, BLDO(backup LDO) and three pads, including PC13 to PC15. In order to ensure the content of the Backup domain registers and the RTC supply, when V_{DD} 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. When V_{DD} is shut down, only the LXTAL is valid for the RTC. Before entering the power saving mode by executing the WFI / WFE instruction, the Cortex®-M33 can setup the RTC register with an expected alarm time and enable the alarm function and according EXTI lines to achieve the RTC alarm event. After entering the power saving mode for a certain amount of time, the RTC alarm will wake up the device when the time match event occurs. The details of the RTC configuration and operation will be described in the ***Real time clock (RTC)***

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

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

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

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

Note: Since PC13, PC14, 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).

5.3.2. Backup SRAM

There is 4K bytes backup SRAM in V_{CORE} domain. The backup SRAM can maintain data in Standby mode or VDD is shut down when BLDOON bit is set in PMU_CTL1 register. In these modes, the backup SRAM powered by BLDO which source from V_{BAK}, and the BLDO voltage level selection refers to the VCSCAL[1:0] bit in RTC_CTL register. In other modes (not in Standby mode or VDD is shut down), the backup SRAM can be accessed by system bus as normal SRAM and powered by LDO source from VDD.

The backup SRAM can only be accessed by user code when FMC is in security protection level low mode to prevent illegal code / data access. When the FMC goes from security protection level low mode to no security protection mode, the backup SRAM will be erased and all data are lost. The backup SRAM is not reset by BKPRST in RCU_BDCTL register.

5.3.3. VDD / VDDA power domain

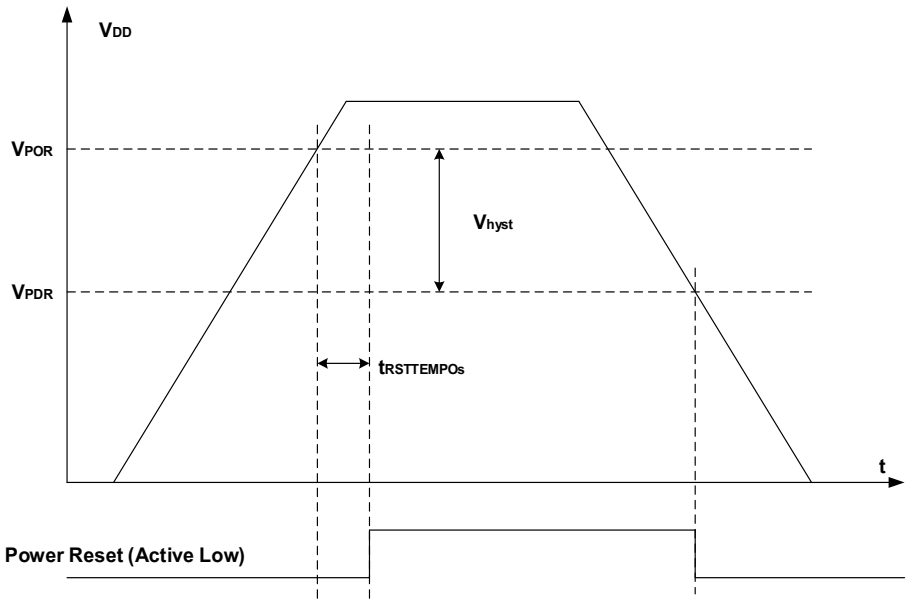
VDD / VDDA domain includes two parts: VDD domain and VDDA domain. VDD domain includes POR / PDR (power on / down reset), HXTAL (high speed crystal oscillator), LDO (voltage regulator), LVD(Low Voltage Detector), OVD(over Voltage Detector), FWDGT (free watchdog timer), IRC8M (internal 8MHz RC oscillator),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

V_{DD} domain

The LDO, which is implemented to supply power for the V_{CORE} domain, is always enabled after reset. It can be configured to operate in three different status, including in the Sleep mode (full power on), in the Deep-sleep / Deep-sleep 1 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 5-3. 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.

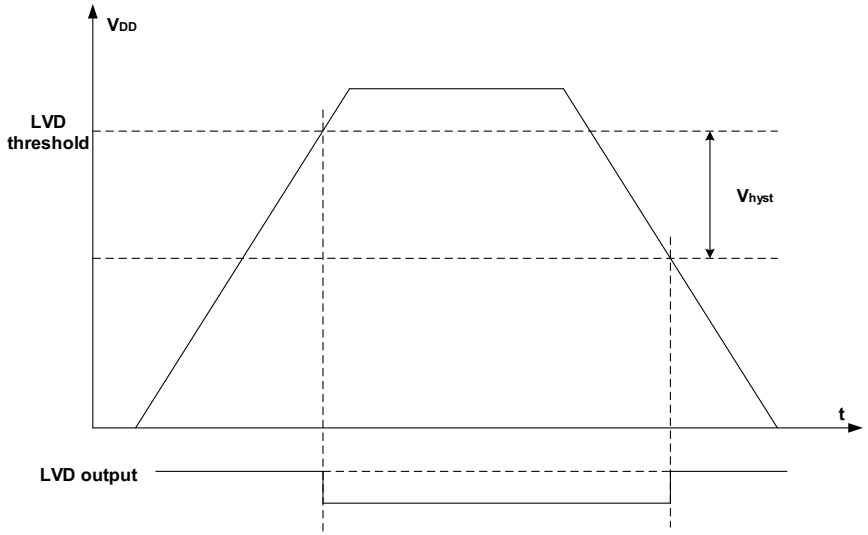
Figure 5-3. Waveform of the POR/PDR



The LVD is used to detect whether the V_{DD} supply voltage is lower than a programmed threshold selected by the LVDT[3:0] bits in the Power control register(PMU_CTL0). The LVD

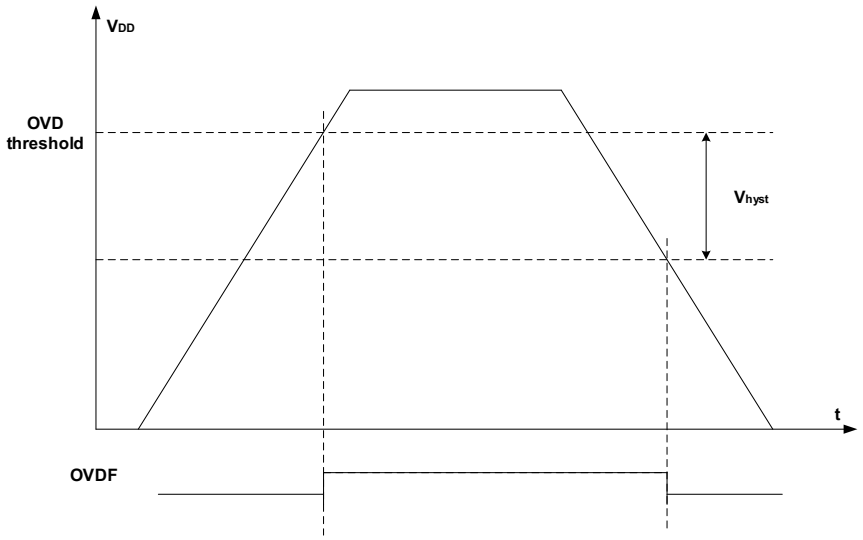
is enabled by setting the LVDEN bit, and LVDF bit, which in the Power status register(PMU_CS0), indicates if V_{DDA} is higher or lower than the LVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. ***Figure 5-4. Waveform of the LVD threshold*** shows the relationship between the LVD threshold and the LVD output (LVD interrupt signal depends on EXTI line 16 rising or falling edge configuration). The following figure shows the relationship between the supply voltage and the LVD signal. Refer to the datasheet for the hysteresis voltage (V_{hyst}) value.

Figure 5-4. Waveform of the LVD threshold



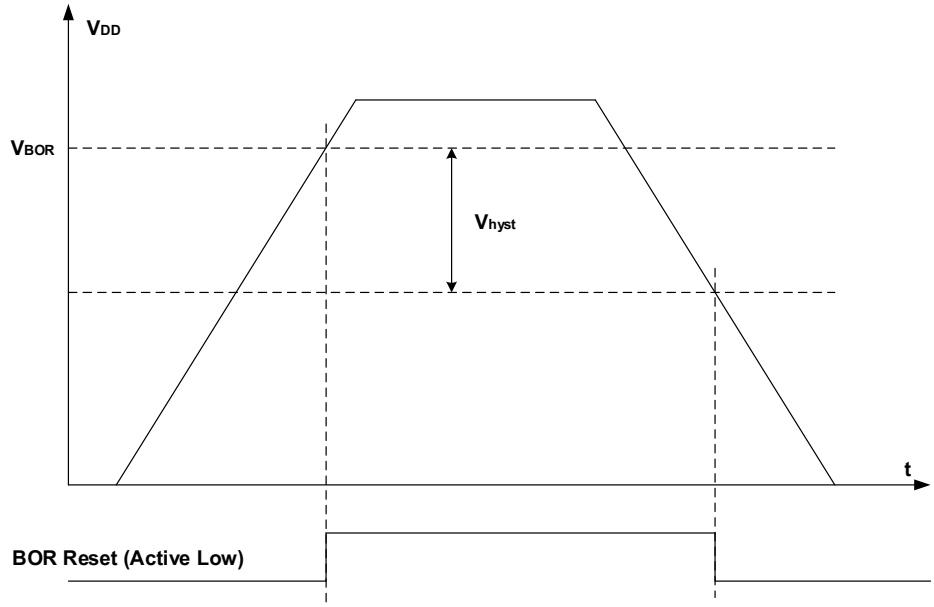
The OVD is used to detect whether the V_{DD} supply voltage is higher than a programmed threshold selected by the OVDT[1:0] bits in the Power control register(PMU_CTL3). The OVD is enabled by setting the OVDEN bit in the Power control register(PMU_CTL3), and OVDF bit, which in the Power status register(PMU_CS0), indicates if V_{DD} is higher or lower than the OVD 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 5-5. Waveform of the OVD threshold*** shows the relationship between the OVD threshold and the OVDF flag. Refer to the datasheet for the hysteresis voltage (V_{hyst}) value.

Figure 5-5. Waveform of the OVD threshold



The BOR circuit is used to detect V_{DD} and generate system reset which resets the whole chip except the Backup domain and LDO when the BOR_EN bits in option bytes is enable and the supply voltage is lower than the specified threshold which defined in the BOR_TH bits in option bytes. ***Figure 5-6. Waveform of the BOR*** shows the relationship between the supply voltage and the BOR reset signal. V_{BOR}, which defined in the BOR_TH bits in option bytes, indicates the threshold of BOR on reset. The V_{hyst} is hysteresis voltage.

Figure 5-6. Waveform of the BOR

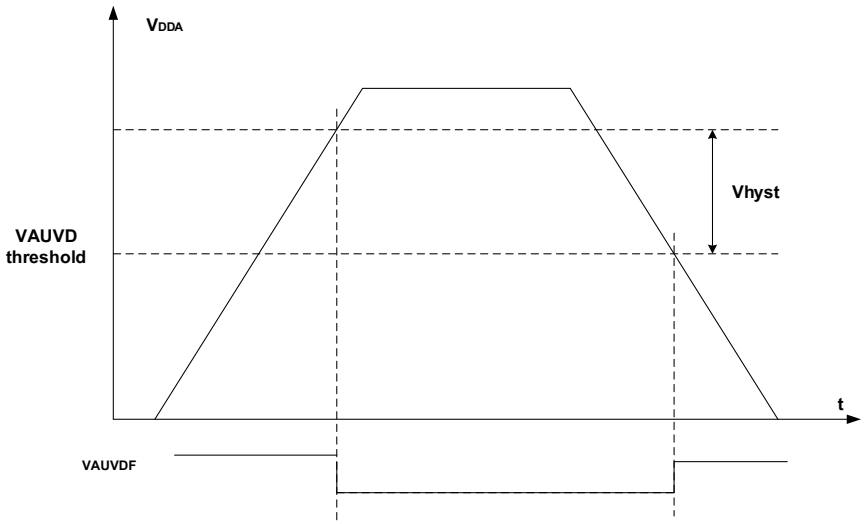


V_{DDA} domain

The V_{DDA} analog voltage detector is used to detect whether the VDDA supply voltage is lower than a programmed threshold selected by the VAUVDVC[3:0] bits in the power control

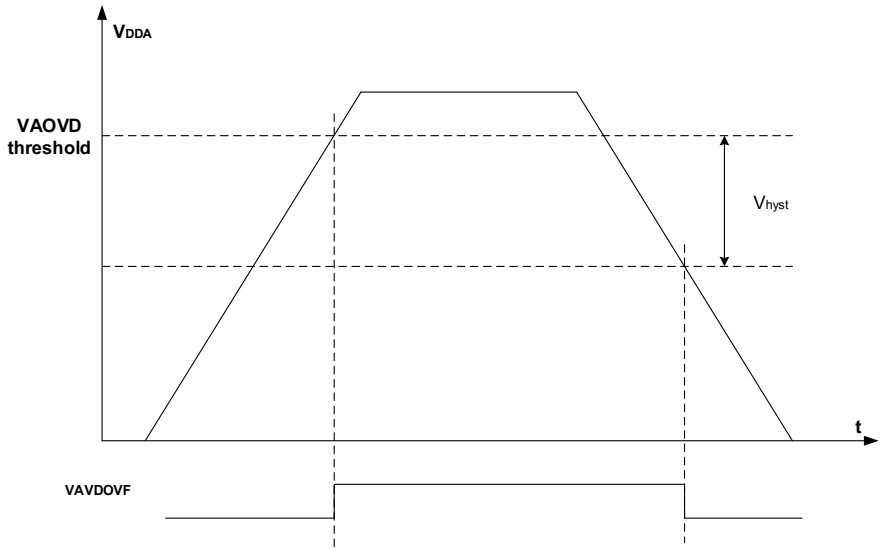
register(PMU_CTL0). The VAUVD is enabled by setting the VAUVDEN bit, and VAUVDF bit, which in PMU_CS0, indicates if V_{DDA} is higher or lower than the specified VAUVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers.*Figure 5-7. Waveform of the VAUVD threshold* shows the relationship between the VAUVD threshold and the VAUVDF The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 5-7. Waveform of the VAUVD threshold



The V_{DDA} analog over voltage detector is used to detect whether the VDDA supply voltage is higher than a programmed threshold selected by the VAOVDVC[1:0] bits in the power control register(PMU_CTL3). The VAOVD is enabled by setting the VAOVDEN bit, and VAOVDF bit, which in PMU_CS0, indicates if V_{DDA} is higher or lower than the specified VAOVD 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 5-8. Waveform of the VAOVD threshold*shows the relationship between the VAOVD threshold and the VAOVDF flag. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 5-8. Waveform of the VAOVD threshold



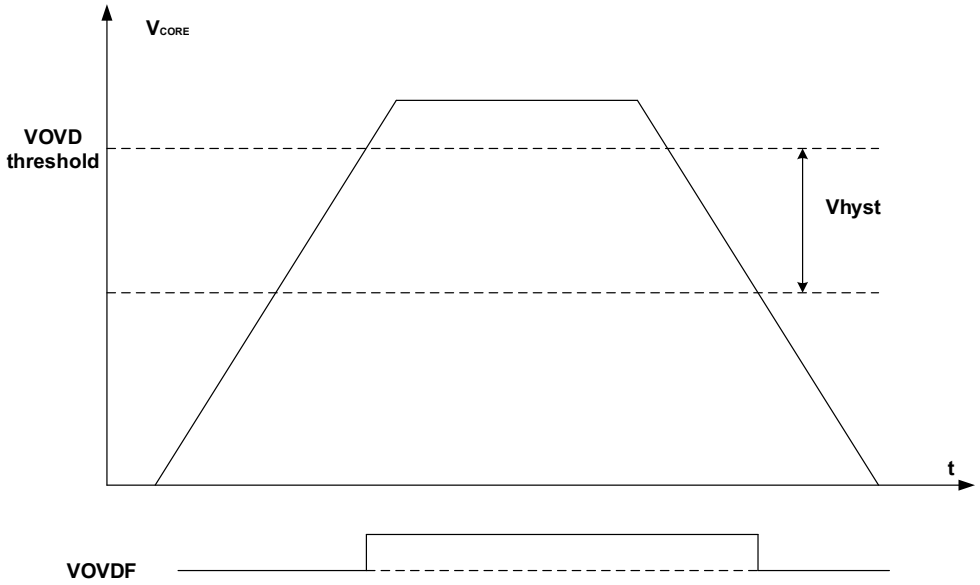
Generally, digital circuits are powered by V_{DD}, while most analog circuits are powered by V_{DDA}. To improve the ADC and DAC conversion accuracy, the independent power supply V_{DDA} is implemented to achieve better performance of analog circuits. V_{DDA} can be externally connected to V_{DD} through the external filtering circuit that avoids noise on V_{DDA}, and V_{SSA} should be connected to V_{SS} through the specific circuit independently. Otherwise, when the V_{DD} and V_{DDA} are provided by different power supplies, the difference between V_{DD} and V_{DDA} during power-up and running time 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 18-2. ADC input pins definition*** and ***Table 19-1. DAC I/O description***, VREFN pin must be connected to VSSA pin. The VREFP pin is only available on no less than 100-pin packages, or else the VREFP pin is not available and internally connected to VDDA. The VREFN pin is only available on no less than 100-pin packages, or else the VREFN pin is not available and internally connected to V_{SSA}.

5.3.4. V_{CORE} power domain

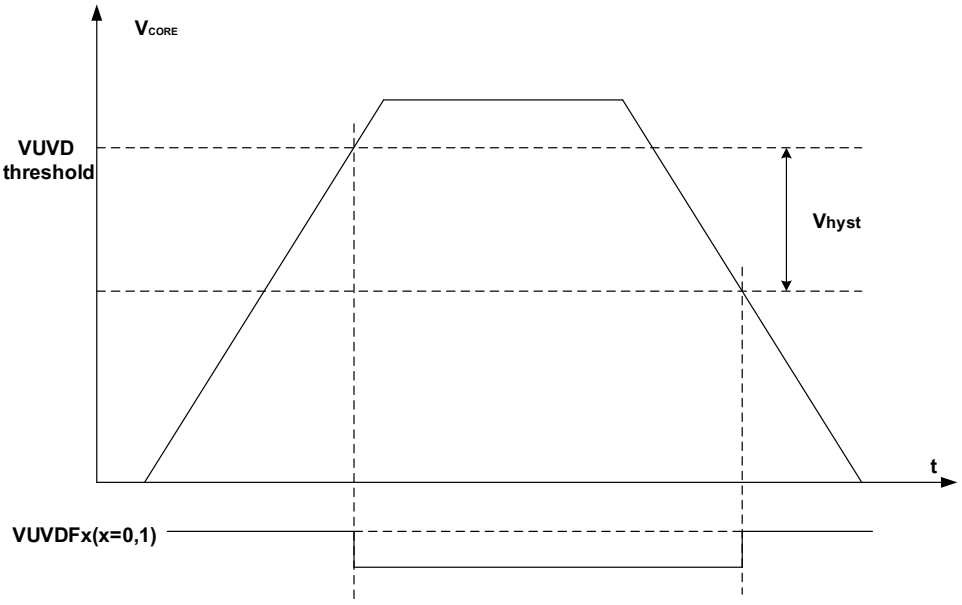
There is an internal V_{CORE} power over voltage detector, when VOVDEN is 0b1, it means V_{CORE} power over voltage detector is enabled. Once V_{CORE} power domain is over than a programmed threshold selected by the VOVDVC[1:0] bits in the power control register(PMU_CTL0), VOVDF will be set immediately after two flip-flops synchronization of the analog output. VOVDF will also be set after digital filter which can be configured by the VOVDO_DNF[7:0] bit in PMU_CTL2 register. This allows to suppress spikes with a programmable length of 1 to 255 multiples of 1024* T_{pclk1}. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 5-9. waveform of VOVD



There is an internal V_{CORE} 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 also be set after digital filter which can be configured by the VUVDO_DNF[7:0] bits in PMU_CTL2 register. This allows to suppress spikes with a programmable length of 1 to 255 multiples of 1024* T_{pclk1}. The hysteresis voltage (Vhyst) please refer to device datasheet.

Figure 5-10. waveform of VUVD



5.3.5. Power saving modes

After a system reset or a power reset, 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. Besides, four power saving modes are provided to achieve even lower power consumption, they are Sleep mode, Deep-sleep mode, Deep-sleep 1 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_CTL0 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_CTL0 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.

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 8-3. EXTI source***. If not, the program will skip the entry process of Deep-sleep mode to continue to execute the

following procedure.

Deep-sleep 1 mode

The Deep-sleep 1 mode is based on the SLEEPDEEP mode of the Cortex®-M33. In Deep-sleep 1 mode, all clocks in the V_{CORE} domain are off, and all of IRC8M, IRC48M, HXTAL and PLLs are disabled. The COREOFF domain is powered off. The contents of SRAM(except for the first 32K) and COREOFF domain peripheral register are lost. The LDO can operate normally or in low power mode depending on the LDOLP bit in the PMU_CTL0 register. Before entering the Deep-sleep 1 mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M33 System Control Register, clear the STBMOD bit in the PMU_CTL0 register, and set DPMOD1 bit in the PMU_CTL1 register. Then, the device enters the Deep-sleep 1 mode after a WFI or WFE instruction is executed. If the Deep-sleep 1 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 from the Deep-sleep 1 mode, the IRC8M is selected as the system clock. The Cortex®-M33 will execute instruction code from the 0x00000000 address. Wake up from Deep-sleep mode 1 requires additional delay time to power on the COREOFF domain. Notice that an additional wakeup delay will be incurred if the LDO operates in low power mode.

Note: To smoothly enter Deep-sleep mode 1, all the suspended states on the EXTI line (in the EXTI_PD register) and related peripheral flag bits must be reset. Otherwise, the program will directly skip the Deep-sleep Mode 1 entry process and continue to execute the following program.

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_CTL0 register, and clear WUF bit in the PMU_CS0 register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS0 register indicates that the MCU has been in Standby mode. There are five wakeup sources for the Standby mode, including the external reset from NRST pin, the rising edge on WKUP pins, the FWDGT reset, the RTC event and LXTAL clock monitor, refer to ***Table 22-2. RTC power saving mode management***.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(except backup SRAM, when BLDOON bit is set) 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 5-1. Power saving mode summary

Mode	Sleep	Deep-sleep	Deep-sleep 1	Standby
Description	Only CPU clock is off	1. All clocks in the V_{CORE} domain are off. 2. Disable IRC8M, IRC48M, HXTAL and PLLs.	1. All clocks in the VCORE domain are off. 2. Disable IRC8M, IRC48M, HXTAL and PLLs. 3. Close CPU, SRAM(except for the first 32K), COREOFF domain peripheral power supply	1. The V_{CORE} domain is power off. 2. Disable IRC8M, IRC48M, HXTAL and PLLs.
LDO Status	On (normal power mode)	On (normal or low power mode)	On (normal or low power mode)	Off
Configuration	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 0 DPMOD1 = 1	SLEEPDEEP = 1 STBMOD = 1, WURST=1
Entry	WFI or WFE	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	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	1. NRST pin 2. WKUP pins 3. FWDGT reset 4. RTC event refer to ***RTC power saving mode management*** 5. LCKM
Wakeup Latency	None	4. IRC8M wakeup time. 5. LDO wakeup time added if LDO is in low power mode	1. IRC8M wakeup time. 2. Power-on time of COREOFF domain. 3. 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 pins if enabled.

5.4. Register definition

PMU base address: 0x4000 7000

5.4.1. Control register 0 (PMU_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000 (reset by wakeup from Standby mode)

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
UNLOCK	Reserved		VUVDVC[1:0]		VOVDVC[1:0]		VUVDEN	VOVDEN	VAUVDVC[3:2]		VAUVDEN	Reserved			
rw			rw		rw		rw	rw	rw		rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				VAUVDVC[1:0]		LVDT[3]	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
				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 (VDD reset) These bits are set and cleared by software 00: Configure V_{CORE} under voltage detector voltage level to 1.075V 01: Configure V_{CORE} under voltage detector voltage level to 0.975V. 1x: Configure V_{CORE} under voltage detector voltage level to 0.875V.
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	VAUVDVC[3:2]	combination with 11:10 bits
20	VAUVDEN	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.
19:12	Reserved	Must be kept at reset value.
11:10	VAUVDVC[1:0]	VDDA analog voltage detector voltage level configure bits These bits are set and cleared by software 0000: Configure V_{DDA} analog voltage detector voltage level to 1.8V 0001: Configure V_{DDA} analog voltage detector voltage level to 1.9V 0010: Configure V_{DDA} analog voltage detector voltage level to 2.0V 0011: Configure V_{DDA} analog voltage detector voltage level to 2.1V 0100: Configure V_{DDA} analog voltage detector voltage level to 2.2V 0101: Configure V_{DDA} analog voltage detector voltage level to 2.3V 0110: Configure V_{DDA} analog voltage detector voltage level to 2.4V 0111: Configure V_{DDA} analog voltage detector voltage level to 2.5V 1000: Configure V_{DDA} analog voltage detector voltage level to 2.6V 1001: Configure V_{DDA} analog voltage detector voltage level to 2.7V 1010: Configure V_{DDA} analog voltage detector voltage level to 2.8V 1011: Configure V_{DDA} analog voltage detector voltage level to 2.9V 1100: Configure V_{DDA} analog voltage detector voltage level to 3.0V 1101: Configure V_{DDA} analog voltage detector voltage level to 3.1V 1110: Configure V_{DDA} analog voltage detector voltage level to 3.2V 1111: Configure V_{DDA} analog voltage detector voltage level to 3.3V
9	LVDT[3]	combination with 7:5 bits
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 0000：1.8V 0001：1.9V 0010：2.0V 0011：2.1V 0100：2.2V 0101：2.3V 0110：2.4V 0111：2.5V

		1000：2.6V 1001：2.7V 1010：2.8V 1011：2.9V 1100：3.0V 1101：3.1V 1110：3.2V 1111：3.3V
4	LVDEN	Low Voltage Detector Enable 0: Disable Low Voltage Detector 1: Enable Low Voltage Detector
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 / Deep-sleep 1 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 / Deep-sleep 1 1: The LDO is in low power mode during the Deep-sleep / Deep-sleep 1 **Note:** Some peripherals may work with the IRC8M clock in the Deep-sleep / Deep-sleep 1 mode. In this case, the LDO automatically switches from the low power mode to the normal mode and remains in this mode until the peripheral stop working.

5.4.2. Control and status register 0 (PMU_CS0)

Address offset: 0x04
Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

Note: Due to the influence of polarity, pull-up/pull-down configuration, and the initial input level, enabling the WKUP pin wake-up function by software may cause the WUF bit to be set to 1. It is recommended that after enabling WUPENx, the software should clear WUF first, and then enter the low-power mode.

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							VUVDF1	VOVDF	Reserved					OVDF	VAOVDF
							r	r						r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUPEN7	LDOVSRF	WUPEN5	WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	WUPEN6	WUPEN8	WUPEN9	VUVDF0	VAUVDF	LVDF	STBF	WUF
rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rc_w1	r	r	r	r

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	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 VUVDVC[1:0] threshold. 1: V_{CORE} is equal or lower than VUVDVC[1:0] threshold.
23	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
22:18	Reserved	Must be kept at reset value.
17	OVDF	V_{DD} Over Voltage Detector Status Flag 0: Over Voltage event has not occurred (V_{DD} is lower than the specified OVDT[1:0] threshold) 1: Over Voltage event occurred (V_{DD} is equal to or higher than the specified OVDT[1:0] threshold)
16	VAOVDF	VDDA analog voltage detector over voltage output status flag This bit is set and cleared by hardware. It is valid only if VAOVDEN is enabled. 0: V_{DDA} is lower than the VAOVD threshold configured by VAOVDVC[1:0] bits. 1: V_{DDA} is equal or higher than the VAOVD threshold configured by VAOVDVC[1:0] bits.
15	WUPEN7	WKUP Pin7(PB9) Enable 0: Disable WKUP pin7 function 1: Enable WKUP pin7 function If WUPEN7 is set before entering the power saving mode, a rising or falling edge on the WKUP pin7 wakes up the system from the power saving mode.
14	LDOVSRF	LDO voltage select ready flag 0: LDO voltage select not ready 1: LDO voltage select ready
13	WUPEN5	WKUP Pin5(PB5) Enable 0: Disable WKUP pin5 function

		1: Enable WKUP pin5 function If WUPEN5 is set before entering the power saving mode, a rising or falling edge on the WKUP pin5 wakes up the system from the power saving mode.
12	WUPEN4	WKUP Pin4(PC5) Enable 0: Disable WKUP pin4 function 1: Enable WKUP pin4 function If WUPEN4 is set before entering the power saving mode, a rising or falling edge on the WKUP pin4 wakes up the system from the power saving mode.
11	WUPEN3	WKUP Pin3(PA2) Enable 0: Disable WKUP pin3 function 1: Enable WKUP pin3 function If WUPEN3 is set before entering the power saving mode, a rising or falling edge on the WKUP pin3 wakes up the system from the power saving mode.
10	WUPEN2	WKUP Pin2(PE6) Enable 0: Disable WKUP pin2 function 1: Enable WKUP pin2 function If WUPEN2 is set before entering the power saving mode, a rising or falling edge on the WKUP pin2 wakes up the system from the power saving mode.
9	WUPEN1	WKUP Pin1(PC13) Enable 0: Disable WKUP pin1 function 1: Enable WKUP pin1 function If WUPEN1 is set before entering the power saving mode, a rising or falling edge on the WKUP pin1 wakes up the system from the power saving mode.
8	WUPEN0	WKUP Pin0(PA0) Enable 0: Disable WKUP pin0 function 1: Enable WKUP pin0 function If WUPEN0 is set before entering the power saving mode, a rising or falling edge on the WKUP pin0 wakes up the system from the power saving mode.
7	WUPEN6	WKUP Pin6(PB15) Enable 0: Disable WKUP pin6 function 1: Enable WKUP pin6 function If WUPEN6 is set before entering the power saving mode, a rising or falling edge on the WKUP pin6 wakes up the system from the power saving mode.
6	WUPEN8	WKUP Pin8(PD4) Enable 0: Disable WKUP pin8 function 1: Enable WKUP pin8 function If WUPEN8 is set before entering the power saving mode, a rising or falling edge on the WKUP pin8 wakes up the system from the power saving mode.
5	WUPEN9	WKUP Pin9(PD9) Enable

0: Disable WKUP pin9 function
1: Enable WKUP pin9 function
If WUPEN9 is set before entering the power saving mode, a rising or falling edge on the WKUP pin9 wakes up the system from the power saving mode.

4	VUVDF0	Record the undervoltage reset of the V_{CORE} 0: No undervoltage reset happened 1: Undervoltage reset happened
3	VAUVDF	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 VAUVDEN is enabled. 0: V_{DDA} is equal or higher than the VAUVD threshold configured by VAUVDVC bits. 1: V_{DDA} is lower than the VAUVD threshold configured by VAUVDVC 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, LCKM or RTC event This bit is cleared only by a POR / PDR or by setting the WURST bit in the PMU_CTL0 register.

5.4.3. Control register 1 (PMU_CTL1)

Address offset: 0x08
Reset value: 0x0000 0000 (reset by wakeup from Standby mode)

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
WUPP9	WUPP8	WUPP7	WUPP6	WUPP5	WUPP4	WUPP3	WUPP2	WUPP1	WUPP0	Reserved			BLDORF	BLDOON	DPMOD1
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	WUPP9	Polarity selection for the WKUP pin (PD9) 0: No reversal 1: Reversal
14	WUPP8	Polarity selection for the WKUP pin (PD4) 0: No reversal 1: Reversal
13	WUPP7	Polarity selection for the WKUP pin (PB9) 0: No reversal 1: Reversal
12	WUPP6	Polarity selection for the WKUP pin (PB15) 0: No reversal 1: Reversal
11	WUPP5	Polarity selection for the WKUP pin (PB5) 0: No reversal 1: Reversal
10	WUPP4	Polarity selection for the WKUP pin (PC5) 0: No reversal 1: Reversal
9	WUPP3	Polarity selection for the WKUP pin (PA2) 0: No reversal 1: Reversal
8	WUPP2	Polarity selection for the WKUP pin (PE6) 0: No reversal 1: Reversal
7	WUPP1	Polarity selection for the WKUP pin (PC13) 0: No reversal 1: Reversal
6	WUPP0	Polarity selection for the WKUP pin (PA0) 0: No reversal 1: Reversal
5:3	Reserved	Must be kept at reset value.
2	BLDORF	Backup voltage stabilizer ready flag This bit is set by hardware to indicate that the backup regulator is ready. 0: Backup voltage stabilizer not ready. 1: Backup voltage stabilizer ready.
1	BLDOON	Backup voltage stabilizer enable

		This bit is set and cleared by software. When set, the backup voltage stabilizer (used to maintain the backup SRAM content in Standby and V_{BAT} modes) is enabled. When reset, the backup voltage stabilizer is off. The backup SRAM can still be used in Run and Deep-sleep modes. However, its content will be lost in Standby and V_{BAT} modes. If enable, the application must wait till the backup voltage stabilizer ready flag (BLDORF) is set. Which indicate that the data written into the BKPSRAM will be maintained in Standby and VBAT modes. 0: Backup voltage stabilizer disabled. 1: Backup voltage stabilizer enabled.
0	DPMOD1	Deep-sleep 1 mode enable 0: Not care 1:Go to Deep-sleep 1 mode when the SLEEPDEEP bit is set and the STBMOD bit is clear.

5.4.4. Control and status register 1 (PMU_CS1)

Address offset: 0x0C
Reset value: 0x0000 0000 (not reset by wakeup from Standby mode)

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															DPF1
															rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	DPF1	Deep-sleep1 mode status flag. This bit set by hardware when enter Deep-sleep 1 mode. And clear by software when write 0.

5.4.5. Control register 2 (PMU_CTL2)

Address offset: 0x10
Reset value: 0x0000 0000

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

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

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

位/位域	名称	描述
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}

5.4.6. Control register 3 (PMU_CTL3)

Address offset: 0x14

Reset value: 0x0000 003C

Bit0 to bit5 are reset by POR, and bit6 to bit15 are reset by system reset.

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
WUPPU9	WUPPU8	WUPPU7	WUPPU6	WUPPU5	WUPPU4	WUPPU3	WUPPU2	WUPPU1	WUPPU0	VAOVDVC[1:0]		OVDT[1:0]		VAOVDEN	OVDEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		rw	rw

位/位域	名称	描述
31:16	Reserved	Must be kept at reset value.。
15	WUPPU9	The WKUP pin (PD9) can be selected by pulling up and down 0: Pull down 1: Pull up
14	WUPPU8	The WKUP pin (PD4) can be selected by pulling up and down

		0: Pull down 1: Pull up
13	WUPPU7	The WKUP pin (PB9) can be selected by pulling up and down 0: Pull down 1: Pull up
12	WUPPU6	The WKUP pin (PB15) can be selected by pulling up and down 0: Pull down 1: Pull up
11	WUPPU5	The WKUP pin (PB5) can be selected by pulling up and down 0: Pull down 1: Pull up
10	WUPPU4	The WKUP pin (PC5) can be selected by pulling up and down 0: Pull down 1: Pull up
9	WUPPU3	The WKUP pin (PA2) can be selected by pulling up and down 0: Pull down 1: Pull up
8	WUPPU2	The WKUP pin (PE6) can be selected by pulling up and down 0: Pull down 1: Pull up
7	WUPPU1	The WKUP pin (PC13) can be selected by pulling up and down 0: Pull down 1: Pull up
6	WUPPU0	The WKUP pin (PA0) can be selected by pulling up and down 0: Pull down 1: Pull up
5:4	VAOVDVC[1:0]	V_{DDA} overvoltage threshold selection signal, monitoring V_{DDA} voltage 00: 3.65V threshold 01: 3.75V threshold 10: 3.85V threshold 11: 3.95V threshold
3:2	OVDT[1:0]	LVD overvoltage threshold selection signal, monitoring V_{DD} voltage 00: 3.65V threshold 01: 3.75V threshold 10: 3.85V threshold 11: 3.95V threshold
1	VAOVDEN	V_{DDA} overvoltage detection enable signal 0: The V_{DDA} overvoltage detection function is turned off

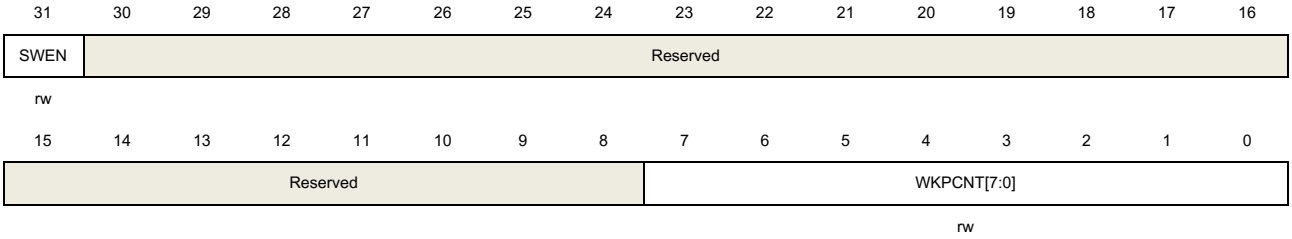
		1: The V_{DDA} overvoltage detection function is enabled
0	OVDEN	LVD overvoltage detection enable signal 0: The LVD overvoltage detection function is turned off 1: The LVD overvoltage detection function is enabled

5.4.7. Parameter register (PMU_PAR)

Address offset: 0x18

Reset value: 0x000A 0040

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



Bits	Fields	Descriptions
31	SWEN	Use configuration value when exit deep-sleep1 0: Use hardware default values. 1:Use software value, the value is set by WKPCNT[7:0] bit.
30:8	Reserved	Must be kept at reset value.
7:0	WKPCNT[7:0]	Wakeup time of deep-sleep1. 2 clock step and max value is PCLK1*512.

6. Reset and clock unit (RCU)

6.1. Reset control unit (RCTL)

6.1.1. Overview

GD32E53x 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.

6.1.2. 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. The reset service routine vector is fixed at address 0x0000_0004 in the memory map.

System reset

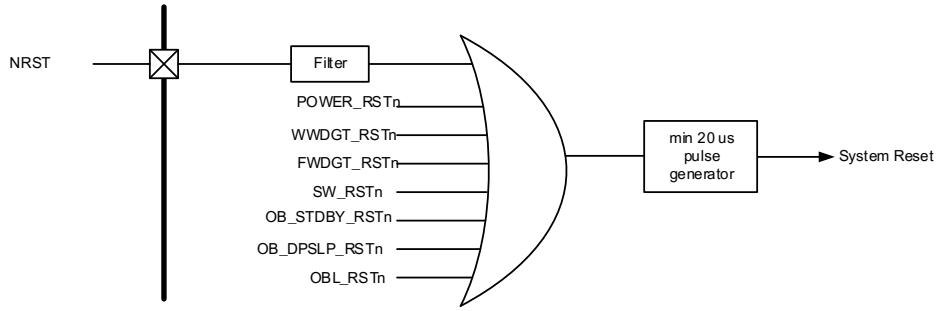
A system reset is generated by the following events:

- 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 6-1. The system reset circuit



Backup domain reset

A backup domain reset is generated when any of the following events occurs:

1. The BKPRST bit in the backup domain control register is set to 1.

2. A backup domain power-up reset occurs: after both V_{DD} and V_{BAT} have powered down, either V_{DD} or V_{BAT} powers up again.

6.2. Clock control unit (CCTL)

6.2.1. 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 (PLLs), a HXTAL clock 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 PLL. The maximum operating frequency of the system clock (CK_SYS) can be up to 252 MHz.

Figure 6-2. GD32E533 Clock tree

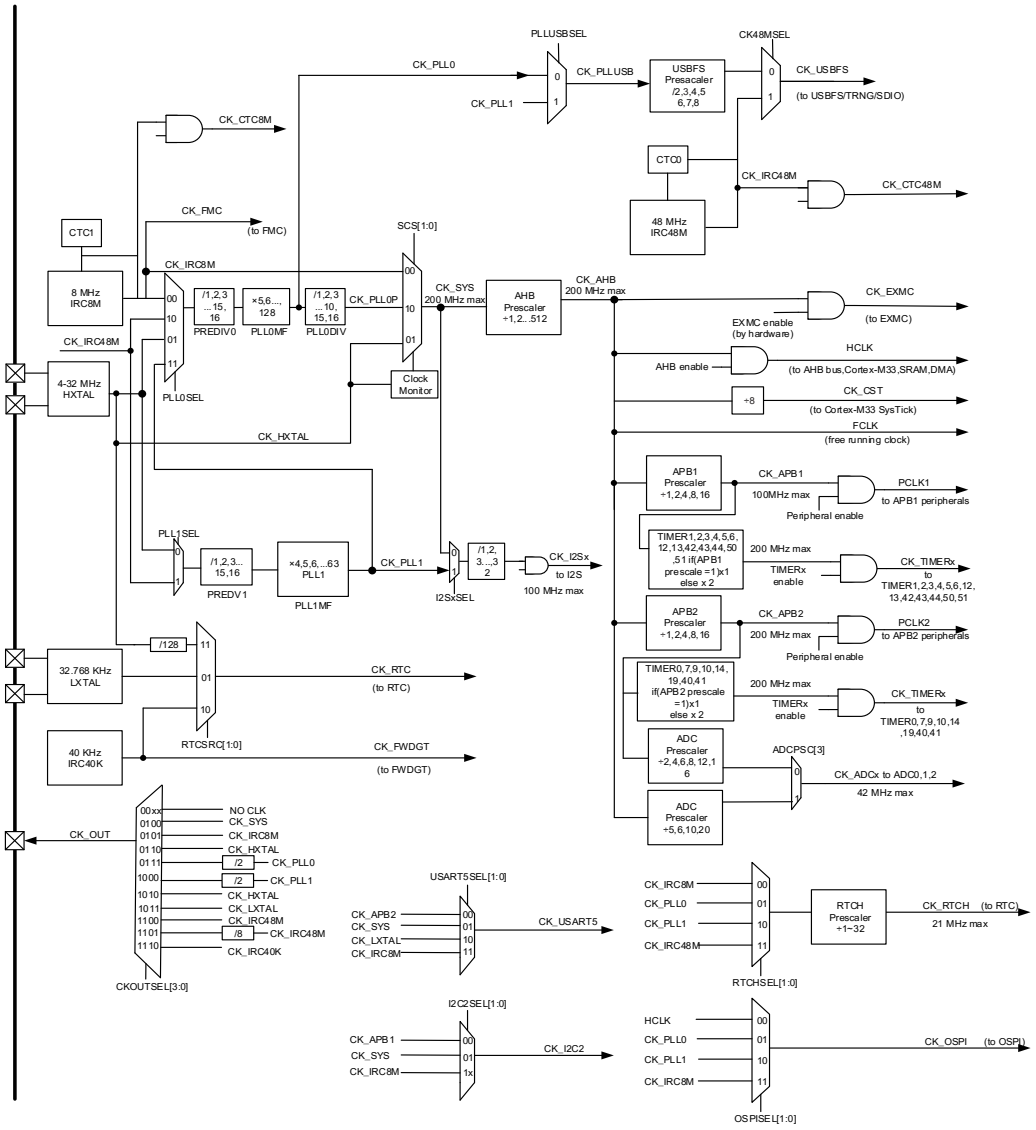
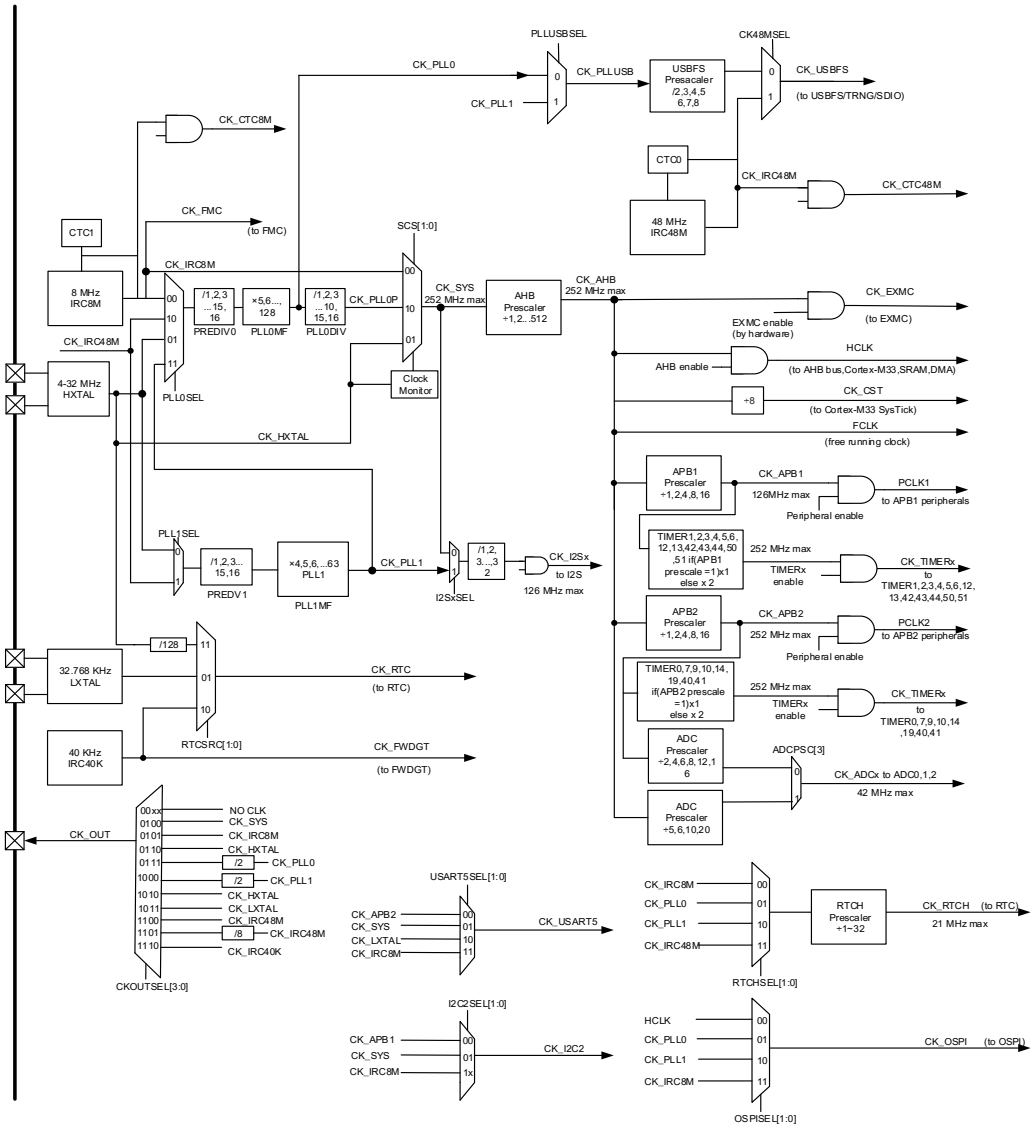


Figure 6-3. GD32E535 and GD32E53P Clock tree



The frequency of AHB, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB, APB2 and APB1 domains is 252 MHz / 252 MHz / 126 MHz. The Cortex® System Timer (SysTick) external clock is clocked with the AHB clock (HCLK) divided by 8. The SysTick can work either with this clock or with the AHB clock (HCLK), configurable in the systick control and status register.

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

The USART5 is clocked by IRC8M clock or LXTAL clock or CK_SYS or APB2 clock, which selected by USART5SEL bits in configuration register 2 (RCU_CFG2).

The I2C2 is clocked by IRC8M clock or CK_SYS or APB1 clock, which selected by I2C2SEL bits in configuration register 2 (RCU_CFG2).

The clock for EXMC is provided by CK_AHB.

The clock for USBFS, TRNG, and SDIO is supplied by CK_USBFS.

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 CTC0 is clocked by the clock of IRC48M. The IRC48M can be automatically trimmed by CTC0 unit.

The CTC1 is clocked by the clock of IRC8M. The IRC8M can be automatically trimmed by CTC1 unit.

The I2S is clocked by the clock of CK_SYS or CLK_PLL1.

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} 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.

The FMC is clocked by IRC8M clock, which is forced on when FMC started.

6.2.2. Characteristics

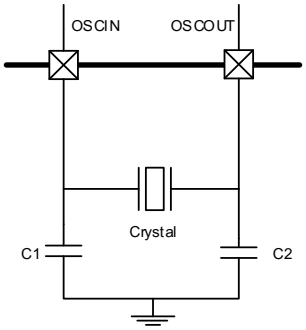
- 4 to 32 MHz high speed crystal oscillator (HXTAL).
- Internal 8 MHz RC oscillator (IRC8M).
- Internal 48 MHz RC oscillator (IRC48M).
- 32,768 Hz low speed crystal oscillator (LXTAL).
- Internal 40KHz RC oscillator (IRC40K).
- PLL0 clock source can be HXTAL, IRC8M, IRC48M or PLL1.
- HXTAL clock monitor.

6.2.3. Function overview

High speed crystal oscillator (HXTAL)

The high speed external crystal oscillator (HXTAL), which has a frequency from 4 to 32 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.

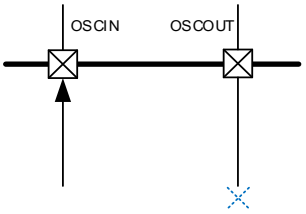
Figure 6-4. HXTAL clock source



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

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the control register RCU_CTL. During bypass mode, the signal is connected to OSCIN, and OSCOUT remains in the suspended state, as shown in ***Figure 6-5. HXTAL clock source in bypass mode***. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 6-5. 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 PLL input clock.

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

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

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 CTC0 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.

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 PLL) used as the CK_SYS, it is not possible to stop it.

HXTAL clock monitor (CKM)

The HXTAL clock monitor function is enabled by the HXTAL clock monitor enable bit, CKMEN, in the control register (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, CKMIF, 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 RTC, the HXTAL failure will reset the RTC clock selection.

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 6-1. Clock output 0 source select

Clock output source selection bits	Clock source
0100	CK_SYS
0101	CK_IRC8M
0110	CK_HXTAL
0111	CK_PLL0/2
1000	CK_PLL1/2
1010	CK_HXTAL
1011	CK_LXTAL
1100	CK_IRC48M
1101	CK_IRC48M/8
1110	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).

Clock frequency monitor (CKFM)

The clock frequency monitor can use IRC48M to monitor IRC8M, HXTAL, PLL0P, PLL1 and SYSCLK clock frequency range. For IRC8M and HXTAL, 1000 IRC48M clock cycle is used as the monitor window. For PLL0P, PLL1 and SYSCLK, 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, 4) 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.

Deep Sleep wakeup

Deep-sleep wakeup provides two modes: fast wakeup and slow wakeup, selectable via the DSLPWKMD bit in the RCU_DSV register.

Slow wakeup: When DSLPVSMD = 1 (user-configured core voltage), the DSLPVS[2:0] field determines the core voltage during deep sleep. If the core voltage is set to 0.85V or 0.9V in deep sleep, extra time is needed after wakeup for the LDO to ramp to 1.2V. In slow wakeup, IRC40K is enabled first; after waiting the period defined by IRC40KCNT[4:0], IRC8M is enabled and the system clock is switched to IRC8M.

Fast wakeup: This mode only supports a deep-sleep core voltage of 1.2V. There is no additional wait; IRC8M is enabled immediately and the system clock is switched to IRC8M. I2C/USART wakeup supports only the fast wakeup mode.

6.3. Register definition

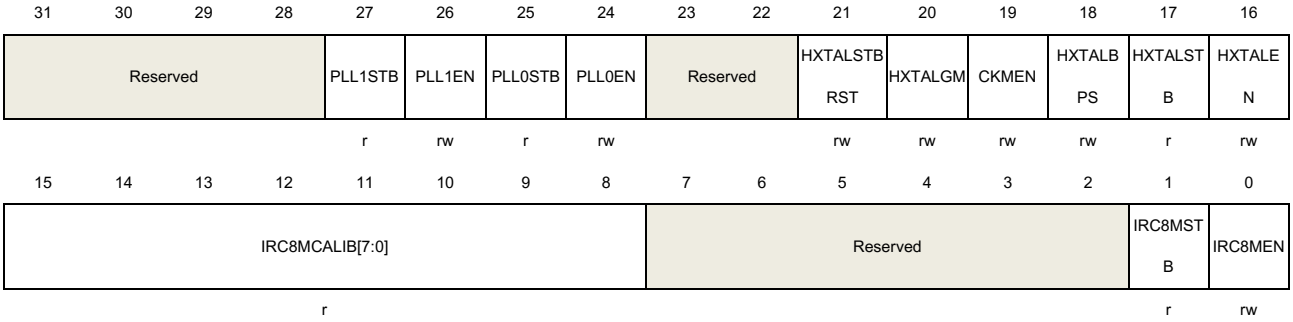
RCU base address: 0x4002 1000

6.3.1. Control register (RCU_CTL)

Address offset: 0x00

Reset value: 0x0000 xx03 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 Software set or reset, when the PLL clock is used as the system clock and PLL1 is used as the input source for the PLL, this bit cannot be reset. 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

23:22	Reserved	Must be kept at reset value.
21	HXTALSTBRST	Reset HXTALSTB bit. Set or reset by software, software control is used when HXTAL ready cannot be cleared to 0. 0: No reset 1: Force reset
20	HXTALGM	HXTAL gm control 0: HXTAL oscillation frequency is 4M - 10M 1: HXTAL oscillation frequency is 10M - 32M
19	CKMEN	HXTAL clock monitor enable 0: Disable the High speed 4 ~ 32 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the High speed 4 ~ 32 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.
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 ~ 32 MHz crystal oscillator disabled 1: High speed 4 ~ 32 MHz crystal oscillator enabled
15:8	IRC8MCALIB[7:0]	Internal 8MHz RC oscillator calibration value register These bits are load automatically at power on.
7:2	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

6.3.2. Clock configuration register 0 (RCU_CFG0)

Address offset: 0x04

Reset value: 0x4000 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
USBFSPSC[2]	PLL0MF[6:5]		ADCPSC[2]	PLL0MF[4]	CKOUTSEL[2:0]			USBFSPSC[1:0]		PLL0MF[3:0]				CKOUTSEL[3]	PLL1SEL
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
ADCPSC[1:0]		APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]	
rw		rw			rw			rw				r		rw	

Bits	Fields	Descriptions
31	USBFSPSC[2]	Bit 2 of USBFSPSC see bits 23:22 of RCU_CFG0
30:29	PLL0MF[6:5]	Bit 6 and Bit 5 of PLL0MF see bits 21:18 of RCU_CFG0
28	ADCPSC[2]	Bit 2 of ADCPSC see bits 15:14 of RCU_CFG0
27	PLL0MF[4]	Bit 4 of PLL0MF see bits 21:18 of RCU_CFG0
26:24	CKOUTSEL[2:0]	CKOUT clock source selection Set and reset by software. CKOUTSEL[3:0]: 00xx: No CLK 0100: CK_SYS 0101: CK_IRC8M 0110: CK_HXTAL 0111: CK_PLL0 / 2 1000: CK_PLL1 / 2

		1010: CK_HXTAL 1011: CK_LXTAL 1100: CK_IRC48M 1101: CK_IRC48M / 8 1110: CK_IRC40K
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 30, bit 29, bit 27 of RCU_CFG0 and these bits are written by software to define the PLL0 multiplication factor. Set and reset by software. **Note**: The PLL output clock frequency range is 32~480MHz VCO output clock frequency = PLL0 loop input clock frequency × PLLN (PLLN is the multiplication factor, its value = PLL0MF[6:0] + 1), and the following constraints must be met: When spread spectrum is enabled, PLL0MF[6:0] >= 11 must be satisfied (since the minimum value of MUL configuration is 4, the minimum value of PLL0MF is 4. In the spread spectrum algorithm, PLL0MF[6:0] may need to be reduced by 7, so the minimum value is 11). When spread spectrum is enabled, 122 >= PLL0MF must be satisfied (since PLL0MF configuration greater than 122 may cause overflow issues in the spread spectrum algorithm, the maximum value is 122). PLL0MF[6:0]: 0x00: PLL0 source clock x 5 0x01: PLL0 source clock x 5 0x02: PLL0 source clock x 5 0x03: PLL0 source clock x 5 0x04: PLL0 source clock x 5 0x05: PLL0 source clock x 6 0x06: PLL0 source clock x 7 0x07: PLL0 source clock x 8 …… 0x7E: PLL0 source clock x 127 0x7F: PLL0 source clock x 128
17	CKOUTSEL[3]	Bit 3 of CKOUTSEL

		see bits 24:26 of RCU_CFG0
16	PLL1SEL	PLL1 clock source selection Set and reset by software to control the PLL1 clock source. 00: HXTAL clock selected as source clock of PLL1 01: IRC48M clock selected as source clock of PLL1
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. ADCPSC[3:0]: 0000: CK_ADC = CK_APB2 / 2 0001: CK_ADC = CK_APB2 / 4 0010: CK_ADC = CK_APB2 / 6 0011: CK_ADC = CK_APB2 / 8 0100: CK_ADC = CK_APB2 / 2 0101: CK_ADC = CK_APB2 / 12 0110: CK_ADC = CK_APB2 / 8 0111: CK_ADC = CK_APB2 / 16 1x00: CK_ADC = CK_AHB / 5 1x01: CK_ADC = CK_AHB / 6 1x10: CK_ADC = CK_AHB / 10 1x11: CK_ADC = CK_AHB / 20
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

6.3.3. 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								CKMIC	LCKMIC	Reserved	PLL0 STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
								w	w		w	w	w	w	w

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	CKMIC	HXTAL clock stuck interrupt clear Write 1 by software to reset the CKMIF flag.

		0: Not reset CKMIF flag 1: Reset CKMIF flag
22	LCKMIC	LXTAL clock stuck interrupt clear Write 1 by software to reset the LCKMIF flag. 0: Not reset LCKMIF flag 1: Reset LCKMIF flag
21	Reserved	Must be kept at reset value.
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	Reserved	Must be kept at reset value.
14	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
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	CKMIF	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	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
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 ~ 32 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 LXTAL 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

6.3.4. 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
CMPRST	Reserved	CAN2RST	USART5RST	CAN1RST	TIMER19RST	TIMER41RST	TIMER14RST	SYSCFGRST	CAN0RST	TIMER10RST	TIMER9RST	TIMER40RST	TRIGSELRST	SDIORST	SPI3RST
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
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	CMPRST	CMP reset This bit is set and reset by software. 0: No reset

		1: Reset the CMP
30	Reserved	Must be kept at reset value.
29	CAN2RST	CAN2 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN2
28	USART5RST	USART5 reset This bit is set and reset by software. 0: No reset 1: Reset the USART5
27	CAN1RST	CAN1 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN1
26	TIMER19RST	TIMER19 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER19
25	TIMER41RST	TIMER41 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER41
24	TIMER14RST	TIMER14 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER14
23	SYSCFGRST	SYSCFG reset This bit is set and reset by software. 0: No reset 1: Reset the SYSCFG
22	CAN0RST	CAN0 reset This bit is set and reset by software. 0: No reset 1: Reset the CAN0
21	TIMER10RST	TIMER10 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER10
20	TIMER9RST	TIMER9 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER9

19	TIMER40RST	TIMER40 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER40
18	TRIGSELRST	TRIGSEL reset This bit is set and reset by software. 0: No reset 1: Reset the TRIGSEL
17	SDIORST	SDIO reset This bit is set and reset by software. 0: No reset 1: Reset the SDIO
16	SPI3RST	SPI3 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI3
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	TIMER7 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER7
12	SPI0RST	SPI0 reset This bit is set and reset by software. 0: No reset 1: Reset the SPI0
11	TIMER0RST	TIMER0 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER0
10	ADC1RST	ADC1 reset This bit is set and reset by software. 0: No reset

		1: Reset the ADC1
9	ADC0RST	ADC0 reset This bit is set and reset by software. 0: No reset 1: Reset the ADC0
8:7	Reserved	Must be kept at reset value.
6	PERST	GPIOE reset This bit is set and reset by software. 0: No reset 1: Reset the GPIOE
5	PDRST	GPIOD reset This bit is set and reset by software. 0: No reset 1: Reset the GPIOD
4	PCRST	GPIOC reset This bit is set and reset by software. 0: No reset 1: Reset the GPIOC
3	PBRST	GPIOB reset This bit is set and reset by software. 0: No reset 1: Reset the GPIOB
2	PARST	GPIOA reset This bit is set and reset by software. 0: No reset 1: Reset the GPIOA
1	Reserved	Must be kept at reset value.
0	AFRST	AFIO reset This bit is set and reset by software. 0: No reset 1: Reset the AFIO

6.3.5. 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	Reserved			I2C2RST	Reserved	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	Reserved
		rw	rw				rw		rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2RST	SPI1RST	TIMER51RST	TIMER50RST	WWDGTRST	TIMER44RST	TIMER43RST	TIMER13RST	TIMER12RST	TIMER42RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
rw	rw	rw	rw	rw	rw	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 the DAC
28	PMURST	PMU reset This bit is set and reset by software. 0: No reset 1: Reset the PMU
27:25	Reserved	Must be kept at reset value.
24	I2C2RST	I2C2 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C2
23	Reserved	Must be kept at reset value.
22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset 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	TIMER51RST	TIMER51 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER51
12	TIMER50RST	TIMER50 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER50
11	WWDGTRST	WWDGT reset This bit is set and reset by software. 0: No reset 1: Reset the WWDGT
10	TIMER44RST	TIMER44 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER44
9	TIMER43RST	TIMER43 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER43

8	TIMER13RST	TIMER13 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER13
7	TIMER12RST	TIMER12 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER12
6	TIMER42RST	TIMER42 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER42
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

6.3.6. AHB enable register (RCU_AHBEN)

Address offset: 0x14

Reset value: 0x0000 001C

Note: Bit 4 cannot be cleared to 0.

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
OSPIEN	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNGEN	HAUEN	CAUEN	USBFSEN	DMAMUXEN	Reserved		EXMCEN	Reserved	CRCEN	Reserved		BKSRAMSPEN	SRAMSPEN	DMA1EN	DMA0EN
rw	rw	rw	rw	rw			rw		rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	OSPIEN	OSPI clock enable This bit is set and reset by software. 0: Disabled OSPI clock 1: Enabled OSPI clock
30:16	Reserved	Must be kept at reset value.
15	TRNGEN	TRNG clock enable This bit is set and reset by software. 0: Disabled TRNG clock 1: Enabled TRNG clock
14	HAUEN	HAU clock enable This bit is set and reset by software. 0: Disabled HAU clock 1: Enabled HAU clock
13	CAUEN	CAU clock enable This bit is set and reset by software. 0: Disabled CAU clock 1: Enabled CAU clock
12	USBFSEN	USBFS clock enable This bit is set and reset by software. 0: Disabled USBFS clock 1: Enabled USBFS clock
11	DMAMUXEN	DMAMUX clock enable This bit is set and reset by software. 0: Disabled DMAMUX clock

		1: Enabled DMAMUX clock
10:9	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
5:4	Reserved	Must be kept at reset value.
3	BKSRAMSPEN	Back up SRAM clock enable when sleep mode This bit is set and reset by software. 0: Disabled back up SRAM clock during Sleep mode 1: Enabled back up SRAM clock during Sleep mode
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

6.3.7. 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
CMPEN	Reserved	CAN2EN	USART5EN	CAN1EN	TIMER19EN	TIMER41EN	TIMER14EN	SYSCFGEN	CAN0EN	TIMER10EN	TIMER9EN	TIMER40EN	TRIGSELEN	SDIOEN	SPI3EN

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
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	CMPEN	CMP clock enable This bit is set and reset by software. 0: Disabled CMP clock 1: Enabled CMP clock
30	Reserved	Must be kept at reset value.
29	CAN2EN	CAN2 clock enable This bit is set and reset by software. 0: Disabled CAN2 clock 1: Enabled CAN2 clock
28	USART5EN	USART5 clock enable This bit is set and reset by software. 0: Disabled USART5 clock 1: Enabled USART5 clock
27	CAN1EN	CAN1 clock enable This bit is set and reset by software. 0: Disabled CAN1 clock 1: Enabled CAN1 clock
26	TIMER19EN	TIMER19 clock enable This bit is set and reset by software. 0: Disabled TIMER19 clock 1: Enabled TIMER19 clock
25	TIMER41EN	TIMER41 clock enable This bit is set and reset by software. 0: Disabled TIMER41 clock 1: Enabled TIMER41 clock
24	TIMER14EN	TIMER14 clock enable This bit is set and reset by software. 0: Disabled TIMER14 clock 1: Enabled TIMER14 clock
23	SYSCFGEN	SYSCFG clock enable This bit is set and reset by software. 0: Disabled SYSCFG clock

		1: Enabled SYSCFG clock
22	CAN0EN	CAN0 clock enable This bit is set and reset by software. 0: Disabled CAN0 clock 1: Enabled CAN0 clock
21	TIMER10EN	TIMER10 clock enable This bit is set and reset by software. 0: Disabled TIMER10 clock 1: Enabled TIMER10 clock
20	TIMER9EN	TIMER9 clock enable This bit is set and reset by software. 0: Disabled TIMER9 clock 1: Enabled TIMER9 clock
19	TIMER40EN	TIMER40 clock enable This bit is set and reset by software. 0: Disabled TIMER40 clock 1: Enabled TIMER40 clock
18	TRIGSELEN	TRIGSEL clock enable This bit is set and reset by software. 0: Disabled TRIGSEL clock 1: Enabled TRIGSEL clock
17	SDIOEN	SDIO clock enable This bit is set and reset by software. 0: Disabled SDIO clock 1: Enabled SDIO clock
16	SPI3EN	SPI3 clock enable This bit is set and reset by software. 0: Disabled SPI3 clock 1: Enabled SPI3 clock
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

6.3.8. 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	Reserved			I2C2EN	Reserved	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	Reserved
		rw	rw				rw		rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2EN	SPI1EN	TIMER51EN	TIMER50EN	WWDGTEN	TIMER44EN	TIMER43EN	TIMER13EN	TIMER12EN	TIMER42EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN
rw	rw	rw	rw	rw	rw	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
27:25	Reserved	Must be kept at reset value.
24	I2C2EN	I2C2 clock enable This bit is set and reset by software. 0: Disabled I2C2 clock 1: Enabled I2C2 clock
~~23~~	Reserved	Must be kept at reset value.

22	I2C1EN	I2C1 clock enable This bit is set and reset by software. 0: Disabled I2C1 clock 1: Enabled I2C1 clock
21	I2C0EN	I2C0 clock enable This bit is set and reset by software. 0: Disabled I2C0 clock 1: Enabled I2C0 clock
20	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	TIMER51EN	TIMER51 clock enable This bit is set and reset by software. 0: Disabled TIMER51 clock 1: Enabled TIMER51 clock
12	TIMER50EN	TIMER50 clock enable

		This bit is set and reset by software. 0: Disabled TIMER50 clock 1: Enabled TIMER50 clock
11	WWDGTEN	WWDGT clock enable This bit is set and reset by software. 0: Disabled WWDGT clock 1: Enabled WWDGT clock
10	TIMER44EN	TIMER44 clock enable This bit is set and reset by software. 0: Disabled TIMER44 clock 1: Enabled TIMER44 clock
9	TIMER43EN	TIMER43 clock enable This bit is set and reset by software. 0: Disabled TIMER43 clock 1: Enabled TIMER43 clock
8	TIMER13EN	TIMER13 clock enable This bit is set and reset by software. 0: Disabled TIMER13 clock 1: Enabled TIMER13 clock
7	TIMER12EN	TIMER12 clock enable This bit is set and reset by software. 0: Disabled TIMER12 clock 1: Enabled TIMER12 clock
6	TIMER42EN	TIMER42 clock enable This bit is set and reset by software. 0: Disabled TIMER42 clock 1: Enabled TIMER42 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

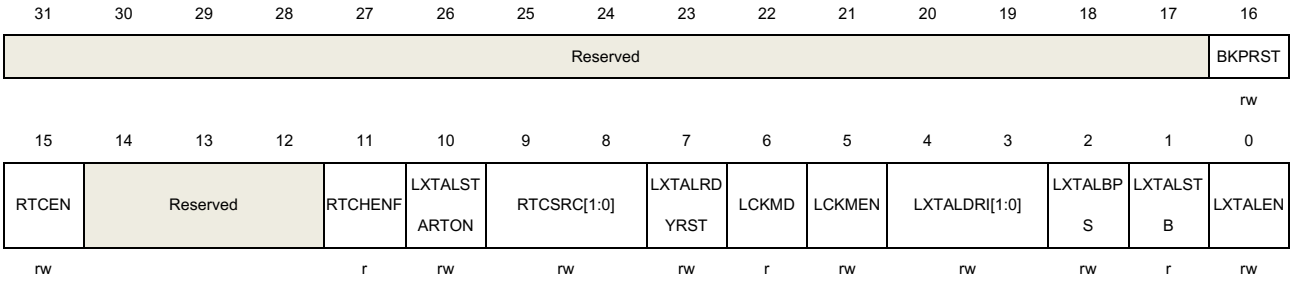
6.3.9. Backup domain control register (RCU_BDCTL)

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, LXTALRDYRST, LXTALSTARTON, LCKMEN, LXTALDRI, 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.



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
14:12	Reserved	Must be kept at reset value.
11	RTCHENF	RTC high frequency clock enable flag This bit indicates the status of the HRCE bit in the RTC_HRCR register. 0: RTCH clock disabled 1: RTCH clock enabled
10	LXTALSTARTON	LXTAL start circuit on control 0: Close the start circuit when LXTAL is ready 1: Hold the start circuit when LXTAL is ready
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

6.3.10. Reset source/clock register (RCU_RSTSCK)

Address offset: 0x24

Reset value: 0x0800 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
LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	BORRSTF	RSTFC	Reserved							
r	r	r	r	r	r	r	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IRC40KSTB	IRC40KEN
														r	rw

Bits	Fields	Descriptions
31	LPRSTF	Low-power reset flag Set by hardware when Deep-sleep /standby reset generated. Reset by writing 1 to the RSTFC bit. 0: No Low-power management reset generated 1: Low-power management reset generated
30	WWDGTRSTF	Window watchdog timer reset flag Set by hardware when a window watchdog timer reset generated. Reset by writing 1 to the RSTFC bit. 0: No window watchdog reset generated

		1: Window watchdog reset generated
29	FWDGTRSTF	Free watchdog timer reset flag Set by hardware when a free watchdog timer 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	BORRSTF	BOR reset flag Set by hardware when a BOR reset generated. Reset by writing 1 to the RSTFC bit. 0: No BOR reset generated 1: BOR reset generated
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

6.3.11. 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
OSPIRST	Reserved														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNGRST	HAURST	CAURST	USBFSRST	DMAMUXRST	Reserved		EXMCRST	Reserved	CRCRST	Reserved				DMA1RST	DMA0RST
rw	rw	rw	rw	rw			rw		rw					rw	rw

Bits	Fields	Descriptions
31	OSPIRST	OSPI reset This bit is set and reset by software. 0: No reset 1: Reset the OSPI
30:16	Reserved	Must be kept at reset value.
15	TRNGRST	TRNG reset This bit is set and reset by software. 0: No reset 1: Reset the TRNG
14	HAURST	HAU reset This bit is set and reset by software. 0: No reset 1: Reset the HAU
13	CAURST	CAU reset This bit is set and reset by software. 0: No reset 1: Reset the CAU
12	USBFSRST	USBFS reset This bit is set and reset by software. 0: No reset 1: Reset the USBFS
11	DMAMUXRST	DMAMUX reset This bit is set and reset by software.

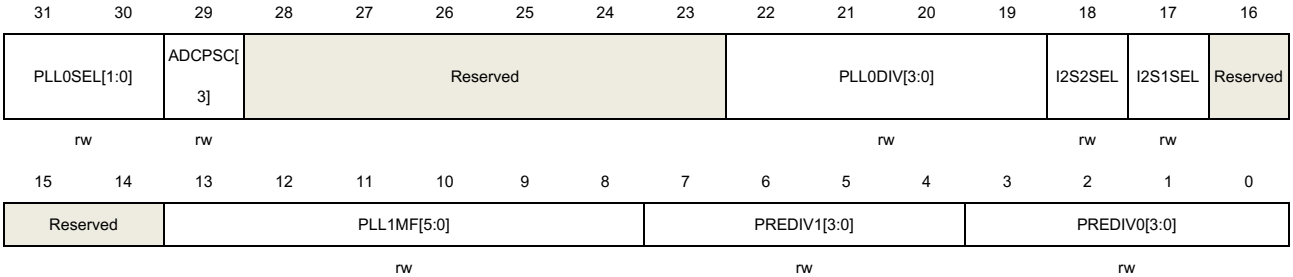
		0: No reset 1: Reset the DMAMUX
10:9	Reserved	Must be kept at reset value.
8	EXMCRST	EXMC reset This bit is set and reset by software. 0: No reset 1: Reset the EXMC
7	Reserved	Must be kept at reset value.
6	CRCRST	CRC reset This bit is set and reset by software. 0: No reset 1: Reset the CRC
5:2	Reserved	Must be kept at reset value.
1	DMA1RST	DMA0 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA0
0	DMA0RST	DMA1 reset This bit is set and reset by software. 0: No reset 1: Reset the DMA1

6.3.12. Clock configuration register 1 (RCU_CFG1)

Address offset: 0x2C

Reset value: 0x0000 2000

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



Bits	Fields	Descriptions
31:30	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
29	ADCPSC[3]	Bit 4 of ADCPSC see bits 15:14 of RCU_CFG0 and bit 28 of RCU_CFG0
28:23	Reserved	Must be kept at reset value.
22:19	PLL0DIV[3:0]	PLL0 divider The divider is PLL0DIV +1.
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
16:14	Reserved	Must be kept at reset value.
13:8	PLL1MF[5:0]	The PLL1 clock multiplication factor Set and reset by software. The frequency range of the VCO output clock after multiplication is 32~240MHz 000000: PLL1 source clock x 4 000001: PLL1 source clock x 4 000010: PLL1 source clock x 4 000011: PLL1 source clock x 4 000100: PLL1 source clock x 4 000101: PLL1 source clock x 5 000110: PLL1 source clock x 6 ..... 111110: PLL1 source clock x 62 111111: PLL1 source clock x 63
7:4	PREDIV1[3:0]	PREDIV1 division factor. The frequency range of the PLL1 loop input clock after division is 4~16MHz. 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 ..... 1110: PREDIV1 input source clock divided by 15

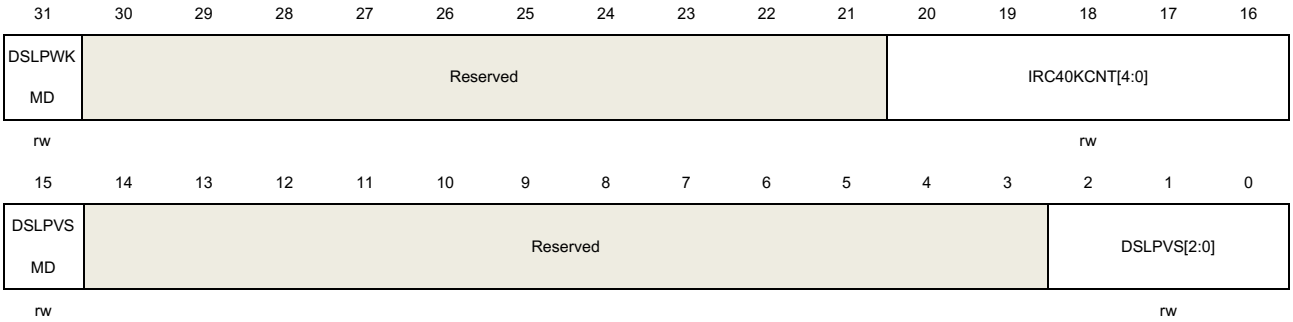
		1111: PREDIV1 input source clock divided by 16
3:0	PREDIV0[3:0]	PREDIV0 division factor. The frequency range of the PLL0 loop input clock after division is 4~16MHz. This bit is set and reset by software. These bits can be written when PLL0 is disable. 0000: PREDIV0 input source clock not divided 0001: PREDIV0 input source clock divided by 2 0010: PREDIV0 input source clock divided by 3 ..... 1110: PREDIV0 input source clock divided by 15 1111: PREDIV0 input source clock divided by 16

6.3.13. Deep-sleep mode voltage register (RCU_DSV)

Address offset: 0x34

Reset value: 0x8003 0000

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



Bits	Fields	Descriptions
31	DSLPWKMD	Deepsleep wake-up method. 0: Slow wake-up mode. Selecting this mode will wait for (IRC40KCNT -1) IRC40K clock cycles and supports voltage boosting under DSLPVS from 0.9V to 1.2V. 1: Fast wake-up mode. This mode only supports DSLPVS configured for core voltage at 1.2V for fast wake-up. The wake-up method for I2C/USART only supports fast wake-up mode.
30:21	Reserved	Must be kept at reset value.
20:16	IRC40KCNT[4:0]	After DeepSleep wake-up, the startup waiting time is forced to be configured as 3 if it is less than 3. The startup time is calculated as (IRC40KCNT - 1) * 25us, which equals 50us when configured as 3.
15	DSLPVSMD	Deep sleep mode voltage selection mode 0: Select system default value 1: Select DSLPVS[2:0] value

14:3	Reserved	Must be kept at reset value.
2:0	DSLPVS[2:0]	Deep sleep mode voltage selection These bits are set and cleared by software 000: Core voltage is 1.2V in deep sleep mode 001: Core voltage is 1.15V in deep sleep mode 010: Core voltage is 1.1V in deep sleep mode 011: Core voltage is 1.05V in deep sleep mode 100: Core voltage is 1.00V in deep sleep mode 101: Core voltage is 0.95V in deep sleep mode 110: Core voltage is 0.9V in deep sleep mode 111: Core voltage is 0.85V in deep sleep mode

6.3.14. 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

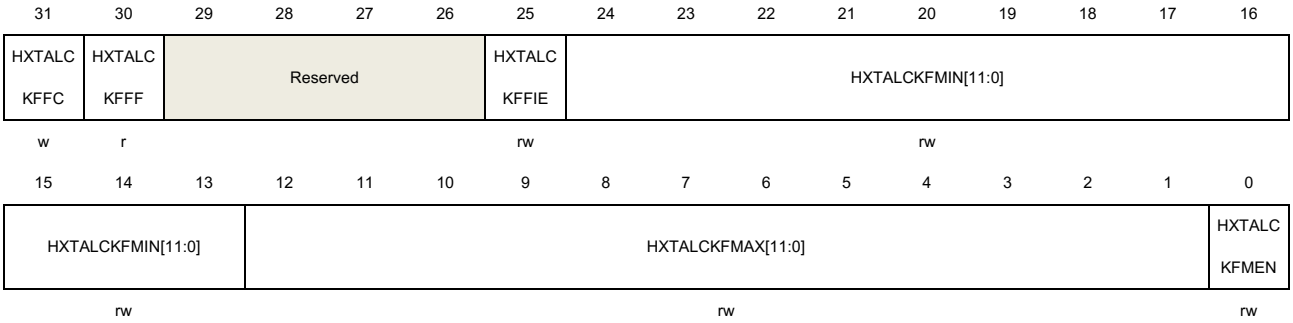
2:1	IRC8MCKFMC[1:0]	IRC8M clock frequency monitor configuration This bit is set and reset by software. 00: The clock frequency monitoring range is 8M Hz +/- 5% 01: The clock frequency monitoring range is 8M Hz +/- 10% 10: The clock frequency monitoring range is 8M Hz +/- 15% 11: The clock frequency monitoring 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

6.3.15. 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

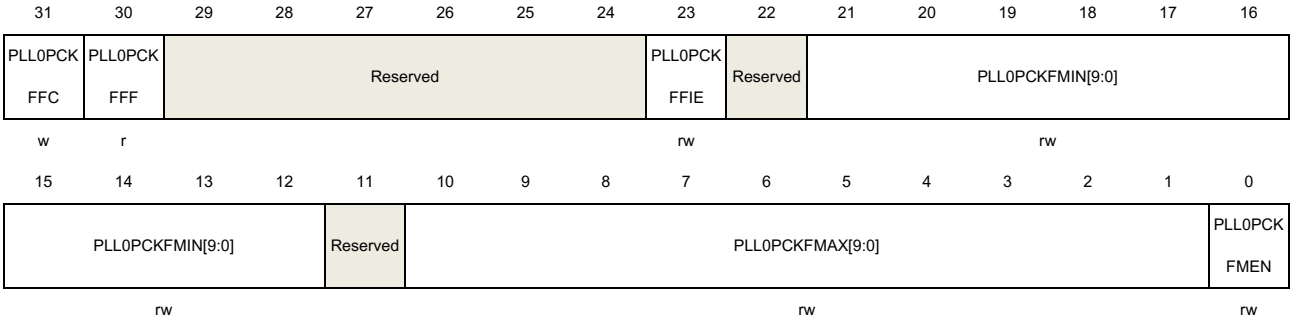
24:13	HXTALCKFMIN[11:0]	HXTAL clock frequency monitor configuration minimum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring window is 1000 IRC48M clock cycle.
12:1	HXTALCKFMAX[11:0]	HXTAL clock frequency monitor configuration maximum value This bit is set and 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

6.3.16. 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.

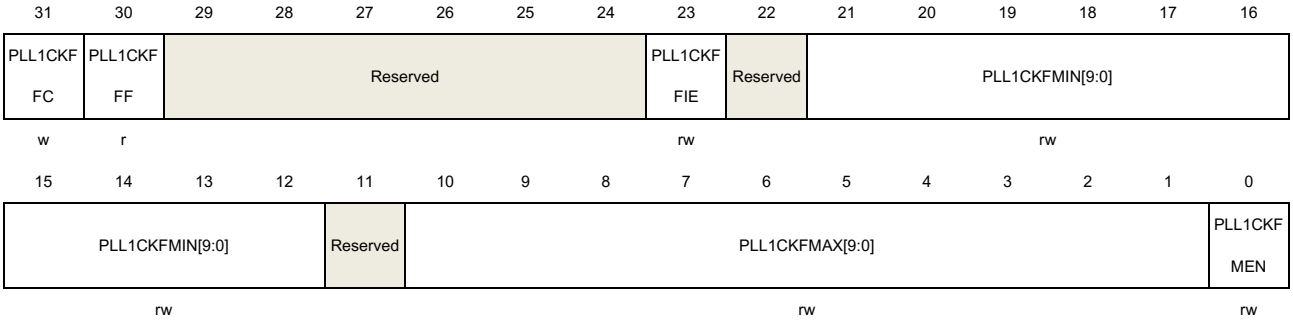
		0: disable PLL0P clock frequency failure interrupt 1: enable PLL0P clock frequency failure interrupt
22	Reserved	Must be kept at reset value.
21:12	PLL0PCKFMIN[9:0]	PLL0P clock frequency monitor configuration minimum value This bit is set and reset by software. The reference clock is IRC48M and a monitoring 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

6.3.17. 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

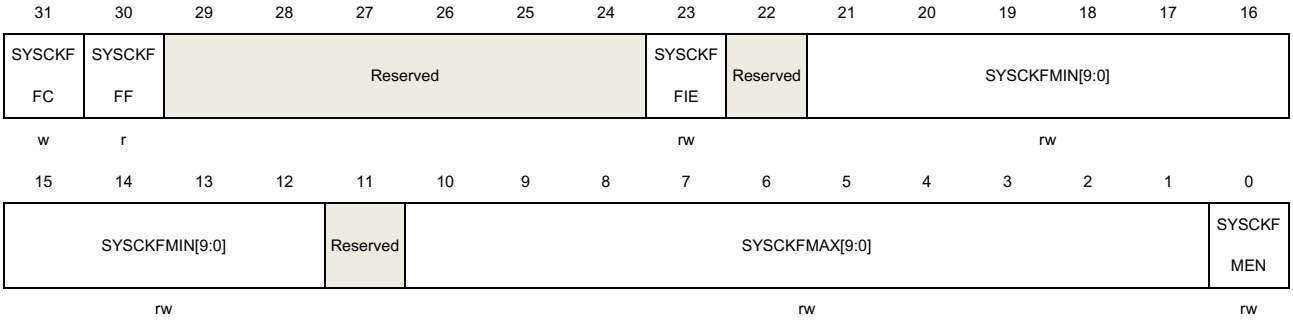
29:24	Reserved	Must be kept at reset value.
23	PLL1CKFFIE	PLL1 clock frequency failure interrupt enable This bit is set and reset by software. 0: disable PLL1 clock frequency failure interrupt 1: enable PLL1 clock frequency failure interrupt
22	Reserved	Must be kept at reset value.
21:12	PLL1CKFMIN[9:0]	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

6.3.18. Clock frequency monitor configuration register 4 (RCU_CKFMCFG4)

Address offset: 0x50

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	SYSCKFFC	SYSCLK clock frequency failure flag clear Write 1 by software to reset the SYSCKFFF flag and this bit is automatically reset by the hardware after SYSCKFFF is cleared. 0: Not reset SYSCKFFF flag 1: Reset SYSCKFFF flag
30	SYSCKFFF	SYSCLK clock frequency failure flag

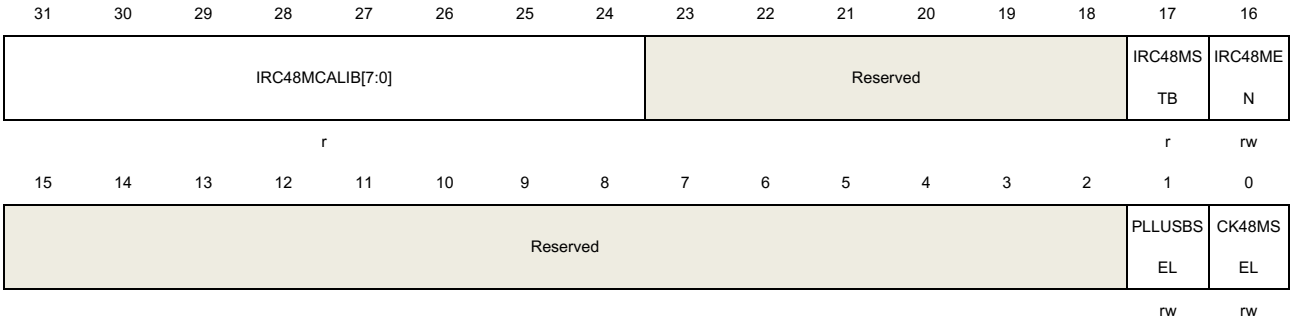
		This bit indicates whether PLL1 clock frequency is out of monitoring range. 0: SYSCLK clock frequency is not out of monitoring range 1: SYSCLK clock frequency is out of monitoring range
29:24	Reserved	Must be kept at reset value.
23	SYSCKFFIE	SYSCLK clock frequency failure interrupt enable This bit is set and reset by software. 0: disable SYSCLK clock frequency failure interrupt 1: enable SYSCLK clock frequency failure interrupt
22	Reserved	Must be kept at reset value.
21:12	SYSCKFMIN[9:0]	SYSCLK 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	SYSCKFMAX[9:0]	SYSCLK 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	SYSCKFMEN	SYSCLK clock frequency monitor enable This bit is set and reset by software. 0: Disable the SYSCLK clock frequency monitor 1: Enable the SYSCLK clock frequency monitor

6.3.19. 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).



Bits	Fields	Descriptions
31:24	IRC48MCALIB [7:0]	Internal 48MHz RC oscillator calibration value register These bits are load automatically at power on.

23:18	Reserved	Must be kept at reset value.
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
15:2	Reserved	Must be kept at reset value.
1	PLLUSBSEL	PLLUSB clock source selection 0: CK_PLL0 1: CK_PLL1
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_PLLUSB clock divided by USBFSPSC) 1: Select IRC48M clock

6.3.20. 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.

6.3.21. PLL clock spread spectrum control register (RCU_PLLSSCTL)

Address offset: 0xD0

Reset value: 0x0000 0000

The spread spectrum modulation is available only for the main PLL0 clock

The RCU_PLLSSCTL register must be written when the main PLL0 is disabled.

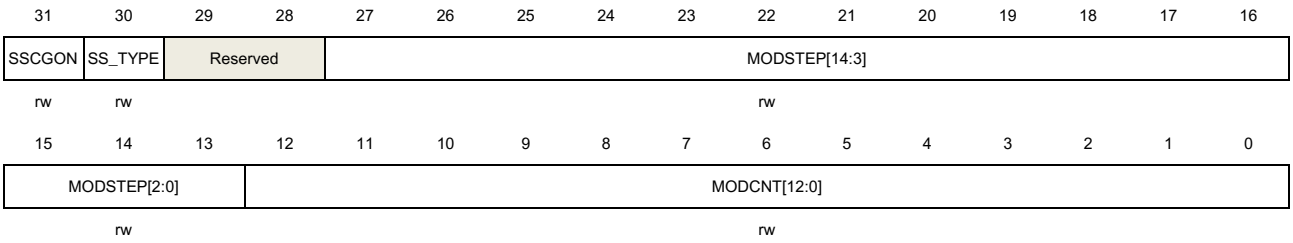
This register is used to configure the PLL0 spread spectrum clock generation according to the following formulas:

$MODCNT = round(f_{PLL0IN}/4/f_{mod})$

$MODSTEP = round(mdamp*PLL0N*2^{14}/(MODCNT*100))$

Where f_{PLL0IN} represents the PLL0 input clock frequency, f_{mod} represents the spread spectrum modulation frequency, mdamp represents the spread spectrum modulation amplitude expressed as a percentage, PLL0N represents the PLL0 clock frequency multiplication factor.

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



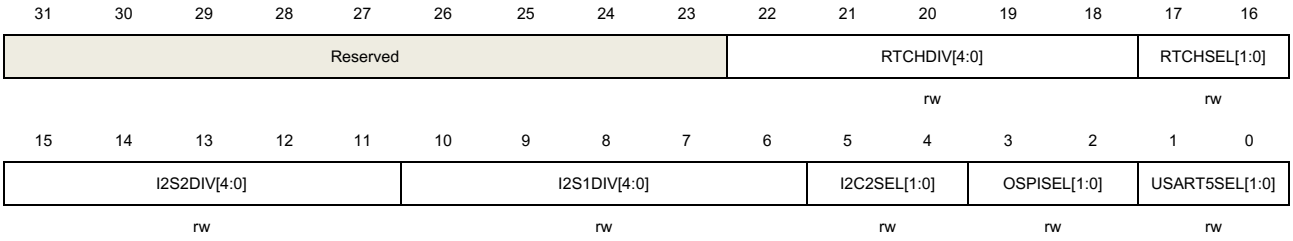
Bits	Fields	Descriptions
31	SSCGON	PLL0 spread spectrum modulation enable 0: Spread spectrum modulation disable

		1: Spread spectrum modulation enable
30	SS_TYPE	PLL0 spread spectrum modulation type select 0: Center spread selected 1: Down spread selected
29:28	Reserved	Must be kept at reset value.
27:13	MODSTEP[14:0]	These bits configure PLL0 spread spectrum modulation profile amplitude and frequency. The following criteria must be met: MODSTEP*MODCNT<=2^{15}-1
12:0	MODCNT[12:0]	These bits configure PLL0 spread spectrum modulation profile amplitude and frequency. The following criteria must be met: MODSTEP*MODCNT<=2^{15}-1

6.3.22. Clock configuration register 2 (RCU_CFG2)

Address offset: 0XD4
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22:18	RTCHDIV[4:0]	RTCH divider The divider is RTCHDIV +1.
17:16	RTCHSEL[1:0]	RTCH clock source selection Set and reset by software to control the RTCH clock source. 00: CK_IRC8M selected as RTCH source clock 01: CK_PLL0 selected as RTCH source clock 10: CK_PLL1 selected as RTCH source clock 11: CK_IRC48M selected as RTCH source clock
15:11	I2S2DIV[4:0]	I2S2DIV divider The divider is I2S2DIV +1.
10:6	I2S1DIV[4:0]	I2S1DIV divider The divider is I2S1DIV +1.
5:4	I2C2SEL[1:0]	I2C2 Clock Source Selection Set and reset by software to control the I2C2 clock source.

		00: APB1 clock selected as I2C2 source clock 01: System clock selected as I2C2 source clock 1x: CK_IRC8M clock selected as I2C2 source clock
3:2	OSPISEL[1:0]	OSPI clock source selection Set and reset by software to control the OSPI clock source. 00: HCLK selected as OSPI source clock 01: CK_PLL0 selected as OSPI source clock 10: CK_PLL1 selected as OSPI source clock 11: CK_IRC8M selected as OSPI source clock
1:0	USART5SEL[1:0]	USART5 clock source selection Set and reset by software to control the USART5 clock source. 00: CK_APB2 selected as USART5 source clock 01: CK_SYS selected as USART5 source clock 10: CK_LXTAL selected as USART5 source clock 11: CK_IRC8M selected as USART5 source clock

6.3.23. 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			CTC1RST	CTC0RST	Reserved										
			rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	CTC1RST	CTC1 reset This bit is set and reset by software. 0: No reset 1: Reset CTC1
27	CTC0RST	CTC0 reset This bit is set and reset by software. 0: No reset 1: Reset CTC0
26:0	Reserved	Must be kept at reset value.

6.3.24. 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			CTC1EN	CTC0EN	Reserved										
			rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	CTC1EN	CTC1 clock enable This bit is set and reset by software. 0: Disabled CTC1 clock 1: Enabled CTC1 clock
27	CTC0EN	CTC0 clock enable This bit is set and reset by software. 0: Disabled CTC0 clock 1: Enabled CTC0 clock
26:0	Reserved	Must be kept at reset value.

6.3.25. 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.

7. Clock trim controller (CTC)

7.1. Overview

The Clock Trim Controller (CTC) is used to trim internal 48MHz/8MHz RC oscillator (IRC48M/ IRC8M) automatically by hardware. The CTC unit trim the frequency of the IRC48M/ IRC8M based on an external accurate reference signal source. It can automatically adjust the trim value to provide a precise IRC48M/ IRC8M clock.

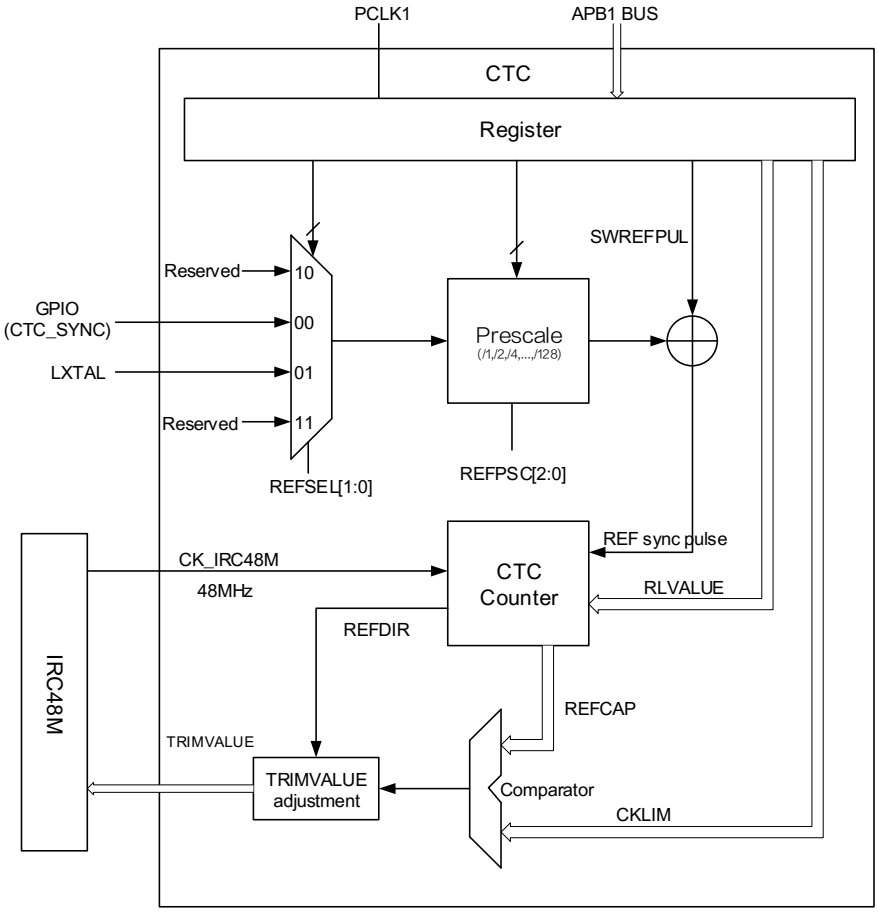
7.2. Characteristics

- Two external reference signal source: 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).

7.3. Function overview

Figure 7-1. CTC overview provides details on the internal configuration of the CTC.

Figure 7-1. CTC overview



7.3.1. REF sync pulse generator

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

Secondly, the selected reference signal source use 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.

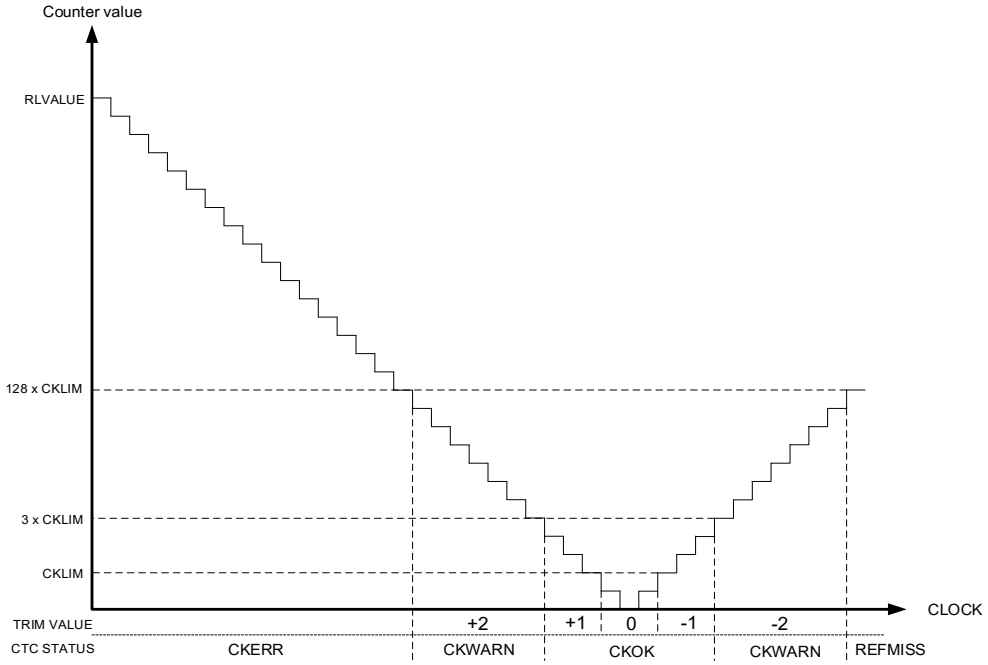
Note: The pulse width of the reference signal must be greater than 1.5 times the width of the clock being calibrated.

7.3.2. CTC trim counter

The CTC trim counter is clocked by CK_IRC48M/ CK_IRC8M. 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 7-2. CTC trim counter*.

Figure 7-2. CTC trim counter



7.3.3. 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/8M).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/8M).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.

7.3.4. 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, IRC8M:8 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:

$$RLVALUE = (F_{clock} / F_{REF}) - 1 \quad (7\text{-}1)$$

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

$$CKLIM = (F_{clock} / F_{REF}) \times 0.13\% / 2 \quad (7\text{-}2)$$

$$CKLIM = (F_{clock} / F_{REF}) \times 0.2\% / 2 \quad (7\text{-}3)$$

The typical step size is 0.13%(IRC48M) and 0.2%(IRC8M). Where the F_{clock} is the frequency of correct clock (IRC48M / IRC8M), the F_{REF} is the frequency of reference sync pulse.

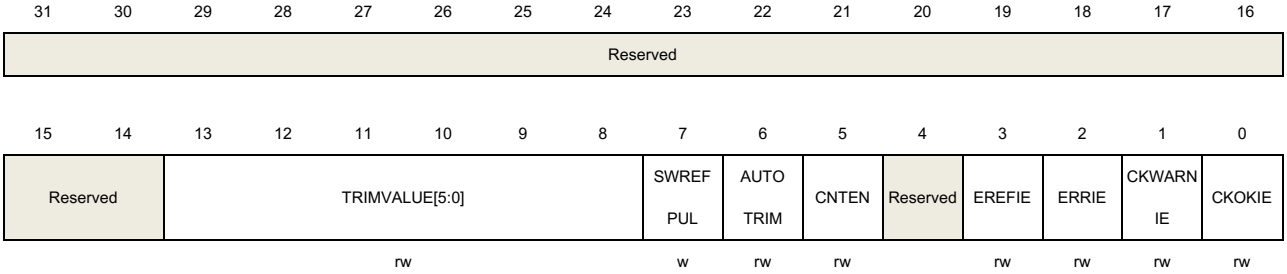
7.4. Register definition

CTC0 base address: 0x4000 C800
CTC1 base address: 0x4000 CC00

7.4.1. Control register 0 (CTC_CTL0)

Address offset: 0x00
Reset value: 0x0000 2000

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



Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	TRIMVALUE[5:0]	IRC48M / IRC8M 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 adjustment unit of IRC48M is 0.13%. The adjustment unit of IRC8M is 0.2%. The middle value is 32. When increase 1, the IRC48M / IRC8M clock frequency add around 0.13% / 0.2%. When decrease 1, the IRC48M / IRC8M clock frequency sub around 0.13% / 0.2%.
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	EREFIF interrupt enable 0: EREFIF interrupt disable 1: EREFIF interrupt enable
2	ERRIE	Error (ERRIF) interrupt enable 0: ERRIF interrupt disable 1: ERRIF interrupt enable
1	CKWARNIE	Clock trim warning (CKWARNIF) interrupt enable 0: CKWARNIF interrupt disable 1: CKWARNIF interrupt enable
0	CKOKIE	Clock trim OK (CKOKIF) interrupt enable 0: CKOKIF interrupt disable 1: CKOKIF interrupt enable

7.4.2. Control register 1 (CTC_CTL1)

Address offset: 0x04
CTC0 Reset value：0x2022 BB7F
CTC1 Reset value：0x2009 1F3F

This register has to be accessed by word (32-bit).
Note: This register cannot be modified when CNTEN is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFPOL	Reserved	REFSEL[1:0]		Reserved	REFPSC[2:0]			CKLIM[7:0]							
rw		rw			rw			rw							

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

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.

7.4.3. 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				TRIM ERR	REF MISS	CKERR	Reserved				EREFIF	ERRIF	CKWARN IF	CKOK IF
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 occur. 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 occur. 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 occur. 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 occur. 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 occur. 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 occur. 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, an interrupt occur. 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

7.4.4. 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.

8. Interrupt/event controller (EXTI)

8.1. Overview

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

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

8.2. Characteristics

- Cortex®-M33 system exception.
- Up to 115 maskable peripheral interrupts.
- 4 bits interrupt priority configuration-16 priority levels.
- Efficient interrupt processing.
- Support exception pre-emption and tail-chaining.
- Wake up system from power saving mode.
- Up to 25 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

8.3. 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. The *Table 8-1. NVIC exception types in Cortex®-M33* and *Table 8-2. Interrupt vector table* list all exception types.

Table 8-1. 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 8-2. Interrupt vector table

Interrupt number	Vector number	Interrupt description	Interrupt address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	VAUVD /VAOVD / LVD / OVD detection interrupt from EXTI16	0x0000_0044
IRQ 2	18	RTC tamper / timestamp interrupt from EXTI24	0x0000_0048
IRQ 3	19	RTC wakeup interrupt from EXTI19	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU and CTC interrupt	0x0000_0054
IRQ 6	22	EXTI line 0 interrupt	0x0000_0058
IRQ 7	23	EXTI line 1 interrupt	0x0000_005C
IRQ 8	24	EXTI line 2 interrupt	0x0000_0060
IRQ 9	25	EXTI line 3 interrupt	0x0000_0064
IRQ 10	26	EXTI line 4 interrupt	0x0000_0068
IRQ 11	27	DMA0 channel 0 global interrupt	0x0000_006C
IRQ 12	28	DMA0 channel 1 global interrupt	0x0000_0070
IRQ 13	29	DMA0 channel 2 global interrupt	0x0000_0074
IRQ 14	30	DMA0 channel 3 global interrupt	0x0000_0078
IRQ 15	31	DMA0 channel 4 global interrupt	0x0000_007C
IRQ 16	32	DMA0 channel 5 global interrupt	0x0000_0080

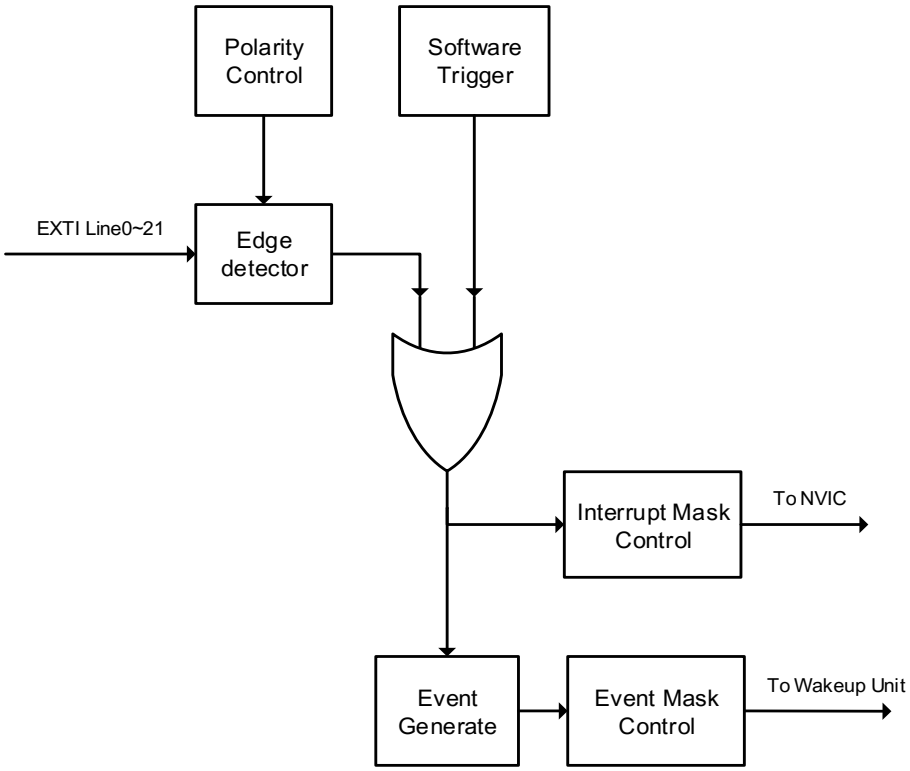
Interrupt number	Vector number	Interrupt description	Interrupt address
IRQ 17	33	DMA0 channel 6 global interrupt	0x0000_0084
IRQ 18	34	ADC0 and ADC1 global interrupt	0x0000_0088
IRQ 19	35	CAN0 transmit interrupt	0x0000_008C
IRQ 20	36	CAN0 receive 0 interrupt	0x0000_0090
IRQ 21	37	CAN0 receive 1 interrupt	0x0000_0094
IRQ 22	38	CAN0 EWMC interrupt	0x0000_0098
IRQ 23	39	EXTI line [9:5] interrupt	0x0000_009C
IRQ 24	40	TIMER0 break interrupt and TIMER14 global interrupt	0x0000_00A0
IRQ 25	41	TIMER0 update interrupt and TIMER9 global interrupt	0x0000_00A4
IRQ 26	42	TIMER0 trigger and commutation interrupt and TIMER10 global interrupt	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 or I2S1ADD 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] interrupt	0x0000_00E0
IRQ 41	57	RTC Alarm interrupt EXTI17	0x0000_00E4
IRQ 42	58	USBFS wakeup interrupt EXTI18	0x0000_00E8
IRQ 43	59	TIMER7 break interrupt	0x0000_00EC
IRQ 44	60	TIMER7 update interrupt and TIMER12 global interrupt	0x0000_00F0
IRQ 45	61	TIMER7 trigger and commutation interrupt and TIMER13 global interrupt	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	SDIO global interrupt	0x0000_0104
IRQ50	66	TIMER4 global interrupt	0x0000_0108
IRQ51	67	SPI2 or I2S2ADD global interrupt	0x0000_010C
IRQ52	68	UART3 global interrupt	0x0000_0110
IRQ53	69	UART4 global interrupt	0x0000_0114
IRQ54	70	TIMER5 or DAC global interrupt	0x0000_0118

Interrupt number	Vector number	Interrupt description	Interrupt address
IRQ55	71	TIMER6 global interrupt	0x0000_011C
IRQ56	72	DMA1 channel 0 global interrupt	0x0000_0120
IRQ57	73	DMA1 channel 1 global interrupt	0x0000_0124
IRQ58	74	DMA1 channel 2 global interrupt	0x0000_0128
IRQ59	75	DMA1 channel 3 global interrupt	0x0000_012C
IRQ60	76	DMA1 channel 4 global interrupt	0x0000_0130
IRQ61	77	DMAMUX global interrupt	0x0000_0134
IRQ62	78	Reserved	0x0000_0138
IRQ63	79	CAN1 transmit interrupt	0x0000_013C
IRQ64	80	CAN1 receive 0 interrupt	0x0000_0140
IRQ65	81	CAN1 receive 1 interrupt	0x0000_0144
IRQ66	82	CAN1 EWMC interrupt	0x0000_0148
IRQ67	83	USBFS global interrupt	0x0000_014C
IRQ68	84	Reserved	0x0000_0150
IRQ69	85	Reserved	0x0000_0154
IRQ70	86	Reserved	0x0000_0158
IRQ71	87	Reserved	0x0000_015C
IRQ72	88	Reserved	0x0000_0160
IRQ73	89	Reserved	0x0000_0164
IRQ74	90	Reserved	0x0000_0168
IRQ75	91	Reserved	0x0000_016C
IRQ76	92	Reserved	0x0000_0170
IRQ77	93	Reserved	0x0000_0174
IRQ78	94	CAN2 transmit interrupt	0x0000_0178
IRQ79	95	CAN2 receive 0 interrupt	0x0000_017C
IRQ80	96	CAN2 receive 1 interrupt	0x0000_0180
IRQ81	97	CAN2 EWMC interrupt	0x0000_0184
IRQ82	98	I2C2 Event interrupt	0x0000_0188
IRQ83	99	I2C2 Error interrupt	0x0000_018C
IRQ84	100	USART5 global interrupt	0x0000_0190
IRQ85	101	I2C2 wakeup interrupt EXTI20	0x0000_0194
IRQ86	102	USART5 wakeup interrupt EXTI21	0x0000_0198
IRQ87	103	FPU interrupt	0x0000_019C
IRQ88	104	TRNG global interrupt	0x0000_01A0
IRQ89	105	CAU global interrupt	0x0000_01A4
IRQ90	106	Timer40 global interrupt	0x0000_01A8
IRQ91	107	Timer41 global interrupt	0x0000_01AC
IRQ92	108	Timer42 global interrupt	0x0000_01B0
IRQ93	109	Timer43 global interrupt	0x0000_01B4
IRQ94	110	Timer44 global interrupt	0x0000_01B8

Interrupt number	Vector number	Interrupt description	Interrupt address
IRQ95	111	Timer50 global interrupt	0x0000_01BC
IRQ96	112	Timer51 global interrupt	0x0000_01C0
IRQ97	113	ICACHE global interrupt	0x0000_01C4
IRQ98	114	DCACHE global interrupt	0x0000_01C8
IRQ99	115	HAU interrupt	0x0000_01CC
IRQ100	116	OSPI interrupt	0x0000_01D0
IRQ101	117	SPI3 interrupt	0x0000_01D4
IRQ102	118	VOVD/VUVD interrupt	0x0000_01D8
IRQ103	119	TIMER0 channel break interrupt	0x0000_01DC
IRQ104	120	TIMER7 channel break interrupt	0x0000_01E0
IRQ105	121	SYSCFG interrupt	0x0000_01E4
IRQ106	122	RCU clock frequency monitor interrupt	0x0000_01E8
IRQ107	123	TIMER19 break interrupt	0x0000_01EC
IRQ108	124	TIMER19 update interrupt	0x0000_01F0
IRQ109	125	TIMER19 trigger and commutation interrupt	0x0000_01F4
IRQ110	126	TIMER19 channel capture compare interrupt	0x0000_01F8
IRQ111	127	TIMER19 channel break interrupt	0x0000_01FC
IRQ112	128	CMP interrupt	0x0000_0200
IRQ113	129	CMP0 wakeup EXTI22	0x0000_0204
IRQ114	130	CMP1 wakeup EXTI23	0x0000_0208

8.4. External interrupt and event (EXTI) block diagram

Figure 8-1. Block diagram of EXTI



8.5. External Interrupt and event function overview

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

The EXTI trigger source includes 16 external lines from GPIO pins and 9 lines from internal modules which refers to ***Table 8-3. EXTI source*** for detail. 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.

8.5.1. 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.

8.5.2. 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 8-3. 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 / 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

EXTI Line Number	Source
16	VAUVD / VAOVD / LVD / OVD
17	RTC Alarm
18	USB wakeup
19	RTC wakeup
20	I2C2 wakeup
21	USART5 wakeup
22	CMP0 wakeup
23	CMP1 wakeup
24	RTC Tamper / Timestamp

8.6. EXTI Register

EXTI base address: 0x4001 0400

8.6.1. 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
保留							INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

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

8.6.2. 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
保留							EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

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

8.6.3. 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
保留							RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
							rw	rw	w	rw	w	rw	rw	rw	rw

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

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

8.6.4. 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
保留							FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

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

8.6.5. 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
保留							SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

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

8.6.6. Pending register (EXTI_PD)

Address offset: 0x14

Reset value: 0xXXXX XXXX

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

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

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

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

9. Trigger selection controller (TRIGSEL)

9.1. Overview

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

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

9.2. Characteristics

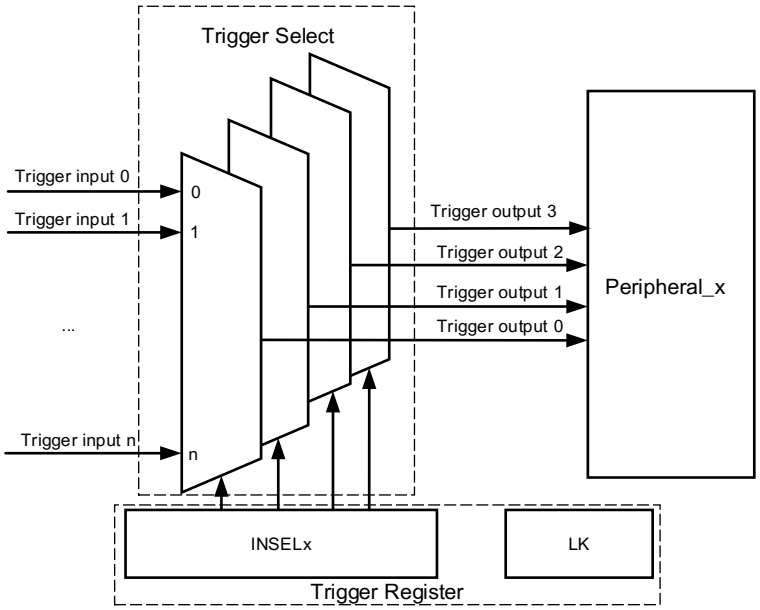
- Support up to 125 trigger input signals.
- Each peripheral has its corresponding register to select trigger input signal.
- TRIGSEL register can be configured up to 4 outputs to peripheral.
- Trigger input source could be external input signal or output of peripheral.
- Trigger selection output could be for external output or peripheral input.

9.3. Function overview

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

The *Figure 9-1. TRIGSEL main composition example* shows the main composition of TRIGSEL.

Figure 9-1. TRIGSEL main composition example



9.4. Internal connections

The TRIGSEL allows software to select the trigger input for peripherals. The *Table 9-1. Trigger input bit fields selection* shows the trigger input register selection.

Table 9-1. 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
	0x0d	TIMER0_CH2
	0x0e	TIMER0_CH3
	0x0f	TIMER7_TRGO
	0x10	TIMER7_CH0

fields	bits value	trigger input selection
	0x11	TIMER7_CH1
	0x12	TIMER7_CH2
	0x13	TIMER7_CH3
	0x14	TIMER19_TRGO
	0x15	TIMER19_CH0
	0x16	TIMER19_CH1
	0x17	TIMER19_CH2
	0x18	TIMER19_CH3
	0x19	TIMER5_TRGO
	0x1a	TIMER6_TRGO
	0x1b	TIMER50_TRGO
	0x1c	TIMER51_TRGO
	0x1d	TIMER1_TRGO
	0x1e	TIMER1_CH0
	0x1f	TIMER1_CH1
	0x20	TIMER1_CH2
	0x21	TIMER1_CH3
	0x22	TIMER2_TRGO
	0x23	TIMER2_CH0
	0x24	TIMER2_CH1
	0x25	TIMER2_CH2
	0x26	TIMER2_CH3
	0x27	TIMER3_TRGO
	0x28	TIMER3_CH0
	0x29	TIMER3_CH1
	0x2a	TIMER3_CH2
	0x2b	TIMER3_CH3
	0x2c	TIMER4_TRGO
	0x2d	TIMER4_CH0
	0x2e	TIMER4_CH1
	0x2f	TIMER4_CH2
	0x30	TIMER4_CH3
	0x31	TIMER9_TRGO
	0x32	TIMER9_CH0
	0x33	TIMER10_TRGO
	0x34	TIMER10_CH0
	0x35	TIMER12_TRGO
	0x36	TIMER12_CH0
	0x37	TIMER13_TRGO

fields	bits value	trigger input selection
	0x38	TIMER13_CH0
	0x39	TIMER14_TRGO
	0x3a	TIMER14_CH0
	0x3b	TIMER14_CH1
	0x3c	TIMER14_MCH0
	0x3d	TIMER40_TRGO
	0x3e	TIMER40_CH0
	0x3f	TIMER40_CH1
	0x40	TIMER40_MCH0
	0x41	TIMER41_TRGO
	0x42	TIMER41_CH0
	0x43	TIMER41_CH1
	0x44	TIMER41_MCH0
	0x45	TIMER42_TRGO
	0x46	TIMER42_CH0
	0x47	TIMER42_CH1
	0x48	TIMER42_MCH0
	0x49	TIMER43_TRGO
	0x4a	TIMER43_CH0
	0x4b	TIMER43_CH1
	0x4c	TIMER43_MCH0
	0x4d	TIMER44_TRGO
	0x4e	TIMER44_CH0
	0x4f	TIMER44_CH1
	0x50	TIMER44_MCH0
	0x51	ADC0_WD0_OUT_H
	0x52	ADC0_WD0_OUT_L
	0x53	ADC0_WD1_OUT_H
	0x54	ADC0_WD1_OUT_L
	0x55	ADC0_WD2_OUT_H
	0x56	ADC0_WD2_OUT_L
	0x57	ADC1_WD0_OUT_H
	0x58	ADC1_WD0_OUT_L
	0x59	ADC1_WD1_OUT_H
	0x5a	ADC1_WD1_OUT_L
	0x5b	ADC1_WD2_OUT_H
	0x5c	ADC1_WD2_OUT_L
	0x5d	ADC2_WD0_OUT_H
	0x5e	ADC2_WD0_OUT_L

fields	bits value	trigger input selection
	0x5f	ADC2_WD1_OUT_H
	0x60	ADC2_WD1_OUT_L
	0x61	ADC2_WD2_OUT_H
	0x62	ADC2_WD2_OUT_L
	0x63	CMP0_OUT
	0x64	CMP1_OUT
	0x65	CK_OUT
	0x66	TIMER0_BRKIN
	0x67	TIMER0_CH0BRKIN
	0x68	TIMER0_CH1BRKIN
	0x69	TIMER0_CH2BRKIN
	0x6a	TIMER7_BRKIN
	0x6b	TIMER7_CH0BRKIN
	0x6c	TIMER7_CH1BRKIN
	0x6d	TIMER7_CH2BRKIN
	0x6e	TIMER19_BRKIN
	0x6f	TIMER19_CH0BRKIN
	0x70	TIMER19_CH1BRKIN
	0x71	TIMER19_CH2BRKIN
	0x72	TIMER14_BRKIN
	0x73	TIMER40_BRKIN
	0x74	TIMER41_BRKIN
	0x75	TIMER42_BRKIN
	0x76	TIMER43_BRKIN
	0x77	TIMER44_BRKIN
	0x78	EXTI9
	0x79	EXTI11
	0x7a	EXTI15
	0x7b	RTC_Alarm
	0x7c	RTC_TPTS

Table 9-2. 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 4 outputs, which are connected to the corresponding peripherals.

Table 9-2. TRIGSEL input and output mapping

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
VSS VDD TRIGSEL_IN0 TRIGSEL_IN1 TRIGSEL_IN2 TRIGSEL_IN3 TRIGSEL_IN4 TRIGSEL_IN5 TRIGSEL_IN6 TRIGSEL_IN7 TIMER0_TRGO TIMER0_CH0 TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER7_TRGO TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3 TIMER19_TRGO TIMER19_CH0 TIMER19_CH1 TIMER19_CH2 TIMER19_CH3 TIMER5_TRGO TIMER6_TRGO TIMER50_TRGO TIMER51_TRGO TIMER1_TRGO TIMER1_CH0 TIMER1_CH1 TIMER1_CH2 TIMER1_CH3 TIMER2_TRGO TIMER2_CH0	INSELx[7:0]	TRIGSEL_EXTOUT_0	Output0 Output1	TRIGSEL_OUT0 TRIGSEL_OUT1
		TRIGSEL_EXTOUT_1	Output0 Output1	TRIGSEL_OUT2 TRIGSEL_OUT3
		TRIGSEL_EXTOUT_2	Output0 Output1	TRIGSEL_OUT4 TRIGSEL_OUT5
		TRIGSEL_EXTOUT_3	Output0 Output1	TRIGSEL_OUT6 TRIGSEL_OUT7
		TRIGSEL_TIMER0ITI	Output0	TIMER0_ITI
		TRIGSEL_TIMER1ITI	Output0	TIMER1_ITI
		TRIGSEL_TIMER2ITI	Output0	TIMER2_ITI
		TRIGSEL_TIMER3ITI	Output0	TIMER3_ITI
		TRIGSEL_TIMER4ITI	Output0	TIMER4_ITI

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER2_CH1 TIMER2_CH2 TIMER2_CH3 TIMER3_TRGO TIMER3_CH0 TIMER3_CH1 TIMER3_CH2 TIMER3_CH3 TIMER4_TRGO TIMER4_CH0 TIMER4_CH1 TIMER4_CH2 TIMER4_CH3 TIMER9_TRGO TIMER9_CH0 TIMER10_TRGO TIMER10_CH0 TIMER12_TRGO TIMER12_CH0 TIMER13_TRGO TIMER13_CH0 TIMER14_TRGO TIMER14_CH0 TIMER14_CH1 TIMER14_MCH0 TIMER40_TRGO TIMER40_CH0 TIMER40_CH1 TIMER40_MCH0 TIMER41_TRGO TIMER41_CH0 TIMER41_CH1 TIMER41_MCH0 TIMER42_TRGO TIMER42_CH0 TIMER42_CH1 TIMER42_MCH0 TIMER43_TRGO TIMER43_CH0 TIMER43_CH1		TRIGSEL_TIMER7ITI	Output0	TIMER7_ITI
		TRIGSEL_TIMER14ITI	Output0	TIMER14_ITI
		TRIGSEL_TIMER19ITI	Output0	TIMER19_ITI
		TRIGSEL_TIMER40ITI	Output0	TIMER40_ITI
		TRIGSEL_TIMER41ITI	Output0	TIMER41_ITI
		TRIGSEL_TIMER42ITI	Output0	TIMER42_ITI
		TRIGSEL_TIMER43ITI	Output0	TIMER43_ITI
		TRIGSEL_TIMER44ITI	Output0	TIMER44_ITI
		TRIGSEL_DAC0OUT0	Output0	DAC0_OUT0_EXTRG
		TRIGSEL_DAC0OUT1	Output0	DAC0_OUT1_EXTRG

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER43_MCH0 TIMER44_TRGO TIMER44_CH0 TIMER44_CH1 TIMER44_MCH0 ADC0_WD0_OUT_H ADC0_WD0_OUT_L ADC0_WD1_OUT_H ADC0_WD1_OUT_L ADC0_WD2_OUT_H ADC0_WD2_OUT_L ADC1_WD0_OUT_H ADC1_WD0_OUT_L ADC1_WD1_OUT_H ADC1_WD1_OUT_L ADC1_WD2_OUT_H ADC1_WD2_OUT_L ADC2_WD0_OUT_H ADC2_WD0_OUT_L ADC2_WD1_OUT_H ADC2_WD1_OUT_L ADC2_WD2_OUT_H ADC2_WD2_OUT_L CMP0_OUT CMP1_OUT CK_OUT TIMER0_BRKIN TIMER0_CH0BRKIN TIMER0_CH1BRKIN TIMER0_CH2BRKIN TIMER7_BRKIN TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN TIMER19_BRKIN TIMER19_CH0BRKIN		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

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
TIMER19_CH1BRKIN TIMER19_CH2BRKIN TIMER14_BRKIN TIMER40_BRKIN TIMER41_BRKIN TIMER42_BRKIN TIMER43_BRKIN TIMER44_BRKIN EXTI9 EXTI11 EXTI15 RTC_Alarm RTC_TPTS		TRIGSEL_TIMER7BRKIN	Output0	TIMER7_BRKIN
		TRIGSEL_TIMER7CHBRKIN	Output0 Output1 Output2	TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN
		TRIGSEL_TIMER19BRKIN	Output0	TIMER19_BRKIN
		TRIGSEL_TIMER19CHBRKIN	Output0 Output1 Output2	TIMER19_CH0BRKIN TIMER19_CH1BRKIN TIMER19_CH2BRKIN
		TRIGSEL_TIMER14BRKIN	Output0	TIMER14_BRKIN
		TRIGSEL_TIMER40BRKIN	Output0	TIMER40_BRKIN
		TRIGSEL_TIMER41BRKIN	Output0	TIMER41_BRKIN
		TRIGSEL_TIMER42BRKIN	Output0	TIMER42_BRKIN
		TRIGSEL_TIMER43BRKIN	Output0	TIMER43_BRKIN

Trigger Source	Trigger select	TRIGSEL Register	TRIGSEL output	Peripherals
		TRIGSEL_TIMER44BRKIN	Output0	TIMER44_BRKIN
		TRIGSEL_TIMER9ITI	Output0 Output1	TIMER9_ITI2 TIMER9_ITI13
		TRIGSEL_TIMER10ITI	Output0 Output1	TIMER10_ITI2 TIMER10_ITI13
		TRIGSEL_TIMER12ITI	Output0 Output1	TIMER12_ITI2 TIMER12_ITI13
		TRIGSEL_TIMER13ITI	Output0 Output1	TIMER13_ITI2 TIMER13_ITI13

Note:

1. TRIGSEL_OUTx (x=0,…,7) cannot select any TRIGSEL_INy (y=0,…,7) as trigger source.
2. All TIMERx_ITI (x=0…4, 7, 9, 10, 12, 13, 14, 19, 40…44) cannot select signals from their own modules as trigger input 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.

9.5. Register definition

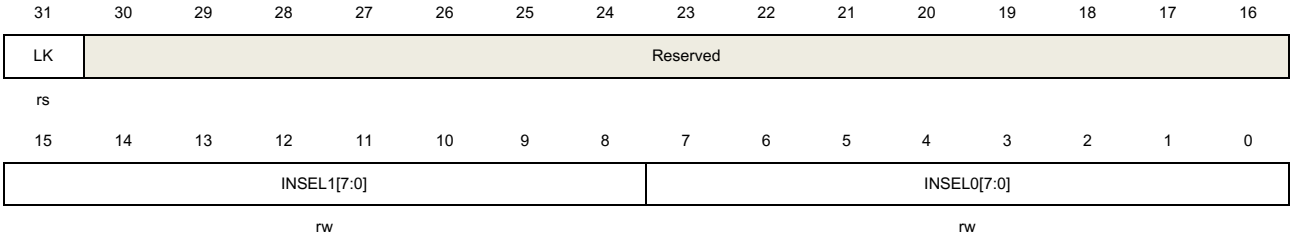
TRIGSEL base address: 0x4001 9000

9.5.1. 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)



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 9-1. 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 9-1. Trigger input bit fields selection***.

9.5.2. 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)



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 9-1. 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 9-1. Trigger input bit fields selection***.

9.5.3. 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

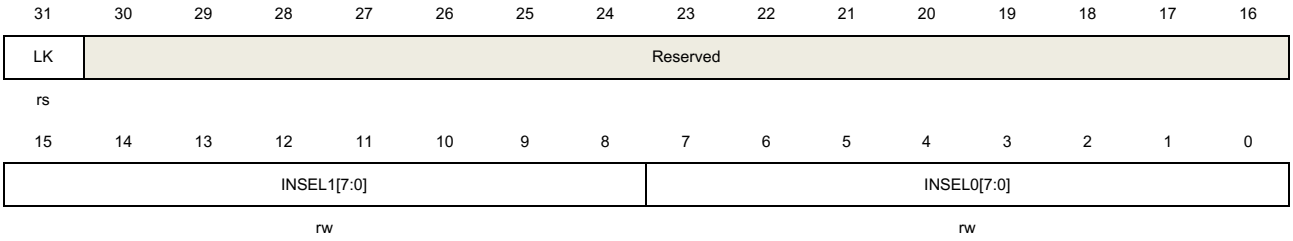
		These bits are used to select trigger input signal connected to output1. The output1 is connected with TRIGSEL_OUT5. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for output0 These bits are used to select trigger input signal connected to output0. The output0 is connected with TRIGSEL_OUT4. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.4. 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)



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 9-1. 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 9-1. Trigger input bit fields selection***.

9.5.5. Trigger selection for TIMER0_ITI register (TRIGSEL_TIMER0ITI)

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														
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_TIMER0ITI register. 0: TRIGSEL_TIMER0ITI register write is enabled. 1: TRIGSEL_TIMER0ITI 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_ITI trigger input. For the detailed configuration, please refer to *Table 9-1. Trigger input bit fields selection*.

9.5.6. Trigger selection for TIMER1_ITI register (TRIGSEL_TIMER1ITI)

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
LK	Reserved														
rs															

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

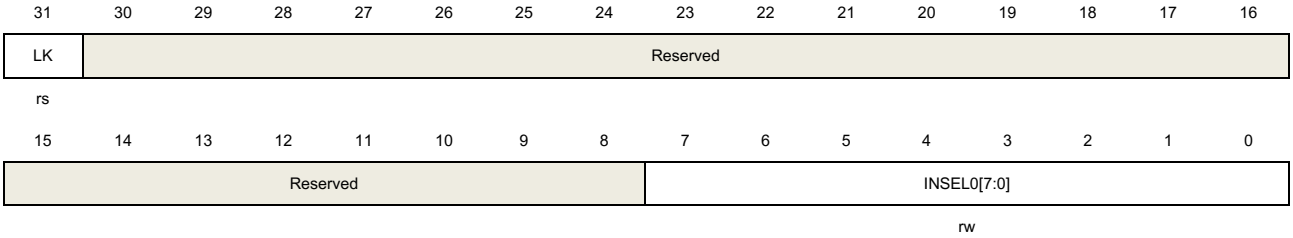
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER1ITI register. 0: TRIGSEL_TIMER1ITI register write is enabled. 1: TRIGSEL_TIMER1ITI register write is disabled.
30:8	Reserved	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 9-1. Trigger input bit fields selection*.

9.5.7. 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)

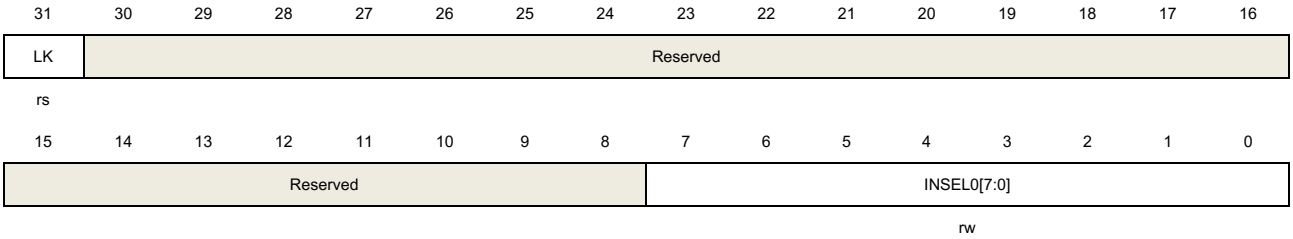


Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER2ITI register. 0: TRIGSEL_TIMER2ITI register write is enabled. 1: TRIGSEL_TIMER2ITI register write is disabled.
30:8	Reserved	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 9-1. Trigger input bit fields selection*.

9.5.8. 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)



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 9-1. Trigger input bit fields selection.***

9.5.9. 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 9-1. Trigger input bit fields selection***.

9.5.10. 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 9-1. Trigger input bit fields selection***.

9.5.11. Trigger selection for TIMER14_ITI register (TRIGSEL_TIMER14ITI)

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
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_TIMER14ITI register. 0: TRIGSEL_TIMER14ITI register write is enabled. 1: TRIGSEL_TIMER14ITI 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 TIMER14_ITI trigger input. For the detailed configuration, please refer to *Table 9-1. Trigger input bit fields selection*.

9.5.12. Trigger selection for TIMER19_ITI register (TRIGSEL_TIMER19ITI)

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_TIMER19ITI register. 0: TRIGSEL_TIMER19ITI register write is enabled. 1: TRIGSEL_TIMER19ITI 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 TIMER19_ITI trigger input. For the detailed configuration, please refer to *Table 9-1. Trigger input bit fields selection*.

9.5.13. Trigger selection for TIMER40_ITI register (TRIGSEL_TIMER40ITI)

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_TIMER40ITI register. 0: TRIGSEL_TIMER40ITI register write is enabled. 1: TRIGSEL_TIMER40ITI 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 TIMER40_ITI trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.14. Trigger selection for TIMER41_ITI register (TRIGSEL_TIMER41ITI)

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_TIMER41ITI register. 0: TRIGSEL_TIMER41ITI register write is enabled. 1: TRIGSEL_TIMER41ITI 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 TIMER41_ITI trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

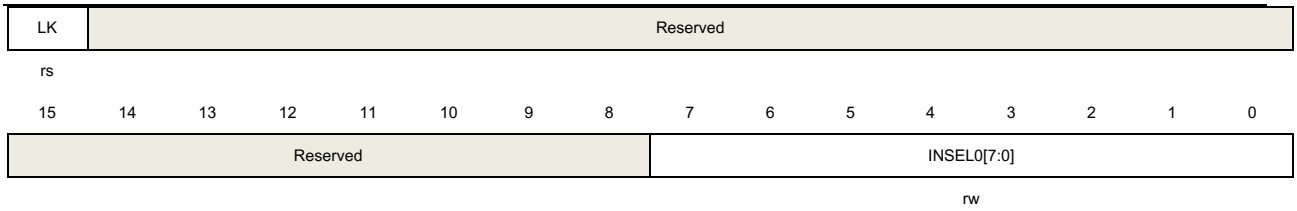
9.5.15. Trigger selection for TIMER42_ITI register (TRIGSEL_TIMER42ITI)

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



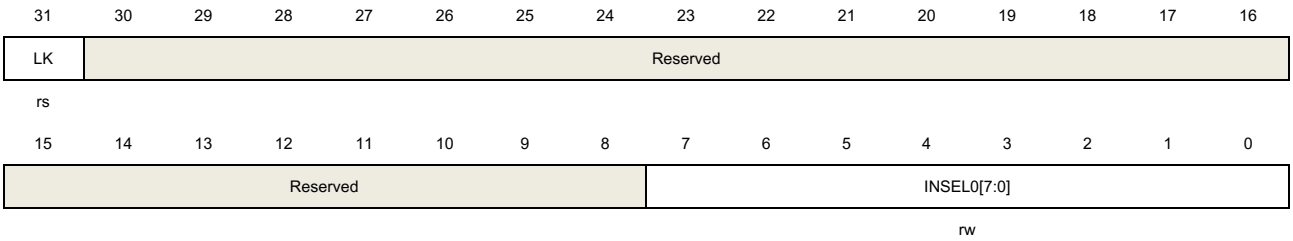
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER42ITI register. 0: TRIGSEL_TIMER42ITI register write is enabled. 1: TRIGSEL_TIMER42ITI register write is disabled.
30:8	Reserved	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 TIMER42_ITI trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.16. Trigger selection for TIMER43_ITI register (TRIGSEL_TIMER43ITI)

Address offset: 0x3C

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_TIMER43ITI register. 0: TRIGSEL_TIMER43ITI register write is enabled. 1: TRIGSEL_TIMER43ITI 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 TIMER43_ITI trigger input. For the detailed configuration, please refer to

Table 9-1. Trigger input bit fields selection.

9.5.17. Trigger selection for TIMER44_ITI register (TRIGSEL_TIMER44ITI)

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_TIMER44ITI register. 0: TRIGSEL_TIMER44ITI register write is enabled. 1: TRIGSEL_TIMER44ITI 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 TIMER44_ITI trigger input. For the detailed configuration, please refer to *Table 9-1. Trigger input bit fields selection*.

9.5.18. Trigger selection for DAC register0 (TRIGSEL_DAC0OUT0)

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_DAC0OUT0 register. 0: TRIGSEL_DAC0OUT0 register write is enabled. 1: TRIGSEL_DAC0OUT0 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 DAC0_OUT0_EXTRG (DAC0_OUT0 external trigger input). For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.19. Trigger selection for DAC register1 (TRIGSEL_DAC0OUT1)

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_DAC0OUT1 register. 0: TRIGSEL_DAC0OUT1 register write is enabled. 1: TRIGSEL_DAC0OUT1 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 DAC0_OUT1_EXTRG (DAC0_OUT1 external trigger input). For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.20. Trigger selection for ADC0_ROUTRG register (TRIGSEL_ADC0_ROUTRG)

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_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 9-1. Trigger input bit fields selection***.

9.5.21. Trigger selection for ADC0_INSTRG register (TRIGSEL_ADC0_INSTRG)

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
LK	Reserved														
rs															

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

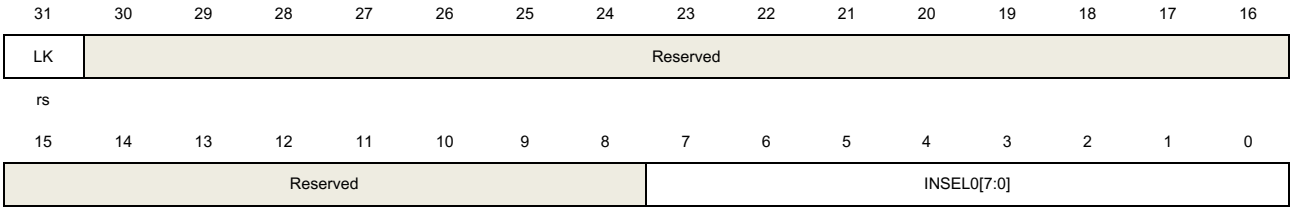
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC0_INSTRG register. 0: TRIGSEL_ADC0_INSTRG register write is enabled. 1: TRIGSEL_ADC0_INSTRG register write is disabled.
30:8	Reserved	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 9-1. Trigger input bit fields selection***.

9.5.22. Trigger selection for ADC1_ROUTRG register (TRIGSEL_ADC1_ROUTRG)

Address offset: 0x54
Reset value: 0x0000 0000

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

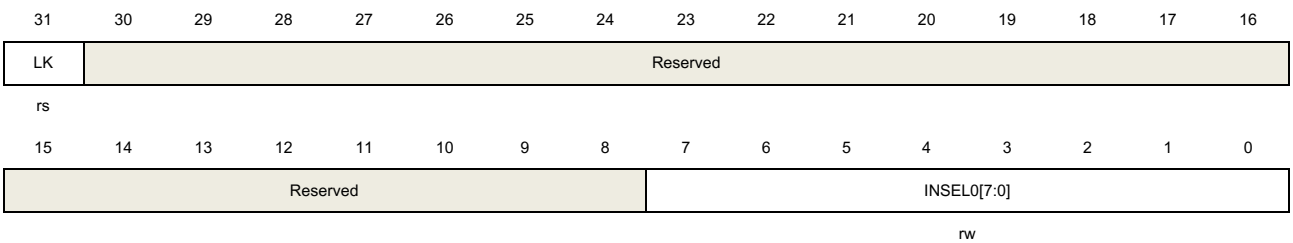


Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC1_ROUTRG register. 0: TRIGSEL_ADC1_ROUTRG register write is enabled. 1: TRIGSEL_ADC1_ROUTRG register write is disabled.
30:8	Reserved	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 9-1. Trigger input bit fields selection***.

9.5.23. Trigger selection for ADC1_INSTRG register (TRIGSEL_ADC1_INSTRG)

Address offset: 0x58
Reset value: 0x0000 0000

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



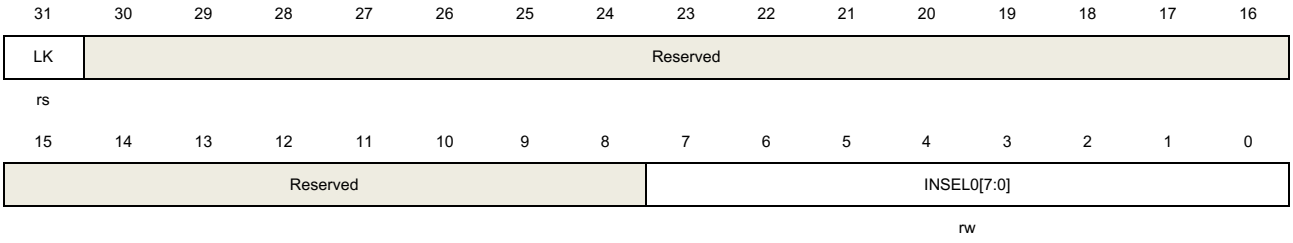
Bits	Fields	Descriptions
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_ADC1_INSTRG register. 0: TRIGSEL_ADC1_INSTRG register write is enabled. 1: TRIGSEL_ADC1_INSTRG register write is disabled.
30:8	Reserved	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 9-1. Trigger input bit fields selection***.

9.5.24. Trigger selection for ADC2_ROUTRG register (TRIGSEL_ADC2_ROUTRG)

Address offset: 0x5C

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_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 9-1. Trigger input bit fields selection.***

9.5.25. Trigger selection for ADC2_INSTRG register (TRIGSEL_ADC2_INSTRG)

Address offset: 0x60

Reset value: 0x0000 0000

This register 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 9-1. Trigger input bit fields selection***.

9.5.26. Trigger selection for TIMER0_BRKIN register (TRIGSEL_TIMER0BRKIN)

Address offset: 0x64

Reset value: 0x0000 0066

This register 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 9-1. Trigger input bit fields selection.***

9.5.27. Trigger selection for TIMER0_CHBRKIN register (TRIGSEL_TIMER0CHBRKIN)

Address offset: 0x68

Reset value: 0x0069 6867

This register 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 9-1. 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 9-1. 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 9-1. Trigger input bit fields selection.***

9.5.28. Trigger selection for TIMER7_BRKIN register (TRIGSEL_TIMER7BRKIN)

Address offset: 0x6C

Reset value: 0x0000 006A

This register 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_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 9-1. Trigger input bit fields selection***.

9.5.29. Trigger selection for TIMER7_CHBRKIN register (TRIGSEL_TIMER7CHBRKIN)

Address offset: 0x70

Reset value: 0x006D 6C6B

This register 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							

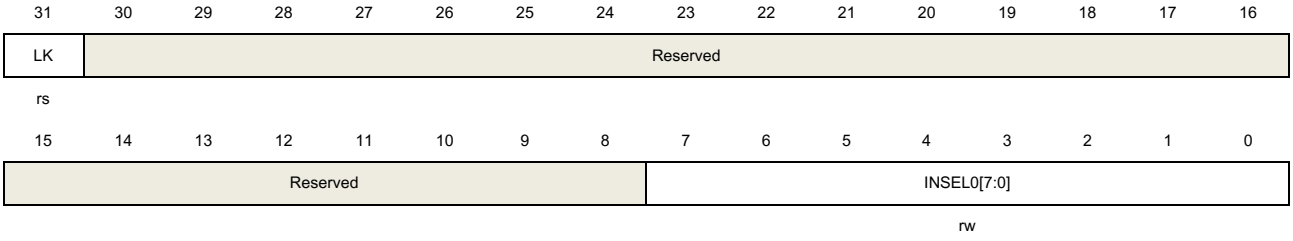
Bits	Fields	Descriptions

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 9-1. 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 9-1. 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 9-1. Trigger input bit fields selection***.

9.5.30. Trigger selection for TIMER19_BRKIN register (TRIGSEL_TIMER19BRKIN)

Address offset: 0x74
Reset value: 0x0000 006E

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_TIMER19BRKIN register. 0: TRIGSEL_TIMER19BRKIN register write is enabled. 1: TRIGSEL_TIMER19BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.

7:0	INSEL0[7:0]	Trigger input source selection for TIMER19_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER19_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.31. Trigger selection for TIMER19_CHBRKIN register (TRIGSEL_TIMER19CHBRKIN)

Address offset: 0x78

Reset value: 0x0071 706F

This register 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_TIMER19CHBRKIN register. 0: TRIGSEL_TIMER19CHBRKIN register write is enabled. 1: TRIGSEL_TIMER19CHBRKIN register write is disabled.
30:24	Reserved	Must be kept at reset value.
23:16	INSEL2[7:0]	Trigger input source selection for TIMER19_CH2BRKIN These bits are used to select trigger input signal connected to output2. The output2 is used as the source of TIMER19_CH2BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
15:8	INSEL1[7:0]	Trigger input source selection for TIMER19_CH1BRKIN These bits are used to select trigger input signal connected to output1. The output1 is used as the source of TIMER19_CH1BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER19_CH0BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER19_CH0BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.32. Trigger selection for TIMER14_BRKIN register (TRIGSEL_TIMER14BRKIN)

Address offset: 0x7C
Reset value: 0x0000 0072

This register 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_TIMER14BRKIN register. 0: TRIGSEL_TIMER14BRKIN register write is enabled. 1: TRIGSEL_TIMER14BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER14_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER14_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.33. Trigger selection for TIMER40_BRKIN register (TRIGSEL_TIMER40BRKIN)

Address offset: 0x80
Reset value: 0x0000 0073

This register 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

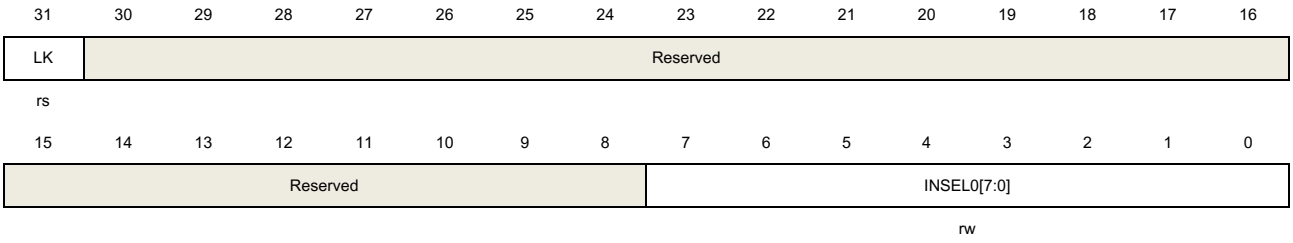
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER40BRKIN register. 0: TRIGSEL_TIMER40BRKIN register write is enabled. 1: TRIGSEL_TIMER40BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER40_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER40_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.34. Trigger selection for TIMER41_BRKIN register (TRIGSEL_TIMER41BRKIN)

Address offset: 0x84

Reset value: 0x0000 0074

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_TIMER41BRKIN register. 0: TRIGSEL_TIMER41BRKIN register write is enabled. 1: TRIGSEL_TIMER41BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER41_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER41_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.35. Trigger selection for TIMER42_BRKIN register (TRIGSEL_TIMER42BRKIN)

Address offset: 0x88
Reset value: 0x0000 0075

This register 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_TIMER42BRKIN register. 0: TRIGSEL_TIMER42BRKIN register write is enabled. 1: TRIGSEL_TIMER42BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER42_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER42_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.36. Trigger selection for TIMER43_BRKIN register (TRIGSEL_TIMER43BRKIN)

Address offset: 0x8C
Reset value: 0x0000 0076

This register 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

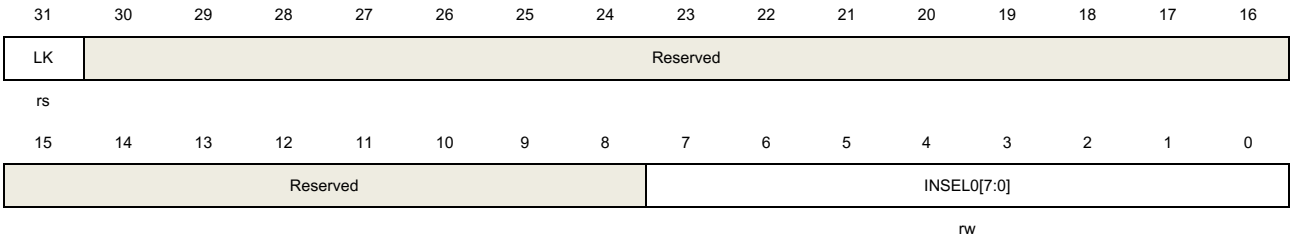
31	LK	TRIGSEL register lock. This bit is set by software and cleared only by a system reset. When it is set, it disables write access to TRIGSEL_TIMER43BRKIN register. 0: TRIGSEL_TIMER43BRKIN register write is enabled. 1: TRIGSEL_TIMER43BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER43_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER43_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.37. Trigger selection for TIMER44_BRKIN register (TRIGSEL_TIMER44BRKIN)

Address offset: 0x90

Reset value: 0x0000 0077

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_TIMER44BRKIN register. 0: TRIGSEL_TIMER44BRKIN register write is enabled. 1: TRIGSEL_TIMER44BRKIN register write is disabled.
30:8	Reserved	Must be kept at reset value.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER44_BRKIN These bits are used to select trigger input signal connected to output0. The output0 is used as the source of TIMER44_BRKIN trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.38. Trigger selection for TIMER9_ITI register (TRIGSEL_TIMER9ITI)

Address offset: 0x94

Reset value: 0x0000 0000

This register 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_TIMER9ITI register. 0: TRIGSEL_TIMER9ITI register write is enabled. 1: TRIGSEL_TIMER9ITI register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for TIMER9_ITI13 These bits are used to select trigger input signal connected to output1. The output1 is used as TIMER9_ITI13 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER9_ITI2 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER9_ITI2 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.39. Trigger selection for TIMER10_ITI register (TRIGSEL_TIMER10ITI)

Address offset: 0x98

Reset value: 0x0000 0000

This register 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_TIMER10ITI register. 0: TRIGSEL_TIMER10ITI register write is enabled.

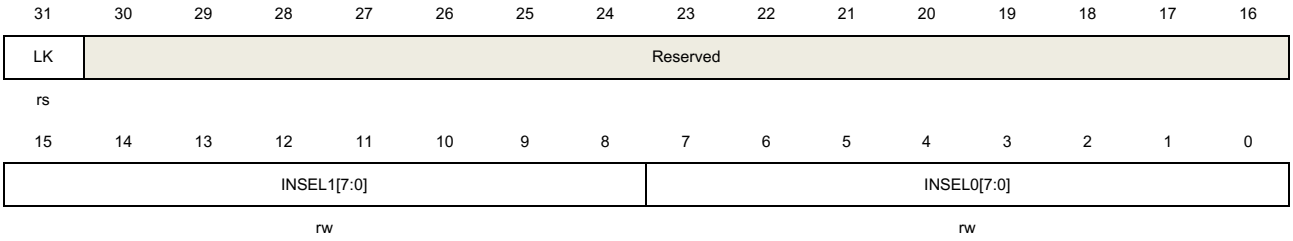
		1: TRIGSEL_TIMER10ITI register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for TIMER10_ITI13 These bits are used to select trigger input signal connected to output1. The output1 is used as TIMER10_ITI13 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER10_ITI2 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER10_ITI2 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.40. Trigger selection for TIMER12_ITI register (TRIGSEL_TIMER12ITI)

Address offset: 0x9C

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_TIMER12ITI register. 0: TRIGSEL_TIMER12ITI register write is enabled. 1: TRIGSEL_TIMER12ITI register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for TIMER12_ITI13 These bits are used to select trigger input signal connected to output1. The output1 is used as TIMER12_ITI13 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER12_ITI2 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER12_ITI2 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

9.5.41. Trigger selection for TIMER13_ITI register (TRIGSEL_TIMER13ITI)

Address offset: 0xA0

Reset value: 0x0000 0000

This register 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_TIMER13ITI register. 0: TRIGSEL_TIMER13ITI register write is enabled. 1: TRIGSEL_TIMER13ITI register write is disabled.
30:16	Reserved	Must be kept at reset value.
15:8	INSEL1[7:0]	Trigger input source selection for TIMER13_ITI13 These bits are used to select trigger input signal connected to output1. The output1 is used as TIMER13_ITI13 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.
7:0	INSEL0[7:0]	Trigger input source selection for TIMER13_ITI2 These bits are used to select trigger input signal connected to output0. The output0 is used as TIMER13_ITI2 trigger input. For the detailed configuration, please refer to ***Table 9-1. Trigger input bit fields selection***.

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 no pull-up/pull-down. 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.

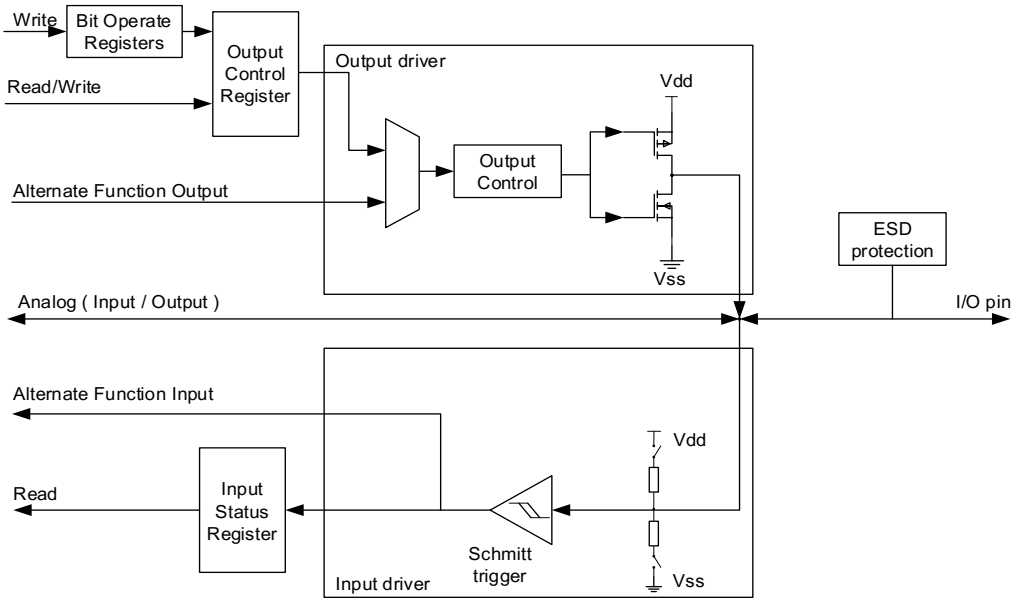
Table 10-1. GPIO configuration table

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

Note: When the port output speed is more than 50 MHz, the user should enable the I/O compensation cell. Refer to IO compensation control register (AFIO_CPSCTL).

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

Figure 10-1. Basic structure of a general-pupose I/O



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 into the input floating mode that input disabled without Pull-Up (PU) / Pull-Down (PD) resistors. But the JTAG / Serial-Wired Debug pins are in input PU/PD mode after reset:

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

There is no need to read-then-write when programming the GPIOx_OCTL at bit level, 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.

10.3.2. External interrupt/event lines

The port can use external interrupt / event lines only if it is configured in input mode.

10.3.3. Alternate functions (AF)

When the port is configured as AFIO (set the GPIOx_CTL register value to "0b10"), the port is used for external device functions. Each port has 16 alternate functions (AF), which need to be configured through the GPIOx_AFSELz (z= 0,1) register. Detailed information about port alternate function assignments can be found in the chip datasheet.

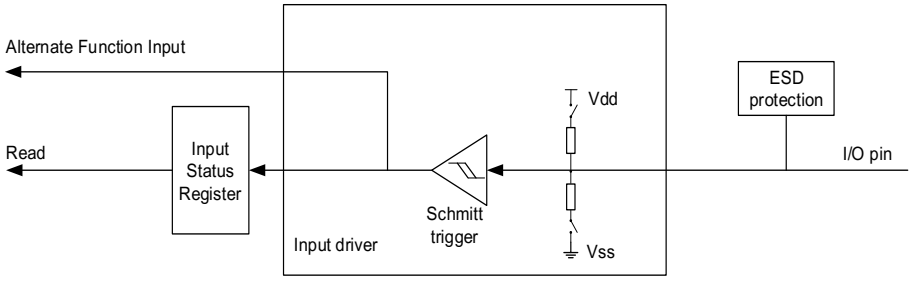
10.3.4. Input configuration

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.
- Disable the output buffer.

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

Figure 10-2. Basic structure of Input configuration



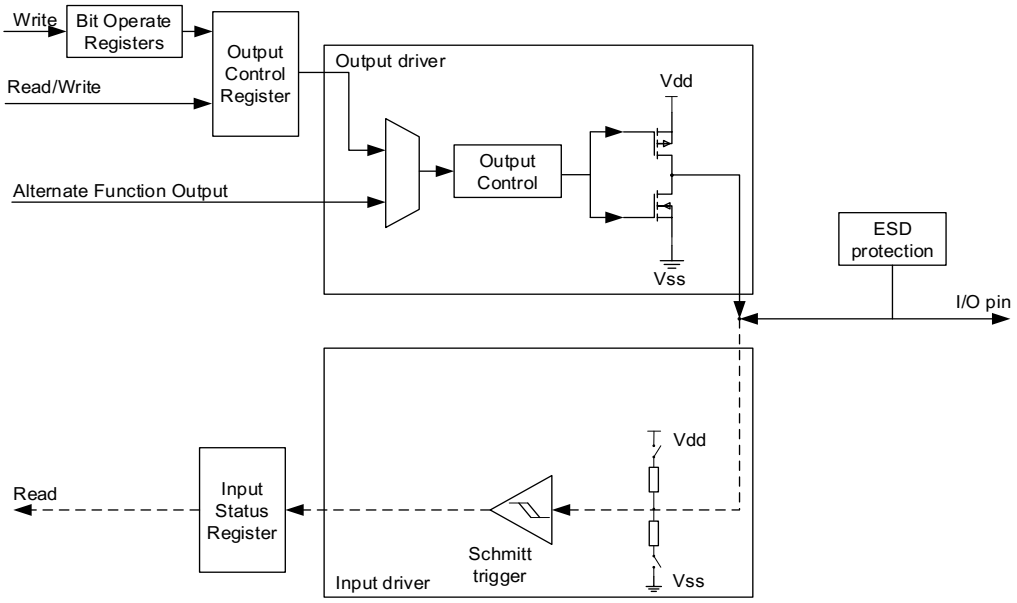
10.3.5. Output configuration

When GPIO pin is configured as output:

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

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

Figure 10-3. Basic structure of Output configuration



10.3.6. Analog configuration

When GPIO pin is used as analog configuration:

- The weak pull-up and pull-down resistors are disabled.
- The output buffer is disabled.
- The schmitt trigger input is 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



10.3.7. Alternate function (AF) configuration

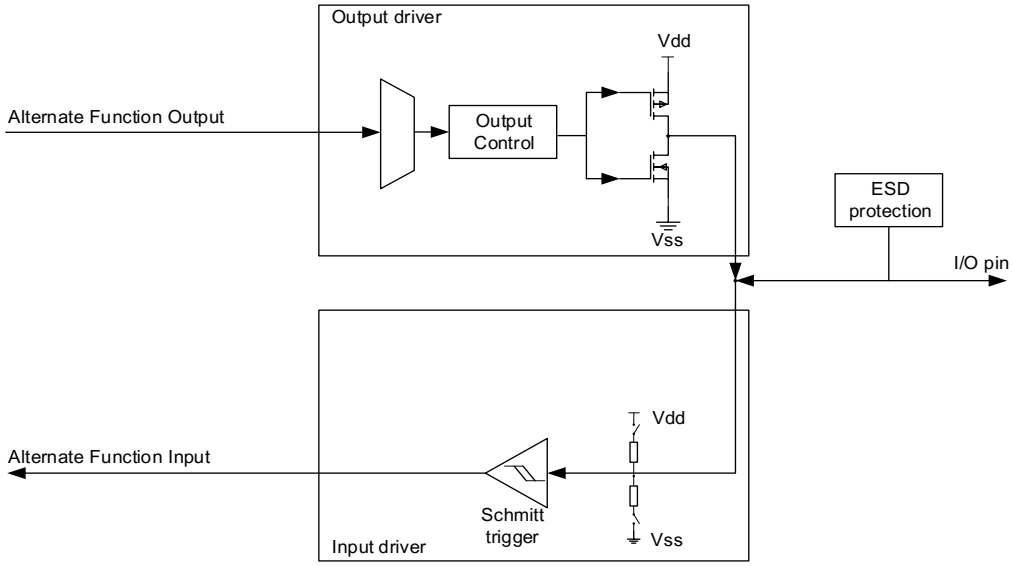
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

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.

Figure 10-5. Basic structure of Alternate function configuration



10.3.8. GPIO locking function

The locking mechanism allows the IO configuration to be protected.

The protected registers are GPIOx_CTL, GPIOx_OMODE, GPIOx_OSPD, GPIOx_PUD, GPIOx_AFSELy(y=0,1). It allows the I/O configuration to be frozen by the 32-bit locking register (GPIOx_LOCK). When the 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.

10.3.9. GPIO I/O compensation cell

By default, the I/O compensation cell is not used. However, when the I/O port output speed is more than 50MHz, it is recommended to use the compensation cell for slew rate control to reduce the I/O noise on power supply.

When the compensation cell is enabled, a complete flag CPS_RDY is set to indicate that the compensation cell is ready and can be used.

10.3.10. Input filtering

The type of input filtering for each GPIO pin can be select by configuring GPIOx_IFTP register. In the case of GPIO, filtering can be specified to synchronize only to CK_APB2 or through the sampling window. For pins configured as peripheral input, in addition to synchronization to CK_APB2 or through the sampling window, the input can also be asynchronous.

Note: The status of the GPIOx_ISTAT register is the signal before GPIO filtering, and the signal output to EXTI is the signal after GPIO filtering.

Asynchronous input

This mode is used for peripherals that do not need to input synchronization or perform synchronization by the peripheral itself. If the pin is used as GPIO, the asynchronous option is invalid, and the input filter is synchronized to CK_APB2 by default.

Note: When the peripheral performs synchronization by itself, using input synchronization may result in unexpected results. In this case, the user should ensure that GPIO is configured asynchronously.

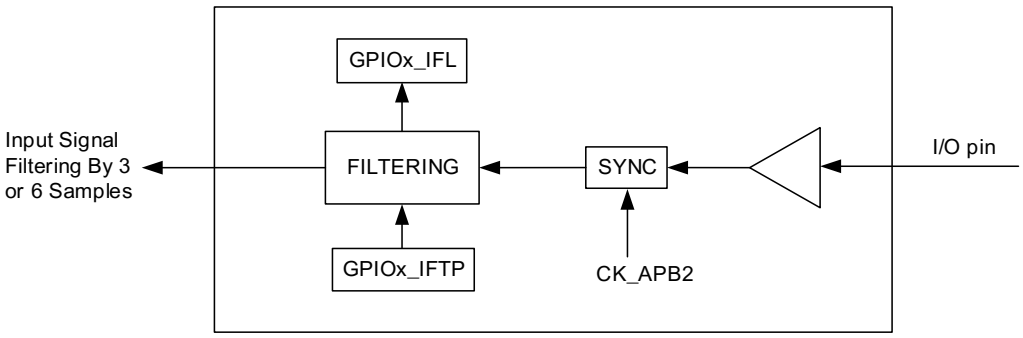
Synchronization to CK_APB2 only

This is the default filter mode for all GPIO by default. In this mode, the input signal is only associated with CK_APB2 synchronization. Since the input signal is asynchronous, it may need a delay period of CK_APB2 to change the input of MCU. The signal will not be further filtered after that.

Filtering using the sampling window

In this mode, the signal first communicates with the system clock (CK_APB2), and then the filtering process through the specified number of cycles will be carry out before allowing the input to be changed. Users need to specify two parameters for this type of filtering: sampling period and sampling times.

Figure 10-6. Filtering using the sampling window



Sampling period

To filtering the signal, the input signal is sampled in a fixed period. The sampling period is specified by the user and determines the duration between samples, or the samping frequency relative to CK_APB2.

The sampling period is determined by FLPRDx in register GPIOx_IFL. The sampling period can be configured as 8 input signal groups. For example, GPIO0 to GPIO7 use FLPRD0, GPIO8 to GPIO15 use FLPRD1.

If FLPRD0 in register GPIOx_IFL is 0, the sampling frequency is f_{CK_APB2}. For example, if f_{CK_APB2}=100MHz, the signal will be sampled at 100 MHz or every 10ns.

If FLPRD0 in register GPIOx_IFL is 0xFF(255), the sampling frequency is f_{CK_APB2}×1/(2×FLPRDx). For example, if f_{CK_APB2}=100MHz, the signal will be sampled at 100MHz×1/(2×255) or every 5.1us.

Sampling times

The sampling times of signal are 3 samples or 6 samples, detailed description in input filter type register (GPIOx_IFTP). When three or six consecutive cycles are the same, the change of input will be transmitted to MCU.

Total sampling window width

The sampling window is the time consumed to sample the input signal, as shown in ***Figure 10-7. Input filtering clock cycle***. The total width of the window can be determined by calculating the sampling period and sampling times

In order for the input filter to detect the change of the input, the signal level must be stable in the width of the sampling window or a longer time.

The number of sampling windows is always one less than the number of samples. For three sampling windows, the width of sampling window is two sampling periods. Similarly, for six sampling windows, the width of sampling window is five sampling periods.

Note: External signal variation and sampling period and CK_APB2 is asynchronous. Due to

the asynchronism of the external signal, the input should be stable for a time greater than the width of the sampling window to ensure that the change of the signal can be detected logically. The extra time required can reach the extra sampling period plus T_{CK_APB2}。

Example of sampling window

As shown in ***Figure 10-7. Input filtering clock cycle***, the input filtering configuration is as follows:

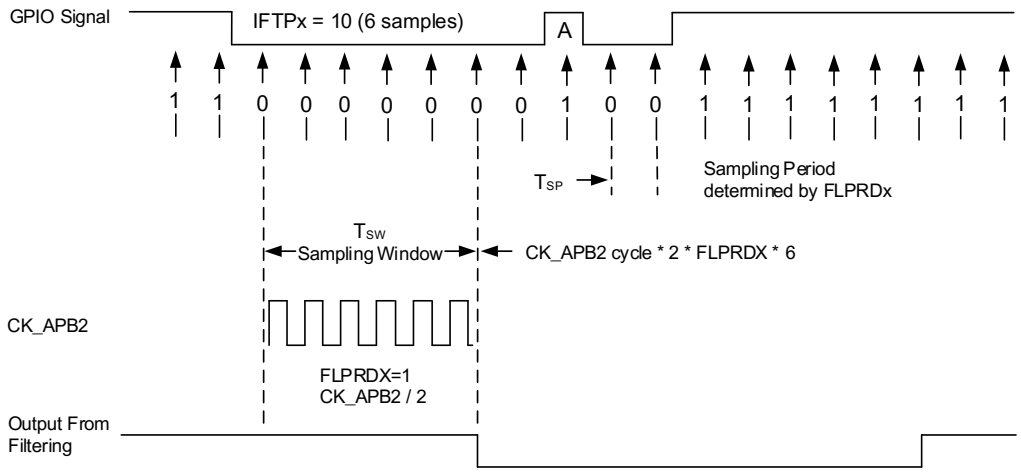
- IFTP0[1:0]=10 in GPIOx_IFTP register, which means there are 6 sampling points;
- FLPRD0=1 in GPIOx_IFL register, sampling period

$T_{SP}=2\times FLPRD0\times T_{CK_APB2}=2\times T_{CK_APB2}$;

The configuration results are as follows:

- The sampling width is: $T_{SW}=6\times T_{SP}=6\times 2\times FLPRD0\times T_{CK_APB2}=6\times 2\times T_{CK_APB2}$;
- If T_{CK_APB2}=10ns, the duration of the sampling window is:
  $T_{SP}=2\times T_{CK_APB2}=2\times 10ns=20ns$
  $T_{SW}=6\times 2\times T_{CK_APB2}=6\times 20ns=120ns$
- To illustrate the asynchronous nature of the input relative to the sampling period and the system clock, an additional smpling period and CK_APB2 cycle may be required to detect the change of input signal.
  $T_{SW}+T_{CK_APB2}=120+10=130ns$
- In ***Figure 10-7. Input filtering clock cycle***, the interference (A) is shorter than the total sampling window, so it will be filtered.

Figure 10-7. Input filtering clock cycle



10.3.11. CLK pins AF remapping

The LXTAL oscillator pins OSC32IN and OSC32OUT can be used as general-purpose I/O PC14 and PC15 individually, when the LXTAL oscillator is off. The LXTAL has priority over the GPIOs function.

Note:

1. But when the V_{DD} domain is powered off (by entering standby mode) or when the backup domain is supplied by V_{BAT} (V_{DD} no more supplied), the PC14/PC15 GPIO functionality is lost and will be set in analog mode.

2. Refer to the note on IO usage restrictions in Section *5.3.1*.

Table 10-2. OSC32 pins configuration

Alternate function	LXTAL= ON	LXTAL= OFF
PC14	OSC32IN	PC14
PC15	OSC32OUT	PC15

The HXTAL oscillator pins OSC_IN/OSC_OUT can be used as general-purpose I/O PD0/PD1.

Table 10-3. OSC pins configuration

Alternate function	HXTAL= ON	HXTAL = OFF
PD0	OSC_IN	PD0
PD1	OSC_OUT	PD1

When configured as AF8 and event output is used, if both OSCIN/OSCOUT and PD0/PD1 exist, after remapping, PD0/PD1 will output 0.

For the 100-pin package, separate PD0/PD1 pins can be used instead of OSC-PD0/PD1 pins.

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; 0xFF3F FEBF 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 up to 2MHz 01: Output speed up to 10MHz 10: Output speed up to 50MHz 11: Output speed up to 100MHz

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 y 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 y 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 ... 1111: AF15 selected

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).



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 ... 1111: AF15 selected

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. Input filtering register (GPIOx_IFL, x=A..E)

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
FLPRD1								FLPRD0							
rw								rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:8	FLPRD1	Filter sampling period for GPIO8 to GPIO15: 00: FLPRDx = CK_APB2 01: FLPRDx = CK_APB2 / 2 02: FLPRDx = CK_APB2 / 4 .... FF: FLPRDx = CK_APB2 / 510
7:0	FLPRD0	Filter sampling period for GPIO1 to GPIO7: 00: FLPRDx = CK_APB2 01: FLPRDx = CK_APB2 / 2 02: FLPRDx = CK_APB2 / 4 .... FF: FLPRDx = CK_APB2 / 510

10.4.13. Input filtering type register (GPIOx_IFTP, x=A…E)

Address offset: 0x34

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
IFTP15[1:0]		IFTP14[1:0]		IFTP13[1:0]		IFTP12[1:0]		IFTP11[1:0]		IFTP10[1:0]		IFTP9[1:0]		IFTP8[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
IFTP7[1:0]		IFTP6[1:0]		IFTP5[1:0]		IFTP4[1:0]		IFTP3[1:0]		IFTP2[1:0]		IFTP1[1:0]		IFTP0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	IFTP15[1:0]	Pin 15 input filtering type bits

		These bits are set and cleared by software. Refer to IFTP0[1:0] description
29:28	IFTP14[1:0]	Pin 14 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
27:26	IFTP13[1:0]	Pin 13 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
25:24	IFTP12[1:0]	Pin 12 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
23:22	IFTP11[1:0]	Pin 11 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
21:20	IFTP10[1:0]	Pin 10 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
19:18	IFTP9[1:0]	Pin 9 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
17:16	IFTP8[1:0]	Pin 8 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
15:14	IFTP7[1:0]	Pin 7 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
13:12	IFTP6[1:0]	Pin 6 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
11:10	IFTP5[1:0]	Pin 5 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
9:8	IFTP4[1:0]	Pin 4 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
7:6	IFTP3[1:0]	Pin 3 input filtering type bits These bits are set and cleared by software.

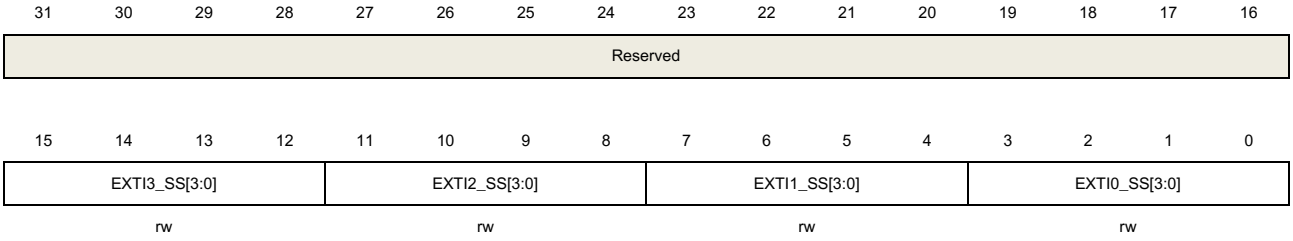
		Refer to IFTP0[1:0] description
5:4	IFTP2[1:0]	Pin 2 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
3:2	IFTP1[1:0]	Pin 1 input filtering type bits These bits are set and cleared by software. Refer to IFTP0[1:0] description
1:0	IFTP0[1:0]	Pin 0 input filtering type bits These bits are set and cleared by software. 00: Synchronization 01: Filtering (3 samples) 10: Filtering (6 samples) 11: Asynchronous (no synchronization or filtering)

10.4.14. EXTI sources selection register 0 (AFIO_EXTISS0)

Address offset: 0x08

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	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.

10.4.15. 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

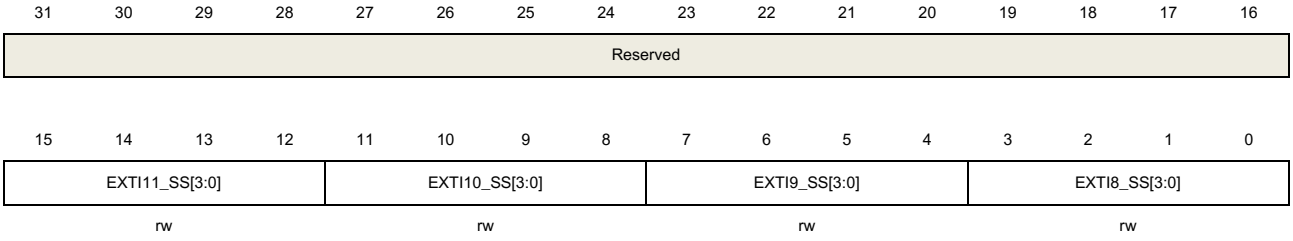
		0001: PB6 pin 0010: PC6 pin 0011: PD6 pin 0100: PE6 pin Other configurations are reserved.
7:4	EXTI5_SS[3:0]	EXTI 5 sources selection 0000: PA5 pin 0001: PB5 pin 0010: PC5 pin 0011: PD5 pin 0100: PE5 pin Other configurations are reserved.
3:0	EXTI4_SS[3:0]	EXTI 4 sources selection 0000: PA4 pin 0001: PB4 pin 0010: PC4 pin 0011: PD4 pin 0100: PE4 pin Other configurations are reserved.

10.4.16. EXTI sources selection register 2 (AFIO_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI11_SS[3:0]	EXTI 11 sources selection 0000: PA11 pin 0001: PB11 pin 0010: PC11 pin 0011: PD11 pin 0100: PE11 pin

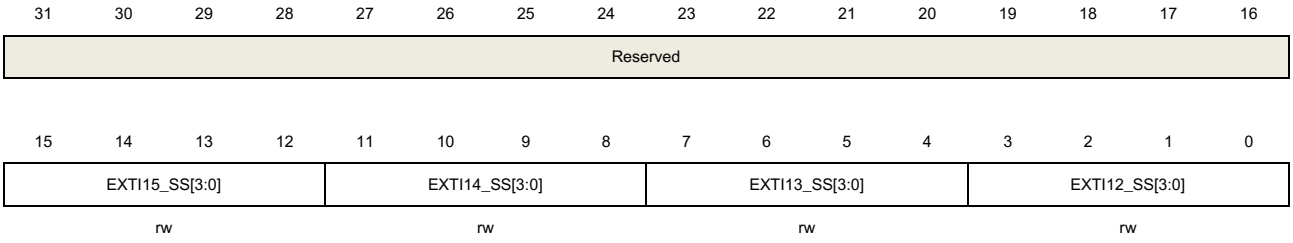
		Other configurations are reserved.
11:8	EXTI10_SS[3:0]	EXTI 10 sources selection 0000: PA10 pin 0001: PB10 pin 0010: PC10 pin 0011: PD10 pin 0100: PE10 pin Other configurations are reserved.
7:4	EXTI9_SS[3:0]	EXTI 9 sources selection 0000: PA9 pin 0001: PB9 pin 0010: PC9 pin 0011: PD9 pin 0100: PE9 pin 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.

10.4.17. EXTI sources selection register 3 (AFIO_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:12	EXTI15_SS[3:0]	EXTI 15 sources selection 0000: PA15 pin 0001: PB15 pin

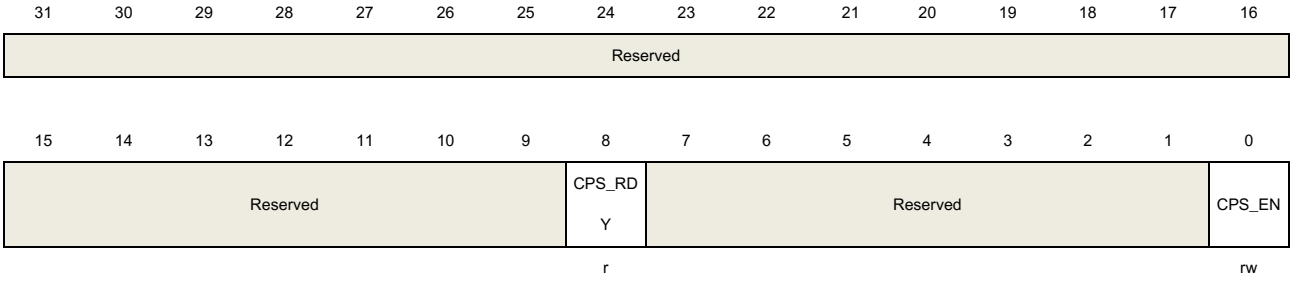
		0010: PC15 pin 0011: PD15 pin 0100: PE15 pin Other configurations are reserved.
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection 0000: PA14 pin 0001: PB14 pin 0010: PC14 pin 0011: PD14 pin 0100: PE14 pin Other configurations are reserved.
7:4	EXTI13_SS[3:0]	EXTI 13 sources selection 0000: PA13 pin 0001: 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.

10.4.18. IO compensation control register (AFIO_CPSCTL)

Address offset: 0x20

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.

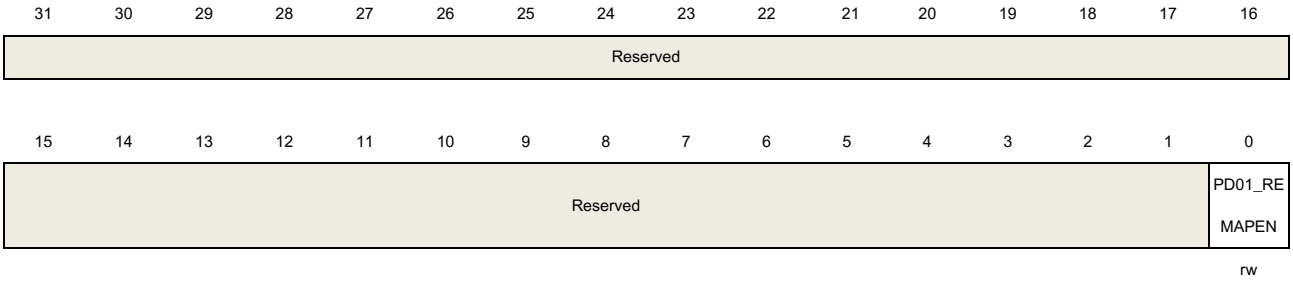
8	CPS_RDY	I/O compensation cell is ready or not. This bit is read-only. 0: I/O compensation cell is not ready 1: I/O compensation cell is ready
7:1	Reserved	Must be kept at reset value.
0	CPS_EN	I/O compensation cell enable. When the port output speed is more than 50 MHz, the user should enable the I/O compensation cell. 0: I/O compensation cell is power-down 1: I/O compensation cell is enable

10.4.19. PD01 remap register (AFIO_PD01)

Address offset: 0x24

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	PD01_REMAPEN	PD0/PD1 remapping register enable. When HXTAL is off, PD0/PD1 can still function as general-purpose I/O by configuring this register. 0: OSCIN and OSCOUT 1: PD0 and PD1

11. Cyclic redundancy checks management unit (CRC)

11.1. Overview

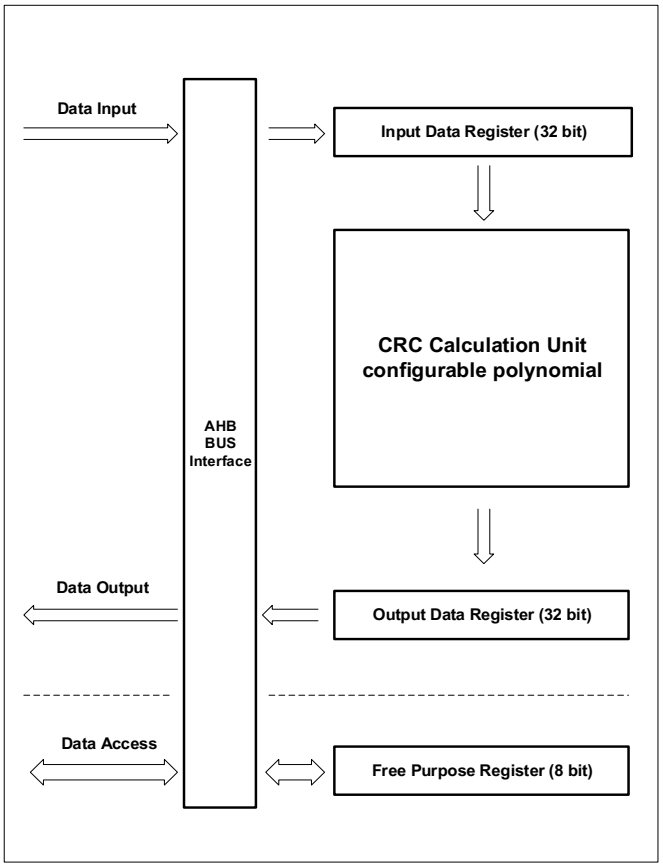
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

This CRC management unit can be used to calculate 7 / 8 / 16 / 32 bits CRC code within user configurable polynomial.

11.2. 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 11-1. Block diagram of CRC calculation unit



11.3. 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.

11.4. Register definition

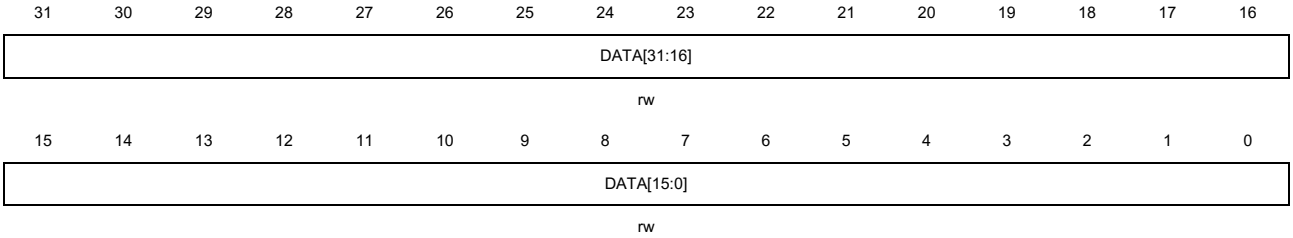
CRC base address: 0x4002 3000

11.4.1. Data register (CRC_DATA)

Address offset: 0x00

Reset value: 0xFFFF FFFF

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



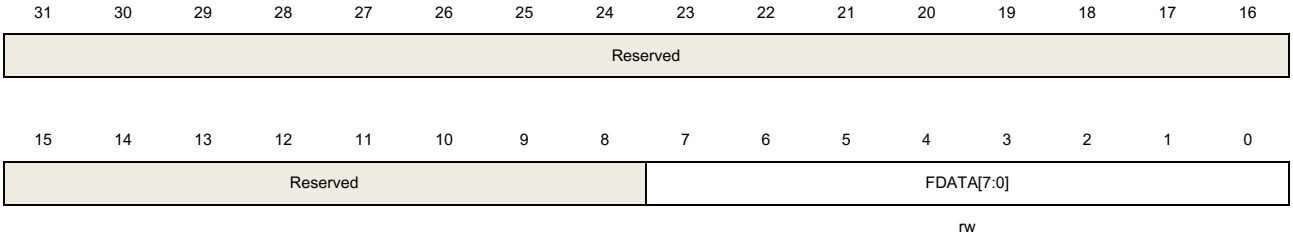
Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software writes and reads. This register is used to calculate new data, and the register can be written the new data directly. Write value cannot be read because the read value is the previous CRC calculation result.

11.4.2. 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

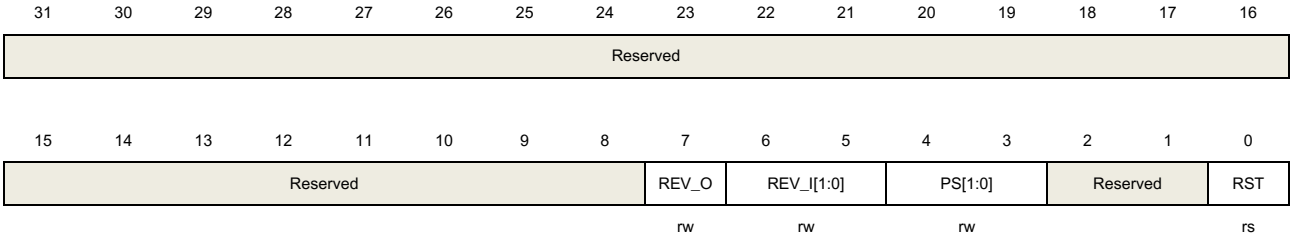
by any other peripheral. The CRC_CTL register will generate no effect to the byte.

11.4.3. 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.

11.4.4. 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.

11.4.5. 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.

12. True random number generator (TRNG)

12.1. Overview

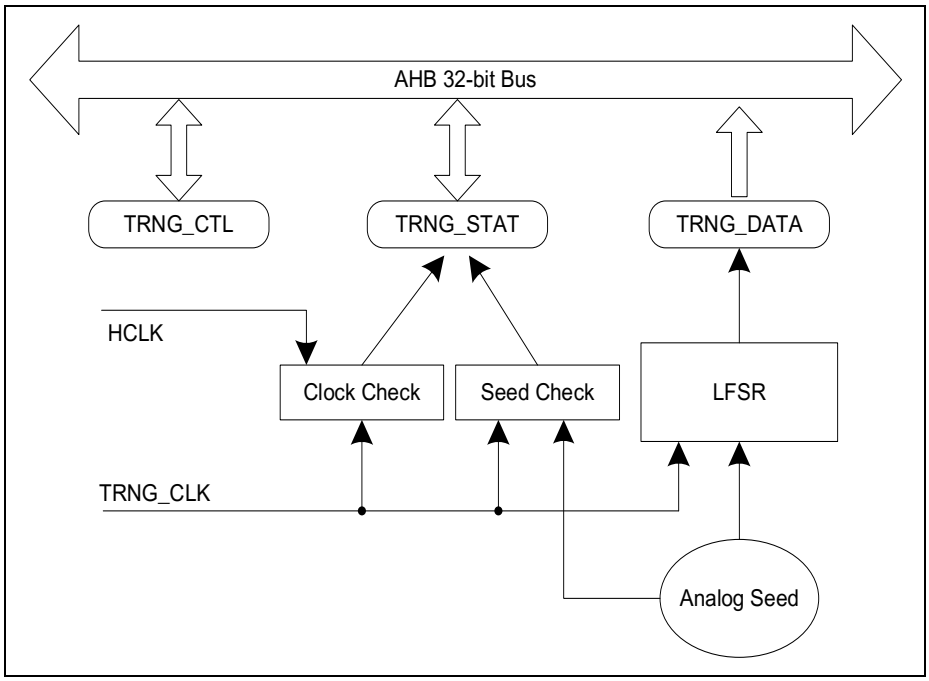
The true random number generator (TRNG) module can generate a 32-bit random value by using continuous analog noise.

12.2. Characteristics

- About 40 periods of TRNG_CLK are needed between two consecutive random numbers.
- Disable TRNG module will significantly reduce the chip power consumption
- 32-bit random value seed is generated from analog noise, so the random number is a true random number.

12.3. Function overview

Figure 12-1. 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.

12.3.1. 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 the TRNGEN bit.

3). When an interrupt occurs, check the status register TRNG_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.

12.3.2. 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.

12.4. Register definition

TRNG base address: 0x4002 3C00

12.4.1. Control register (TRNG_CTL)

Address offset: 0x00

Reset value: 0x3000 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						NR[1:0]		Reserved							
						rw									

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

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.

12.4.2. 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).

31	30	29	28	27	26	25	24	23	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									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.

12.4.3. 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

13. Cryptographic Acceleration Unit (CAU)

13.1. Overview

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with AES (128, 192, or 256) algorithms. 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.

13.2. Characteristics

- AES encryption / decryption algorithms are supported.
- Electronic codebook (ECB) is supported.
- DMA transfer for incoming and outgoing data is supported.

AES

- Supports the ECB chaining algorithms.
- 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.

13.3. CAU data type and initialization vectors

13.3.1. 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.

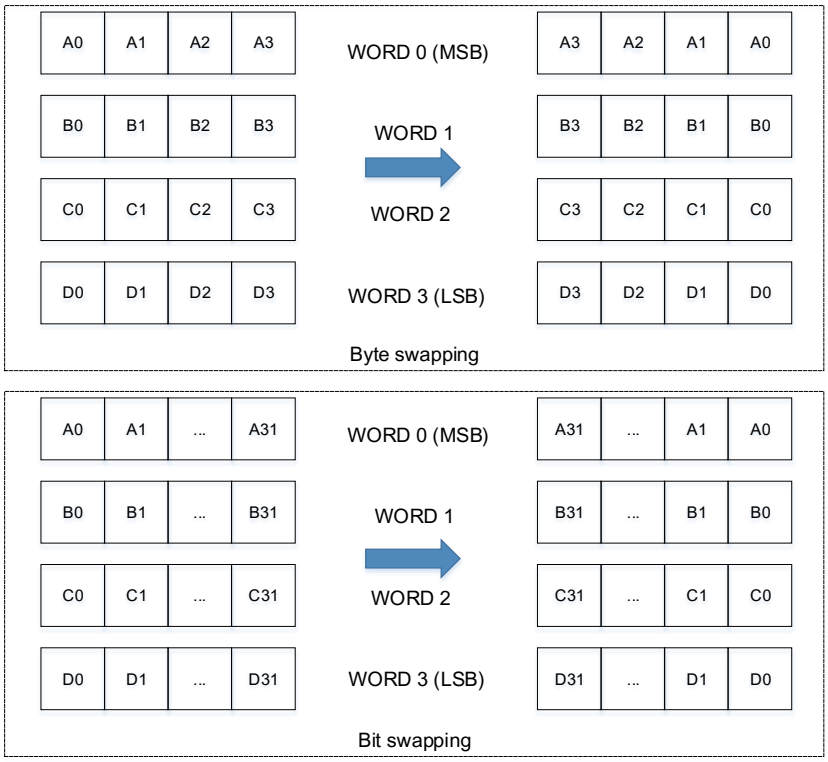
Figure 13-1. DATAM No swapping and Half-word swapping and *Figure 13-2. DATAM*

Byte swapping and Bit swapping illustrate the 128-bit AES block data swapping according to different data types.

Figure 13-1. DATAM No swapping and Half-word swapping



Figure 13-2. DATAM Byte swapping and Bit swapping

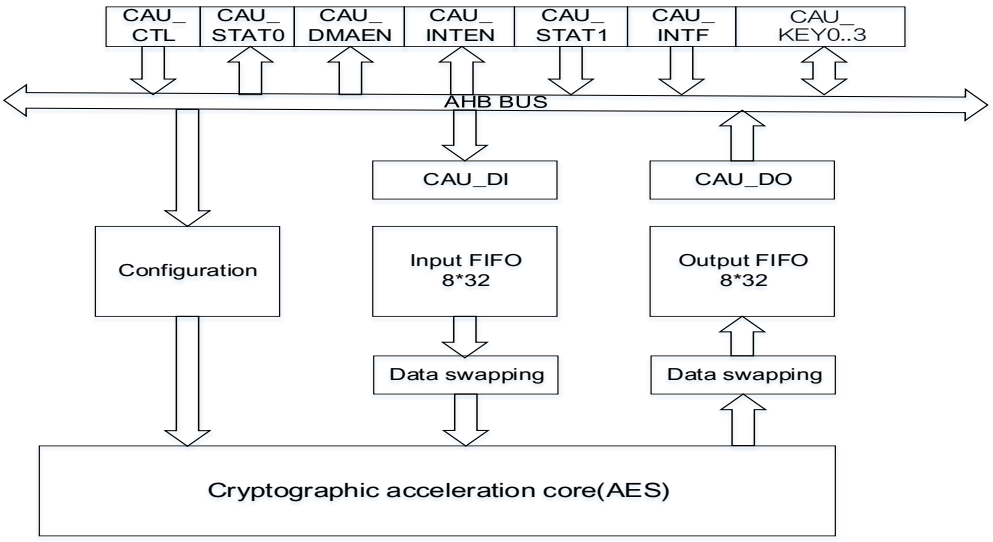


13.4. Cryptographic acceleration processor

The cryptographic acceleration unit implements AES cryptographic acceleration processors, which are detailed described in section ***AES cryptographic acceleration processor***.

Figure 13-3. CAU diagram shows the block diagram of the cryptographic acceleration unit.

Figure 13-3. CAU diagram



13.4.1. 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.

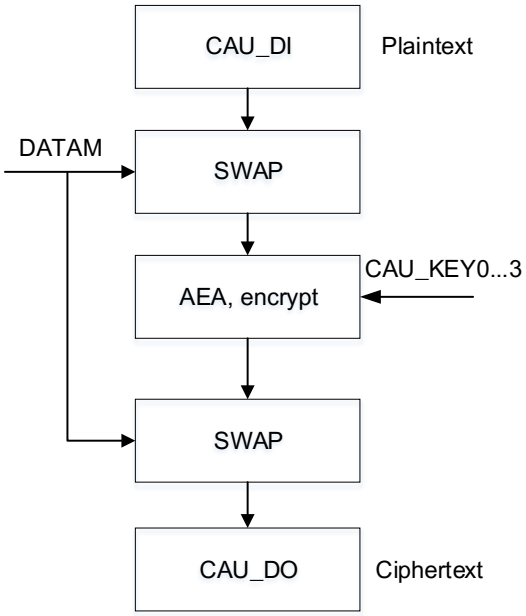
The AES key is used as [KEY3 KEY2] when the key size is configured as 128, [KEY3 KEY2 KEY1] when the key size is configured as 192 and [KEY3 KEY2 KEY1 KEY0] when the key size is configured as 256.

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 13-4. AES ECB encryption***.

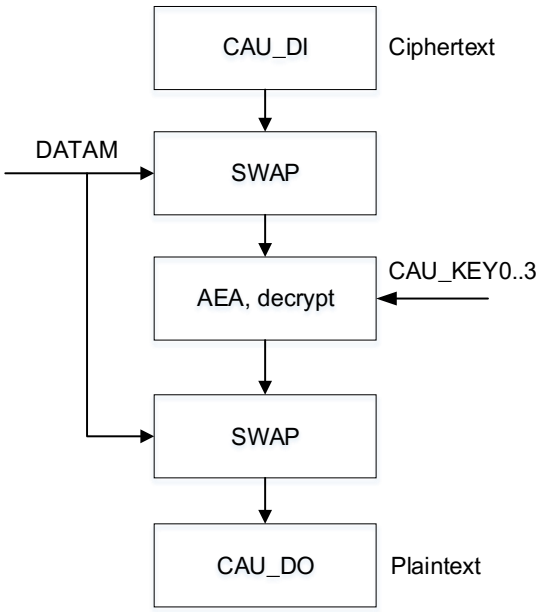
Figure 13-4. AES ECB encryption



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 13-5. AES ECB decryption***.

Figure 13-5. AES ECB decryption



13.5. 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 DATAM bit in the CAU_CTL register to select the data swapping type.
4. Configure the algorithm and the chaining mode (ECB) by writing the ALGM[2:0] bit in the CAU_CTL register.
5. Configure the encryption direction by writing 0 to the CAUDIR bit in the CAU_CTL register.
6. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
7. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
8. 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.
9. Wait for ONE bit in the CAU_STAT0 register is 1 then read the CAU_DO registers. The output data can also be transferred by DMA/CPU during interrupts/no DMA or interrupts.
10. Repeat steps 8, 9 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 DATAM bit in the CAU_CTL register to select the data swapping type.
4. Enable the CAU by set the CAUEN bit as 1.
5. Wait until the BUSY and CAUEN bit return to 0 to make sure that the decryption keys are prepared.
6. Configure the algorithm and the chaining mode (ECB) by writing the ALGM[2:0] bit in the CAU_CTL register.
7. Configure the decryption direction by writing 1 to the CAUDIR bit in the CAU_CTL register.
8. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
9. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
10. 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.
11. Wait for ONE bit in the CAU_STAT0 register is 1, then read the CAU_DO registers. The output data can also be transferred by DMA/CPU during interrupts/no DMA or interrupts.
12. Repeat steps 10, 11 until all data blocks has been decrypted.

13.6. 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.

Note: After the CAU output DMA transfer is started, the DMAOEN bit in the CAU_DMAEN register and the CAUEN bit in the CAU_CTL register must not be cleared during the transfer. If either bit is accidentally cleared, the DMA transfer will be interrupted. In this case, even if the corresponding control bit is set again, the interrupted transfer cannot resume from the point of interruption. The CAU and DMA must be reconfigured before starting the transfer again.

13.7. 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

An input FIFO interrupt is generated when the data in the input FIFO is less than four words, and the ISTA bit is set. At this time, if the IINTEN bit is set to 1 to enable the input FIFO interrupt, the IINTF bit will be set. Note if the CAUEN is low, then the ISTA and IINTF are also always low.

Output FIFO interrupt

An output FIFO interrupt is generated when one or more word data exist in the output FIFO, and the OSTA bit is set. At this time, if the OINTEN bit is set to 1 to enable the output FIFO interrupt, the OINTF bit will be set. Note Unlike that of Input FIFO interrupt, the value of CAUEN will never affect the situation of OSTA and OINTF.

13.8. Register definition

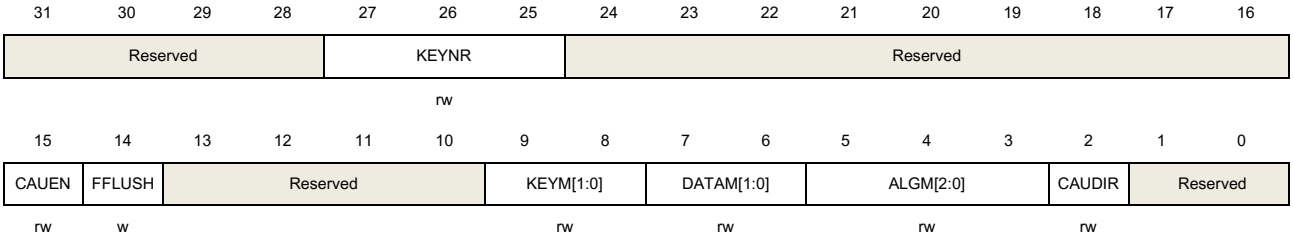
CAU base address: 0x4002 3400

13.8.1. Control register (CAU_CTL)

Address offset: 0x00

Reset value: 0x0800 0000

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:25	KEYNR	This bit field defines the number of the interal key for the selected operation. 000: select k0(from flash) as AES key 001: select k1(from flash) as AES key 010: select k2(from flash) as AES key 011: select k3(from flash) as AES key other: select k4(register) as AES key
24:16	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=1, 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 These bits and bit 19 of CAU_CTL must be configured when BUSY=0 100: 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. Note: When an unsupported mode is selected, no encryption operation will be performed.
2	CAUDIR	CAU direction, must be configured when BUSY=0 0: encryption 1: decryption
1:0	Reserved	Must be kept at reset value.

13.8.2. 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

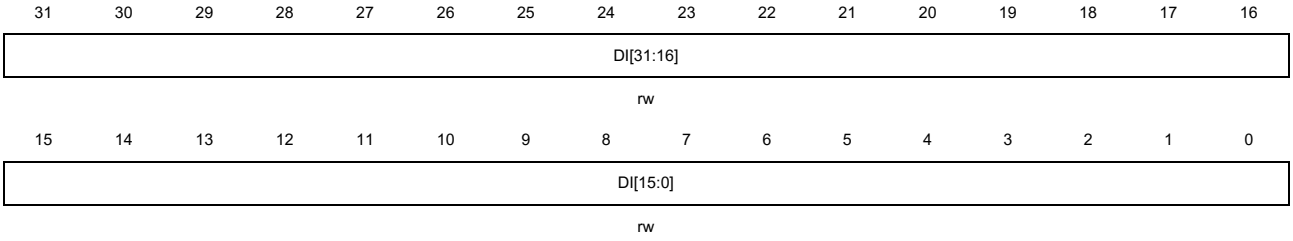
		0: Output FIFO is not full 1: Output FIFO is full
2	ONE	Output FIFO is not empty 0: Output FIFO is empty 1: Output FIFO is not empty
1	INF	Input FIFO is not full 0: Input FIFO is full 1: Input FIFO is not full
0	IEM	Input FIFO is empty 0: Input FIFO is not empty 1: Input FIFO is empty

13.8.3. 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).



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

13.8.4. 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.

13.8.5. DMA enable register (CAU_DMAEN)

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														DMAOEN	DMAIEN
														rw	rw

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

13.8.6. Interrupt enable register (CAU_INTEN)

Address offset: 0x14

Reset value: 0x0000 0000

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

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

Reserved

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

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

13.8.7. 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

13.8.8. 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

13.8.9. 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

13.8.10. Error flag register (CAU_ERROR_FLAG)

Address offset: 0x100

Reset value: 0x0000 0000

This register has 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															FMC_KEY_ERROR_FLAG
															Rc_w1

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	FMC_KEY_ERROR_FLAG	KEYNR = 000 1) Keysize ==10, 256 bits all 0 / all FF, use this key and enable CAUEN 2) Keysize ==01, k0[191:0] all 0 / all FF, use this key and enable CAUEN 3) Keysize ==00, k0[127:0] all 0 / all F, use this key and enable CAUEN KEYNR = 001 1) Keysize ==10, 256 bits all 0 / all FF, use this key and enable CAUEN 2) Keysize ==01, k1[191:0] all 0 / all FF, use this key and enable CAUEN 3) Keysize ==00, k1[127:0] all 0 / all FF, use this key and enable CAUEN KEYNR = 010 1) Keysize ==10, 256 bits all 0 / all FF, use this key and enable CAUEN 2) Keysize ==01, k2[191:0] all 0 / all FF, use this key and enable CAUEN 3) Keysize ==00, k2[127:0] all 0 / all FF, use this key and enable CAUEN KEYNR = 011 1) Keysize ==10,256 bits all 0/ all FF, use this key and enable CAUEN 2) Keysize ==01, k3[191:0] all 0/ all FF, use this key and enable CAUEN 3) Keysize ==00, k3[127:0] all 0/ all FF, use this key and enable CAUEN

Note: FMC_KEY_ERROR_FLAG is only used under the AES algorithm

14. Hash Acceleration Unit (HAU)

14.1. 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).

14.2. 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.

14.3. HAU data type

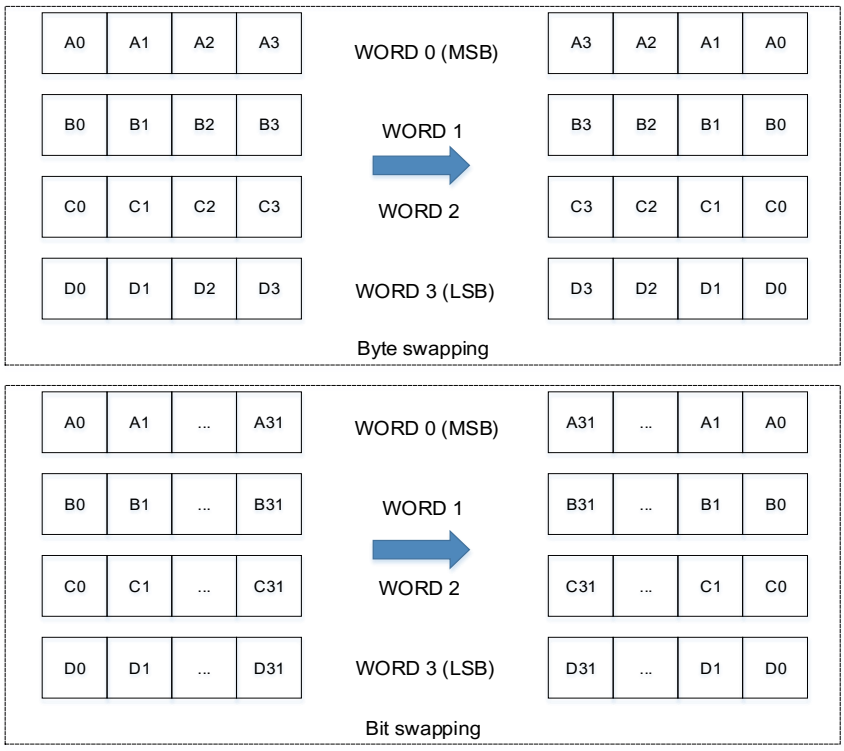
The hash acceleration unit receives data words of 32 bits at a time, while they are processed in 512-bits blocks. For each input word, according to the data type, the data could be bit / byte / half-word / no swapped before they are transferred into the hash acceleration core. The same swapping operation 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 14-1. DATAM No swapping and Half-word swapping and *Figure 14-2. DATAM Byte swapping and Bit swapping* llustrate the data swapping according to different data types.

Figure 14-1. DATAM No swapping and Half-word swapping



Figure 14-2. DATAM Byte swapping and Bit swapping

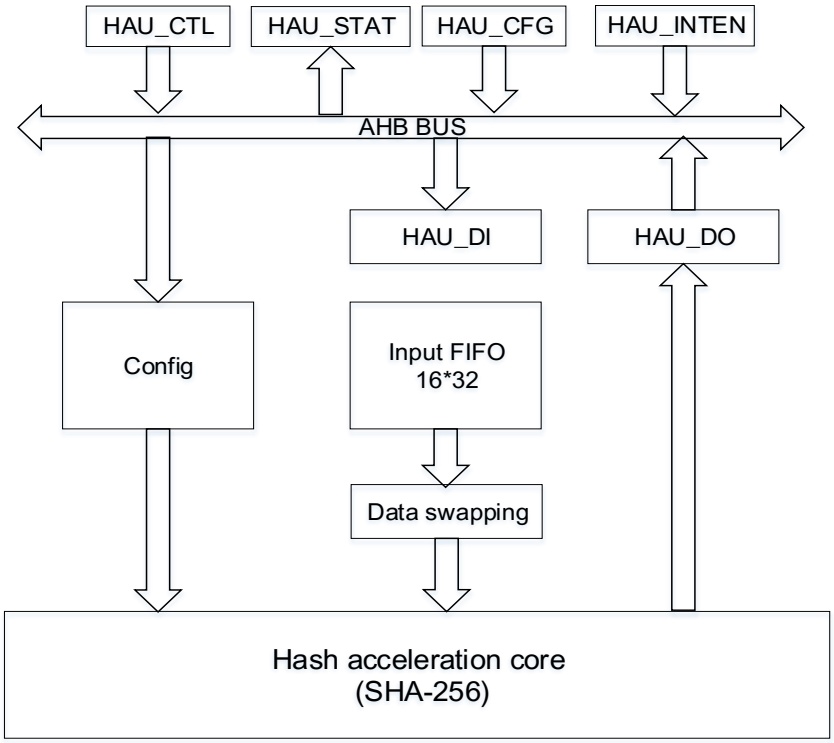


14.4. HAU core

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 14-3. HAU block diagram



14.4.1. 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

14.4.2. 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.

14.4.3. 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.

14.5. HAU interrupt

There are two interrupt sources in HAU, with corresponding status bits in the 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.

14.5.1. 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.

14.5.2. 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.

14.6. Register definition

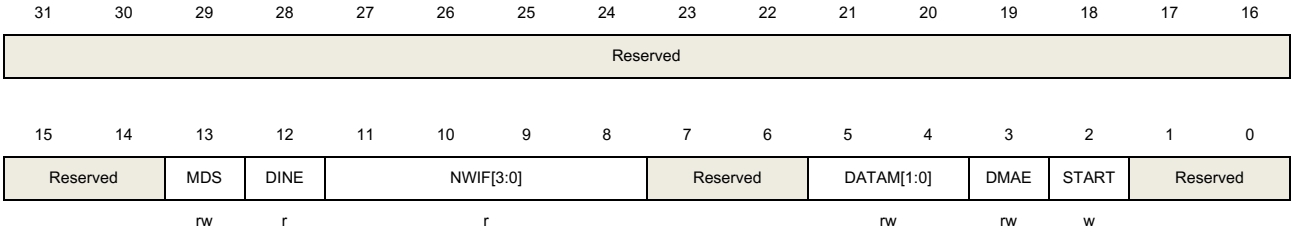
HAU base address: 0x4002 3800

14.6.1. 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: Bit swapping. The data written into HAU_DI need bit 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.

14.6.2. 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.

14.6.3. 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.

14.6.4. 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

14.6.5. 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

14.6.6. 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) and no 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)

15. Direct memory access controller (DMA)

15.1. 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. 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.

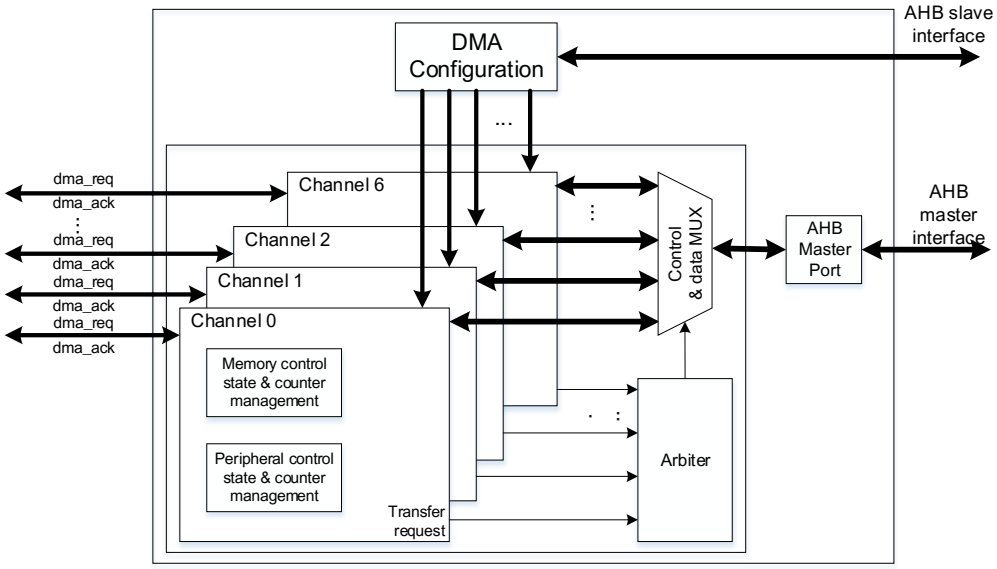
The system bus is shared by the DMA controller and the Cortex®-M33 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system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

15.2. Characteristics

- Programmable length of data to be transferred, max to 65535.
- 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.
- 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, increasing, decreasing and offset (only channel 0) address generation algorithm of memory and peripheral.
- Support circular transfer mode.
- Support channel preemption, when current channel is preempted, it is set as pending and will run the next time its priority is the highest.
- Support four transfer mode:
  - Single data element per request
  - One block transfer per request
  - Per link-list descriptor transfer per request
  - All link-list transfer per request
- Support repeat area access
- Support peripheral to memory, memory to peripheral, and memory to memory transfers.
- One separate interrupt per channel with four types of event flags.
- Support interrupt enable and clear.

15.3. Block diagram

Figure 15-1. Block diagram of DMA



As shown in ***Figure 15-1. 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.

15.4. Function overview

15.4.1. 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_CHxCTL1 registers. The DMA_CHxCNT register controls how many transfers to be transmitted on the channel. The PWIDTH and MWIDTH bits in the DMA_CHxCTL1 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 15-1. DMA transfer operation***.

Table 15-1. 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_CHxCTL0 register.

There are two possible scenarios for clearing CHEN: software clearing CHEN and hardware automatically clearing CHEN after detecting an address error.

- When the CHEN bit corresponding to DMA transmission is cleared by software, the transmission will be disabled. The stopping time for each mode of transmission is as follows:
  - TRANSMOD=00: Transmission stops after completing the current single transfer, and the FTFIFx flag is set.
  - TRANSMOD=01: Transmission stops after completing the currently started block transfer, and the FTFIFx flag is set.
  - TRANSMOD=10/11: Transmission stops after completing the currently started linklist transfer and loading the next linklist, and the FTFIFx flag is set.
  - If the CHEN bit is cleared while the current channel has not completed and no transmission is in progress, the FTFIFx flag is directly set.
  - After CHEN is disabled, software can use the FTFIFx flag to determine whether the channel has stopped transmission. Configuration modifications can only be made after transmission has stopped.
- If DMA transmission detects an address error in the data to be transmitted, it will not access the erroneous address. Hardware clears the corresponding CHEN bit to stop transmission and sets the ERRIFx flag.
- If the DMA transmission is not completed when the CHEN bit is cleared, four 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.
  - TRANSMOD=00: If any register configuration operations to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel occur, the DMA will restart a new transmission.
  - TRANSMOD=01: If any register configuration operations to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR or DMA_CHxBCNT of corresponding channel occur, the DMA will restart a new transmission.
  - TRANSMOD=10/11: If register configurations are modified before restart the DMA channel , the DMA will transfer according to the modified configurations. It should be noted that all relevant configurations need to be modified.
- If the DMA transmission has been finished when clearing the CHEN bit, enable the DMA channel without any register configuration operation will not launch any DMA transfer..

15.4.2. Link list descriptor

When TRANSMOD = 10 and ALINK = 1, the DMA enters link-list transfer mode.

In this mode, after the transfer defined by the current descriptor finishes, the DMA reloads the channel's configuration registers from the descriptor located at the link address LADDR stored in DMA_LADDR. The following registers are reloaded:

DMA_CHxCTL1, DMA_CHxCNT, DMA_CHxPADDR, DMA_CHxMADDR, DMA_CH0PAOFS, DMA_CH0MAOFS.

Until the reload completes, the channel's ACK signal is held inactive so the channel cannot be preempted by other requests.

Notes:

- While using LADDR to reload DMA transfer operation registers, software must ensure that the TRANSMOD mode is not altered.
- Memory regions used as LADDR (descriptor storage) for different channels must not overlap.
- DMA_CH0PAOFS and DMA_CH0MAOFS registers are only valid for channel0.

Table 15-2. DMA transfer operation

Register	Loading address
DMA_CHxCTL1	LADDRx[31:0] + 0x00
DMA_CHxCNT	LADDRx[31:0] + 0x04
DMA_CHxPADDR	LADDRx[31:0] + 0x08
DMA_CHxMADDR	LADDRx[31:0] + 0x0C
DMA_CHxLADDR	LADDRx[31:0] + 0x10
DMA_CH0PAOFS	LADDRx[31:0] + 0x14
DMA_CH0MAOFS	LADDRx[31:0] + 0x18

15.4.3. Data Transfer mode

The DMA supports four transfer modes as set by the field TRANSMOD in DMA_CHxCTL1.

1. When TRANSMOD is set to 00, the DMA transfers a single data element from the source address to the destination address for each DMA request. If continuous transfer of multiple data elements is needed, multiple requests must be generated. The total number of transfers (data elements) is defined in the DMA_CHxCNT register.

2. When TRANSMOD is set to 01, the DMA transfers the entire amount of data defined in the DMA_CHxCNT register after it receives one request. In this mode, block-based transfer can also be enabled through the DMA_CHxBCNT register. The DMA_CHxBCNT register encodes (Number_of_Blocks - 1); therefore the actual number of blocks transferred equals DMA_CHxBCNT + 1, and each block transfer completes one DMA handshake. During the block transfer, if the current channel is not reconfigured, the

address updates follow the configuration rules of PNAGA and MNAGA, while other configuration fields retain their initial values from the first configuration. When the block transfer is completed, an interrupt is triggered. If the BOFSMEN or BOFSPEN bit in the DMA_CHxCTL0 register is enabled, the inter-block address offset is determined by DMA_CHxPABOFS or DMA_CHxMABOFS.

3. TRANSMOD=10 is always used in conjunction with ALINK; it transfers the entire amount of data specified in DMA_CHxCNT on each request, and if ALINK is 1, the DMA will automatically flip to the next link-list descriptor and continue the transfer with one new request. In this mode, each request completes one link-list descriptor transfer, the acknowledge (ACK) is asserted when that descriptor transfer is finished, and the overall transfer sequence terminates when the ALINK in the descriptor is 0.

4. TRANSMOD=11 is also used in conjunction with ALINK; it transfers the entire amount of data specified in DMA_CHxCNT on one request, and if ALINK is 1, the DMA will automatically flip to the next link-list descriptor and continue the transfer without requiring a new request. In this mode, one request completes all link-list descriptor transfers, the acknowledge (ACK) is asserted when the entire chained transfer is finished, and the transfer sequence terminates when the ALINK in the descriptor is 0.

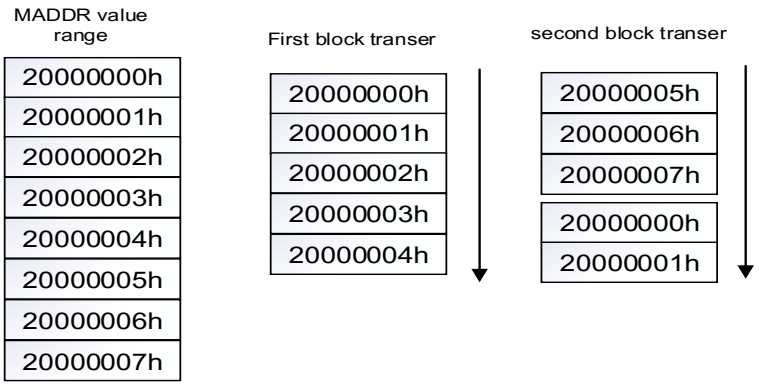
15.4.4. Repeat area access

DMA supports repeat area access as defined by the RPTAREA register. To repeatedly read from or write to a certain area while keeping the upper bits of the transfer address register fixed, users can configure the RAPA or RAPA bits of DMA_CHxCNT register. When the address reaches the boundary, it automatically returns to the starting address to continue the transfer. The size of the repeat area supports powers of 2 (in bytes).

Note: In block mode, it's not recommended to enable both repeat enable and block offset enable. If both are set to 1, only the block offset function is executed.

The following figure shows an example of repeat area access. TRANSMOD is set to 2'b01, MARA[4:0] is set to 5'b00011, MWIDTH is set to 2'b00, and BCNT is set to 16'd1.

Figure 15-2. an example of repeat address access in block transfer mode



15.4.5. Peripheral handshake

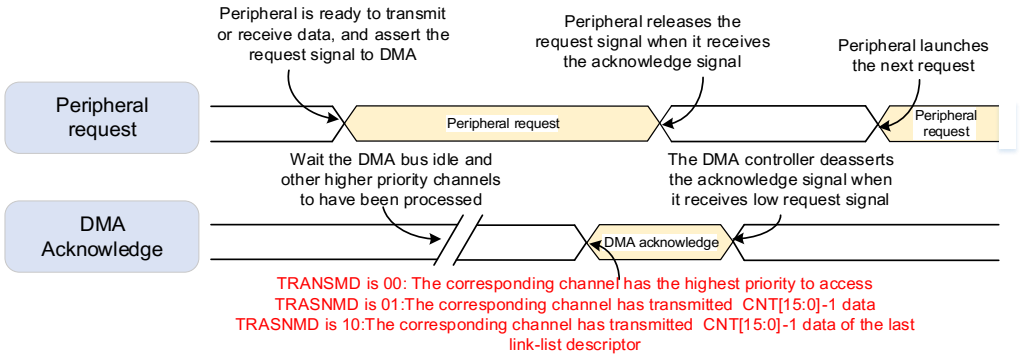
To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and an acknowledge signal:

- 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.

Note: When a peripheral issues a request to trigger the DMA, software must first enable the corresponding DMA channel (channel enable), and only then enable the related DMA function bit in the peripheral register (e.g., SPI).

Figure 15-3. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 15-3. Handshake mechanism



15.4.6. 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_CHxCTL1 register.
- For channels with equal software priority level, priority is given to the channel with lower channel number.

Preemption is programmable for each channel via the PREN bit in the DMA_CHxCTL1 register. If enabled, higher priority request will be arbitrated before current transfer completed. The channel which be preemptabled is set as pending and will run the next time its priority is the highest. If disabled, the current transfer cannot be interrupted until completed.

15.4.7. Address generation

Four kinds of address generation algorithm are implemented independently for memory and peripheral, including the fixed mode, the increasing mode, the decreasing mode and the offset mode. The PNAGA and MNAGA bit in the DMA_CHxCTL1 register are used to configure the next address generation algorithm of peripheral and memory.

In fixed mode, the next address is always equal to the base address configured in the base address registers (DMA_CHxPADDR, DMA_CHxMADDR).

In increasing mode, the next address equals the current address plus 1, 2, or 4, depending on the transfer data width.

In decreasing mode, the next address equals the current address minus 1, 2, or 4, depending on the transfer data width.

In offset mode (only Channel 0 is available), the next address equals the current address plus the offset value in DMA_CH0PAOFS or DMA_CH0MAOFS, which is represented in two's complement and supports both positive and negative offsets.

15.4.8. 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_CHxCTL0 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_CHxCTL0 register is cleared.

Note:circular mode is only valid when TRANSMOD is set to 00.

15.4.9. Memory to memory mode

The memory to memory mode is enabled by setting the M2M bit in the DMA_CHxCTL1 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_CHxCTL0 register, and completed when the DMA_CHxCNT register reaches zero.

Note: When TRANSMOD = 2'b11, the M2M bit in all descriptors must be consistent (set to the same value).

15.4.10. 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 FTFIFx bit is read as '1', configuring and starting a new DMA transfer is allowed.
2. Configure the TRANSMOD bits, M2M bit and DIR bit in the DMA_CHxCTL1 register to set the transfer mode.
3. Configure the CMEN bit in the DMA_CHxCTL0 register to enable/disable the circular mode.
4. Configure the PRIO bits in the DMA_CHxCTL1 register to set the channel software priority.
5. Configure MWIDTH, PWIDTH, MNAGA and PNAGA in the DMA_CHxCTL1 register to set the memory and peripheral transfer width and memory and peripheral address generation algorithm. If offset address mode is selected, configure DMA_CH0PAOFS and DMA_CH0MAOFS to set the address offset.
6. Configure the FTFIE, HTFIE, ERRIE and BTFIE in the DMA_CHxCTL0 register to enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer error interrupt and block transfer finish interrupt.
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 CNT bits, RASEL bits, RAPA bits and RAMA bits in the DMA_CHxCNT register to set the block transfer data number and repeat area.
10. If TRANSMOD is 01, enable block address offset by setting BOFSMEN and BOFSPEN in the DMA_CHxCTL0 register, and configure the block address offset values in the DMA_CHxPABOFS and DMA_CHxMABOFS registers.
11. If TRANSMOD is 10/11, configure the ALINK bit in the DMA_CHxCTL1 register and the DMA_CHxLADDR register to set the next descriptor address and enable automatic fetch.
12. Configure the CHEN bit with '1' in the DMA_CHxCTL0 register to enable the channel.

15.4.11. Interrupt

Each DMA channel has a dedicated interrupt. There are four types of interrupt event, including full transfer finish, half transfer finish, transfer error and block transfer finish.

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_CHxCTL0 register. The relationship is described in the following .

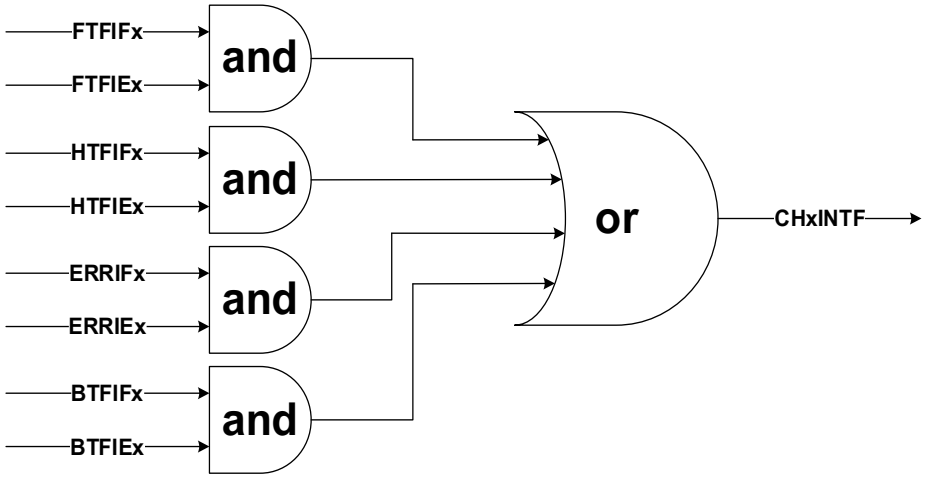
Table 15-2. interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit	Mode
	DMA_INTF	DMA_INTC	DMA_CHxCTL0	TRANSMOD
Full transfer finish	FTFIF	FTFIFC	FTFIE	00/01/10/11
Half transfer finish	HTFIF	HTFIFC	HTFIE	00/01
Transfer error	ERRIF	ERRIFC	ERRIE	00/01/10/11
Block transfer finish	BTFIF	BTFIFC	BTFIE	01/10/11

The DMA interrupt logic is shown in the *Figure 15-4. DMA interrupt logic*, an interrupt can

be produced when any type of interrupt event occurs and enabled on the channel.

Figure 15-4. DMA interrupt logic



Note: "x" indicates channel number (DMA0 corresponds to x=0...6; DMA1 corresponds to x=0...4).

15.4.12. 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 ***DMA request multiplexer (DMAMUX)***.

15.5. Register definition

DMA0 base address: 0x4002 0000

DMA1 base address: 0x4002 0400

The DMA registers have to be accessed by words (32 bits)

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

15.5.1. Interrupt flag register 0 (DMA_INTF0)

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		BTFIF5	ERRIF5	HTFIF5	FTFIF5	GIF5	BTFIF4	ERRIF4	HTFIF4	FTFIF4	GIF4	BTFIF3	ERRIF3	HTFIF3	FTFIF3
		r	r	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
GIF3	BTFIF2	ERRIF2	HTFIF2	FTFIF2	GIF2	BTFIF1	ERRIF1	HTFIF1	FTFIF1	GIF1	BTFIF0	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
4+5 *x	BTFIFx	Block transfer finish flag of channel x (x=0…5) Hardware set and software cleared by configuring DMA_INTC register. 0: Block transfer has not finished on channel x 1: Block transfer has finished on channel x
3+5*x	ERRIFx	Error flag of channel x (x=0…5) 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
2+5*x	HTFIFx	Half transfer finish flag of channel x (x=0…5) 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
1+5*x	FTFIFx	Full Transfer finish flag of channel x (x=0…5) 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
5*x	GIFx	Global interrupt flag of channel x (x=0…5)

Hardware set and software cleared by configuring DMA_INTC register.

0: None of ERRIF, HTFIF or FTFIF occurs on channel x

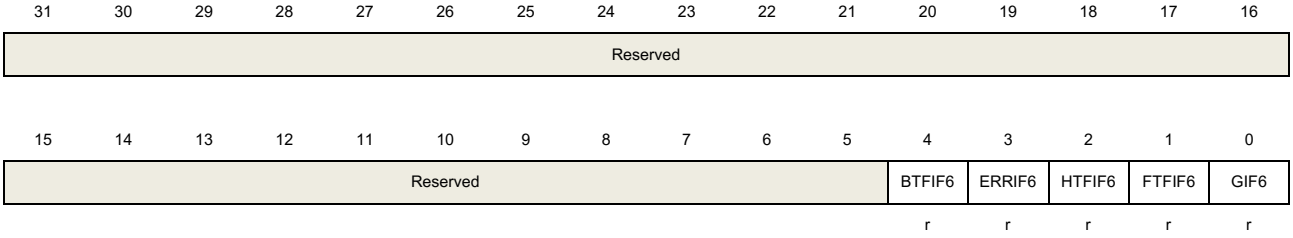
1: At least one of ERRIF, HTFIF or FTFIF occurs on channel x

15.5.2. Interrupt flag register 1 (DMA_INTF1)

Address offset: 0x04

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	BTFIFx	Block transfer finish flag of channel x (x=6) Hardware set and software cleared by configuring DMA_INTC register. 0: Block transfer has not finished on channel x 1: Block transfer has finished on channel x
3	ERRIFx	Error flag of channel x (x=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
2	HTFIFx	Half transfer finish flag of channel x (x=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
1	FTFIFx	Full Transfer finish flag of channel x (x=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
0	GIFx	Global interrupt flag of channel x (x=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

15.5.3. Interrupt flag clear register 0 (DMA_INTC0)

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		BTFIFC5	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	BTFIFC4	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4	BTFIFC3	ERRIFC3	HTFIFC3	FTFIFC3
		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
GIFC3	BTFIFC2	ERRIFC2	HTFIFC2	FTFIFC2	GIFC2	BTFIFC1	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	BTFIFC0	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
4+5 *x	BTFIFCx	Clear bit for block transfer finish flag of channel x (x=0…5) 0: No effect 1: Clear block transfer end flag
3+5*x	ERRIFCx	Clear bit for error flag of channel x (x=0…5) 0: No effect 1: Clear error flag
2+5*x	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0…5) 0: No effect 1: Clear half transfer finish flag
1+5*x	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=0…5) 0: No effect 1: Clear full transfer finish flag
5*x	GIFCx	Clear global interrupt flag of channel x (x=0…5) 0: No effect 1: Clear GIFx, ERRIFx, HTFIFx, FTFIFx and BTFIFx bits in the DMA_INTF register

15.5.4. Interrupt flag clear register 1 (DMA_INTC1)

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	BTFIFC6	ERRIFC6	HTFIFC6	FTFIFC6	GIFC6
	w	w	w	w	w

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	BTFIFCx	Clear bit for block transfer finish flag of channel x (x=6) 0: No effect 1: Clear block transfer finish flag
3	ERRIFCx	Clear bit for error flag of channel x (x=6) 0: No effect 1: Clear error flag
2	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=6) 0: No effect 1: Clear half transfer finish flag
1	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=6) 0: No effect 1: Clear full transfer finish flag
0	GIFCx	Clear global interrupt flag of channel x (x=6) 0: No effect 1: Clear GIFx, ERRIFx, HTFIFx, FTFIFx and BTFIFx bits in the DMA_INTF register

15.5.5. Channel x control register 0 (DMA_CHxCTL0)

x = 0...6, where x is a channel number

Address offset: 0x18 + 0x10 × 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						BOFSMEN	BOFSPEN	Reserved							
						rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTFIE	Reserved									CMEN	Reserved	ERRIE	HTFIE	FTFIE	CHEN
rw										rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	BOFSMEN	Enable memory block offset

		This bit can not be written when CHEN is '1' 0: Disable 1: Enable
24	BOFSPEN	Block peripheral offset enable This bit can not be written when CHEN is '1' 0: Disable 1: Enable
23:16	Reserved	Must be kept at reset value.
15	BTFIE	Enable bit for Block transfer finish interrupt Software set and cleared 0: Disable channel block transfer finish interrupt 1: Enable channel block transfer finish finish interrupt
14:6	Reserved	Must be kept at reset value.
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	Reserved	Must be kept at reset value.
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

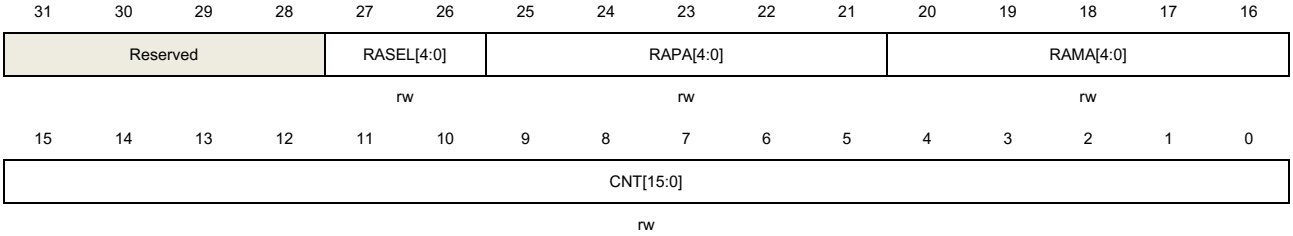
15.5.6. Channel x counter register (DMA_CHxCNT)

x = 0...6, where x is a channel number

Address offset: 0x1C + 0x10 × x

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:26	RASEL[1:0]	Repeat area selection 00: no repeat area 01: peripheral repeat area access 10: memory repeat area access 11: peripheral and memory repeat area access These bits can not be written when CHEN is '1'.
25:21	RAPA[4:0]	Repeat area of peripheral address 5'b00000: no repeat area 5'b00001: 2 bytes specified as extended repeat area by the lower 1 bit of PADDR 5'b00010: 4 bytes specified as extended repeat area by the lower 2 bit of PADDR 5'b00011: 8 bytes specified as extended repeat area by the lower 3 bit of PADDR 5'b00100: 16 bytes specified as extended repeat area by the lower 4 bit of PADDR …… 5'b11010: 64M bytes specified as extended repeat area by the lower 26 bit of PADDR 5'b11011: 128M bytes specified as extended repeat area by the lower 27 bit of PADDR These bits can not be written when CHEN is '1'.
20:16	RAMA[4:0]	Repeat area of memory address 5'b00000: no repeat area 5'b00001: 2 bytes specified as extended repeat area by the lower 1 bit of MADDR 5'b00010: 4 bytes specified as extended repeat area by the lower 2 bit of MADDR 5'b00011: 8 bytes specified as extended repeat area by the lower 3 bit of MADDR 5'b00100: 16 bytes specified as extended repeat area by the lower 4 bit of MADDR …… 5'b11010: 64M bytes specified as extended repeat area by the lower 26 bit of MADDR 5'b11011: 128M bytes specified as extended repeat area by the lower 27 bit of MADDR These bits can not be written when CHEN is '1'.

15:0	CNT[15:0]	Transfer counter These bits can not be written when CHEN 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 value is written to zero, the hardware will set it to 1, and the actual transfer count will be 1. 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.

15.5.7. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...6, where x is a channel number

Address offset: 0x20 + 0x10 × 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 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.

15.5.8. Channel x memory base address register (DMA_CHxMADDR)

x = 0...6, where x is a channel number

Address offset: 0x24 + 0x10 × 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 is '1'. When MWIDTH is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address. When MWIDTH is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

15.5.9. Channel 0 peripheral address offset register (DMA_CH0PAOFS)

Address offset: 0x118

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	PAOFS[31:0]	Paddr offset value when PNAGA is 11. 0000 0000h to 00FF FFFFh (0 bytes to (16M - 1) bytes) FF00 0000h to FFFF FFFFh (-16 Mbytes to -1 byte) These bits can not be written when CHEN is '1'.

15.5.10. Channel 0 memory address offset register (DMA_CH0MAOFS)

Address offset: 0x11C

Reset value: 0x0000 0000

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

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

Bits	Fields	Descriptions
31:0	MAOFS[31:0]	Maddr offset value when MNAGA is 11 0000 0000h to 00FF FFFFh (0 bytes to (16M - 1) bytes) FF00 0000h to FFFF FFFFh (-16 Mbytes to -1 byte)

These bits can not be written when CHEN is '1'.

15.5.11. Channel x link-list address register (DMA_CHxLADDR)

x = 0...6, where x is a channel number

Address offset: 0x120 + 0x08 × 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
LADDR[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	LADDR[31:0]	The descriptor address in link-list mode These bits can not be written when CHEN is '1'.

15.5.12. Channel x block cnounter register (DMA_CHxBCNT)

x = 0...6, where x is a channel number

Address offset: 0x124 + 0x08 × 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
BCNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	BCNT[15:0]	The block numer of DMA transferred is BCNT[15:0] + 1. These bits can not be written when CHEN is '1'.

15.5.13. Channel x peripheral block offset register (DMA_CHxPABOFS)

y = 0...6, where y is the number of channels that support block offset

Address offset: 0x1D8 + 0x08 × 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
PABOFS[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	PABOFS[31:0]	PADDR offset value between blocks when BOFSPEN is 1. 0000 0000h to 00FF FFFFh (0 bytes to (16 M - 1) bytes) FF00 0000h to FFFF FFFFh (-16 Mbytes to -1 byte) These bits can not be written when CHEN is '1'.

15.5.14. Channel x memory block offset register (DMA_CHxMABOFS)

y = 0...6, where y is the number of channels that support block offset

Address offset: 0x1DC + 0x08 × y

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	MABOFS[31:0]	MADDR offset value between blocks when BOFSMEN is 1'b1 0000 0000h to 00FF FFFFh (0 bytes to (16 M - 1) bytes) FF00 0000h to FFFF FFFFh (-16 Mbytes to -1 byte) These bits can not be written when CHEN is '1'.

15.5.15. Channel x control1 register (DMA_CHxCTL1)

x = 0...6, where x is a channel number

Address offset: 0x258 + 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									ALINK	TRANSMOD[1:0]		MNAGA[1:0]		PNAGA[1:0]	
									rw	rw		rw		rw	

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]		PREN	Reserved		DIR	Reserved			
	rw	rw		rw		rw		rw			rw				

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	ALINK	Automatically flip to the next descriptor enable This bit can not be written when CHEN is '1'. 0: Disable 1: Enable
21:20	TRANSMOD[1:0]	The data transfer mode 00: one request trig one data transfer 01: one request trig all data transfer at DMA_CHxCNT 10: one request trig one link-list descriptor transfer 11: one request trig all link-list descriptors transfer This bit can not be written when CHEN is '1'.
19:18	MNAGA[1:0]	Next address generation algorithm of memory Software set and cleared 00: Fixed address mode 01: Increasing address mode 10: Decreasing address mode 11: Offset address mode This bit can not be written when CHEN is '1'.
17:16	PNAGA[1:0]	Next address generation algorithm of peripheral Software set and cleared 00: Fixed address mode 01: Increasing address mode 10: Decreasing address mode 11: Offset address mode This bit can not be written when CHEN is '1'.
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	PREN	Preemption enable 0: Disable preemption until the data transfer is completed 1: Enable preemption when a higher-priority request arrives These bits can not be written when CHEN is '1'.
6:5	Reserved	Must be kept at reset value.
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'.
3:0	Reserved	Must be kept at reset value.

16. DMA request multiplexer (DMAMUX)

16.1. Overview

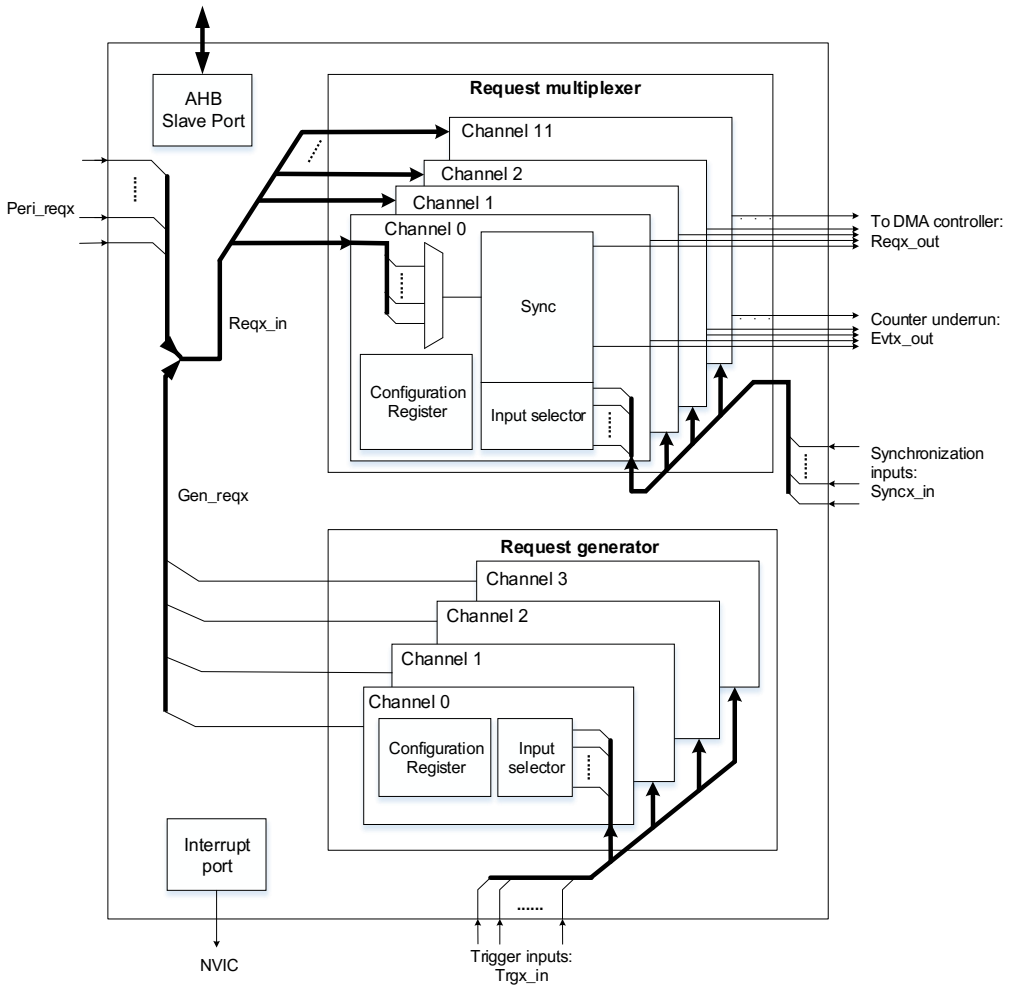
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, 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).

16.2. 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:
  - 127 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

16.3. Block diagram

Figure 16-1. Block diagram of DMAMUX



16.4. Function overview

As shown in ***Figure 16-1. 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.

16.4.1. DMAMUX signals

Table 16-1. 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

16.4.2. 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[7:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 16-3. 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[7:0] bits corresponds to no DMA request line is selected. It is not allowed to configure the same DMA request line (same non-null MUXID[7:0]) 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.

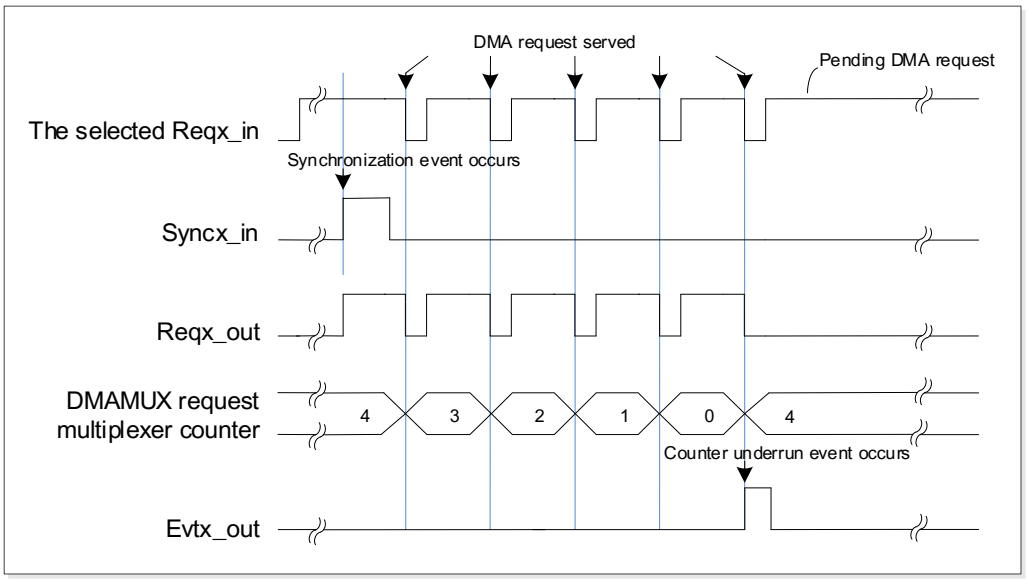
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

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 16-2. Synchronization mode shows an example when NBR[4:0]=4, SYNCEN=1, EVGEN=1, SYNCP[1:0]=01.

Figure 16-2. 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 16-5. 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

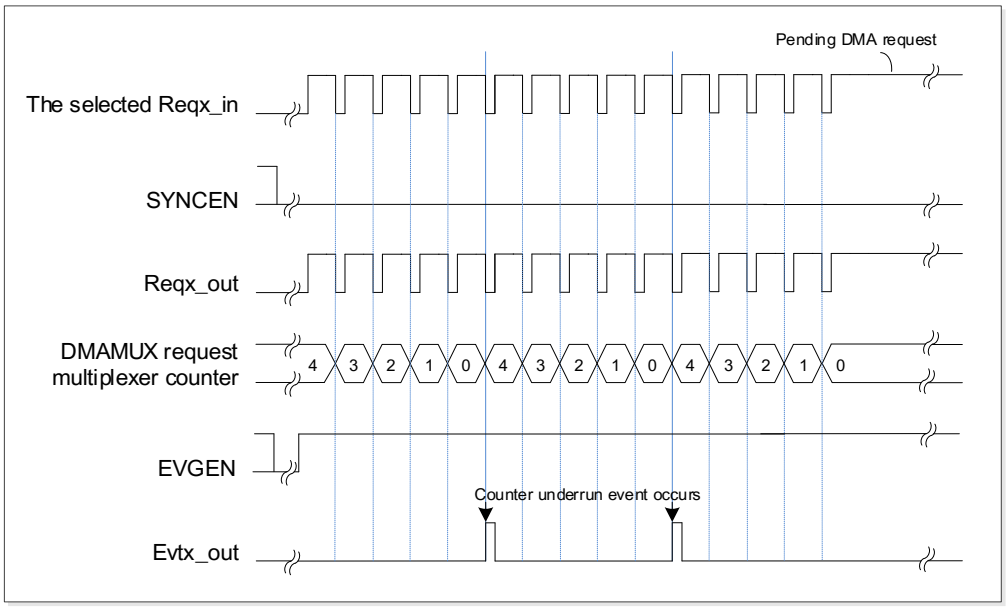
occurs again.

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 16-3. Event generation shows an example when NBR[4:0]=4, SYNCEN=0, EVGEN=1.

Figure 16-3. 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.

16.4.3. 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 16-4. 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.

16.4.4. 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.

16.4.5. 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 16-2. Interrupt events***.

Table 16-2. 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.

16.4.6. DMAMUX mapping

DMAMUX works in conjunction with DMA0 and DMA1. Channels 0 to 6 of DMAMUX are connected to channels 0 to 6 of DMA0, and channels 7 to 11 of DMAMUX are connected to channels 0 to 4 of DMA1.

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 16-3. Request multiplexer input mapping***, configured by the

MUXID[7:0] bits in the DMAMUX_RM_CHxCFG register for the DMAMUX request multiplexer channel x.

Table 16-3. Request multiplexer input mapping

Request multiplexer channel input identification MUXID[7:0]	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_TI
34	TIMER0_CMT
35	TIMER7_CH0
36	TIMER7_CH1

Request multiplexer channel input identification MUXID[7:0]	Source
37	TIMER7_CH2
38	TIMER7_CH3
39	TIMER7_UP
40	TIMER7_TI
41	TIMER7_CMT
42	TIMER1_CH0
43	TIMER1_CH1
44	TIMER1_CH2
45	TIMER1_CH3
46	TIMER1_UP
47	TIMER1_TI
48	TIMER2_CH0
49	TIMER2_CH1
50	TIMER2_CH2
51	TIMER2_CH3
52	TIMER2_UP
53	TIMER2_TI
54	TIMER3_CH0
55	TIMER3_CH1
56	TIMER3_CH2
57	TIMER3_CH3
58	TIMER3_UP
59	TIMER3_TI
60	TIMER4_CH0
61	TIMER4_CH1
62	TIMER4_CH2
63	TIMER4_CH3
64	TIMER4_UP
65	TIMER4_TI
66	ADC1_ROUTINE
67	ADC1_INSERTED
68	ADC2_ROUTINE
69	ADC2_INSERTED
70	Reserved
71	TIMER19_CH0
72	TIMER19_CH1
73	TIMER19_CH2
74	TIMER19_CH3
75	TIMER19_UP

Request multiplexer channel input identification MUXID[7:0]	Source
76	TIMER19_TI
77	TIMER19_CMT
78	TIMER14_CH0
79	TIMER14_CH1
80	TIMER14_CH0N
81	TIMER14_UP
82	TIMER14_TI
83	TIMER14_CMT
84	TIMER40_CH0
85	TIMER40_CH1
86	TIMER40_CH0N
87	TIMER40_UP
88	TIMER40_TI
89	TIMER40_CMT
90	TIMER41_CH0
91	TIMER41_CH1
92	TIMER41_CH0N
93	TIMER41_UP
94	TIMER41_TI
95	TIMER41_CMT
96	TIMER42_CH0
97	TIMER42_CH1
98	TIMER42_CH0N
99	TIMER42_UP
100	TIMER42_TI
101	TIMER42_CMT
102	TIMER43_CH0
103	TIMER43_CH1
104	TIMER43_CH0N
105	TIMER43_UP
106	TIMER43_TI
107	TIMER43_CMT
108	TIMER44_CH0
109	TIMER44_CH1
110	TIMER44_CH0N
111	TIMER44_UP
112	TIMER44_TI
113	TIMER44_CMT
114	TIMER50_UP

Request multiplexer channel input identification MUXID[7:0]	Source
115	TIMER51_UP
116	CAU_IN
117	CAU_OUT
118	HAU_IN
119	SPI3_RX
120	SPI3_TX
121	USART5_RX
122	USART5_TX
123	OSPI_TC
124	OSPI_FT
125	SDIO
126	DAC0_CH1
127	I2C2_RX
128	I2C2_TX
129	I2S1_RX
130	I2S1_TX
131	I2S2_RX
132	I2S2_TX

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 16-4. Trigger input mapping***.

Table 16-4. 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

Trigger input identification TID[4:0]	Source
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

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 16-5. Synchronization input mapping*.

Table 16-5. 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

16.5. Register definition

DMAMUX base address: 0x4002 0800

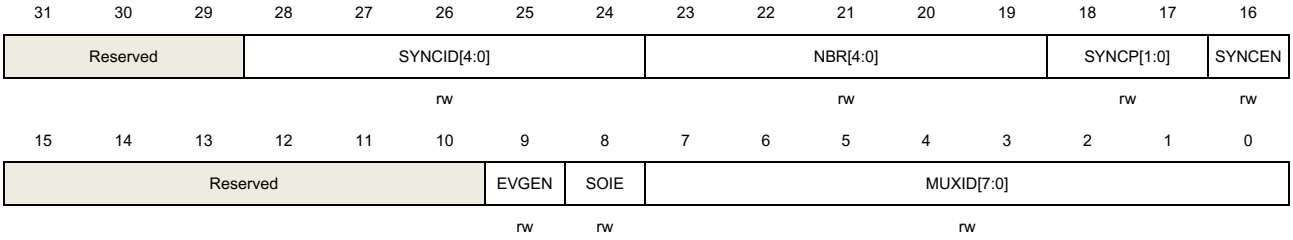
16.5.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

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
7:0	MUXID[7:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

16.5.2. 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.

16.5.3. 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.

16.5.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

x = 0...3, where x is a channel number
Address offset: 0x100 + 0x04 × x
Reset value: 0x0000 0000

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



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

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

16.5.5. 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.

16.5.6. 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.

17. Debug (DBG)

17.1. Introduction

The GD32E53x series provides a large variety of debug, trace and test features. They are implemented with a standard configuration of the Arm® CoreSight™ module together with a daisy-chained standard test access port (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 Joint Test Action Group (JTAG) debug. For details of 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 the debugger to debug power saving mode, TIMER, I2C, WWDGT, FWDGT, RTC and CAN. When corresponding bits are set, provide clock when in power saving mode or hold the state for TIMER, WWDGT, FWDGT, I2C, RTC or CAN.

17.2. 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).

17.2.1. 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.

17.2.2. 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) provides 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 asynchronous trace is enabled.

The pin assignment is shown in the following table:

Table 17-1. Pin assignment

Pin	Debug interface
PA15	JTDI
PA14	JTCK/SWCLK
PA13	JTMS/SWDIO
PB4	NJTRST
PB3	JTDO/TRACESWO

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 as other GPIO functions (NJTRST tied to 1 by hardware). If switched to SW debug mode, PA15/PB4/PB3 are released to other GPIO functions. If JTAG and SW are not used, all five pins can be released to other GPIO functions. Please refer to ***General-purpose and alternate-function I/Os (GPIO and AFIO).***

17.2.3. JTAG daisy chained structure

The Cortex®-M33 JTAG TAP is connected to an MCU JTAG TAP. The MCU JTAG instruction register (IR) is 5-bit width, while the Cortex®-M33 JTAG IR is 4-bit width. So, when JTAG in IR shift step, it first shifts 5-bit BYPASS instruction (5'b11111) for MCU JTAG, and then shifts 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.

17.2.4. Secure JTAG

1. Secure JTAG only supports JTAG but not SW

2. OTP configuration:

OTP related bits: JTAGNSW, NDBG[1:0], DBGPSWD_DATA[127:0]

Table 17-2. Debug mode OTP configuration

Mode	Register configuration
No debug	NDBG[1:0] = 2'b0X JTAGNSW: Don't care DBGPSWD_DATA[127:0]: Don't care
SW	NDBG[1:0] = 2'b10 JTAGNSW = 1'b1 DBGPSWD_DATA[127:0]: Don't care
Normal JTAG	NDBG[1:0] = 2'b11

	JTAGNSW = 1'b0 DBGPSWD_DATA[127:0]: Don't care
Secure JTAG	NDBG[1:0] = 2'b10 JTAGNSW = 1'b0 DBGPSWD_DATA[127:0]: OTP debug password value

3. Using Secure JTAG

a) Configure OTP for secure JTAG mode: DBGPSWD_DATA[127:0] as the JTAG secure password. JTAGNSW=1'b0, NDBG[1:0]= 2'b10.

b) Power reset: After the power is reset, the JTAG is in a secure state, and secure_jtag is "1". At this time, the CPU cannot be operated through JTAG.

c) Deactivate Secure JTAG: The JTAG host sequentially writes the 128-bit password to the MCU JTAG in four 32-bit words in the following sequence to release the security mode. At this time, secure_jtag is 0, and the CPU can be operated through JTAG.

Write 5'b10011 to IR, Write DBGPSWD_DATA[31:0] to data register (DR)

Write 5'b10100 to IR, Write DBGPSWD_DATA[63:32] to DR

Write 5'b10101 to IR, Write DBGPSWD_DATA[95:64] to DR

Write 5'b10110 to IR, Write DBGPSWD_DATA[127:96] to DR

Note:

1. If an incorrect password or an incorrect input sequence is entered, a power reset is required to re-enter the password.
2. Any wrong input sequence occurs, a power reset is required to re-decrypt.

3. Entering the correct password to open debug.

d) Get written value and JTAG status:

Get the written value and JTAG status via JTAG:

Write 5'b10111 to IR, Read value from DR: It can be judged whether the read IR value is the written 5'b10011

Write 5'b11000 to IR, Read value from DR: It can be judged whether the read IR value is the written 5'b10100.

Write 5'b11001 to IR, Read value from DR: It can be judged whether the read IR value is the written 5'b10101

Write 5'b11010 to IR, Read value from DR: It can be judged whether the read IR value is the written 5'b10110.

Write 5'b11011 to IR, Read value from DR: {30'b0, wrong_seq, secure_jtag}, Among them,

secure_jtag indicates the JTAG status. "1": The CPU cannot be operated via JTAG "0": The CPU can be operated via JTAG. wrong_seq indicates the decryption process error flag,"1": An error occurred in the decryption, "0": Decryption process without errors.

17.2.5. Debug reset

The JTAG-DP and SW-DP registers are in the power-on reset domain. The system reset initializes 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 features 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.

17.2.6. 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.

17.3. Debug hold function description

17.3.1. Debug support for power saving modes

When STB_HOLD bit in DBG control register (DBG_CTL0) is set and the device enters standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, so the debugger can debug in standby mode. A system reset is generated when the device exits standby mode.

When DSLP_HOLD bit in DBG control register (DBG_CTL0) is set and the device enters deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M, so the debugger can debug in deep-sleep mode.

When SLP_HOLD bit in DBG control register (DBG_CTL0) is set and the device enters sleep mode, the clock of AHB bus for CPU is not closed, so the debugger can debug in sleep mode.

17.3.2. Debug support for TIMER, I2C, WWDGT, FWDGT, CAN and RTC

When the core is halted and the corresponding bit in DBG control register (DBG_CTL0 or DBG_CTL1) is set, the peripherals behave as follows:

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

For I2C, system management bus (SMBus) timeout hold for debug.

For WWDGT or FWDGT, the counter clock stopped for debug.

For CAN, the receive register stops receiving data for debug.

For RTC, the counter clock is held for debug.

17.4. DBG registers

DBG base address: 0xE005 4000

17.4.1. 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 can only be read by software and are constant.

17.4.2. Control register 0 (DBG_CTL0)

Address offset: 0x04
Reset value: 0x0000 0000, power reset only.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TIMER10_HOLD	TIMER9_HOLD	Reserved	TIMER13_HOLD	TIMER12_HOLD	Reserved			I2C2_HOLD	CAN1_HOLD	TIMER6_HOLD	TIMER5_HOLD	TIMER4_HOLD	TIMER7_HOLD	I2C1_HOLD
	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
I2C0_HOLD	CAN0_HOLD	TIMER3_HOLD	TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGT_HOLD	FWDGT_HOLD	Reserved		TRACE_IOEN	Reserved	CAN2_HOLD	STB_HOLD	DSLP_HOLD	SLP_HOLD
rw	rw	rw	rw	rw	rw	rw	rw			rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	TIMER10_HOLD	TIMER10 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER10 counter for debug when core halted.
29	TIMER9_HOLD	TIMER9 hold bit This bit is set and reset by software. 0: no effect

		1: Hold the TIMER9 counter for debug when core halted.
28	Reserved	Must be kept at reset value.
27	TIMER13_HOLD	TIMER13 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER13 counter for debug when core halted.
26	TIMER12_HOLD	TIMER12 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER12 counter for debug when core halted.
25:23	Reserved	Must be kept at reset value.
22	I2C2_HOLD	I2C2 hold bit This bit is set and reset by software. 0: no effect 1: Hold the I2C2 SMBus timeout for debug when core halted.
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

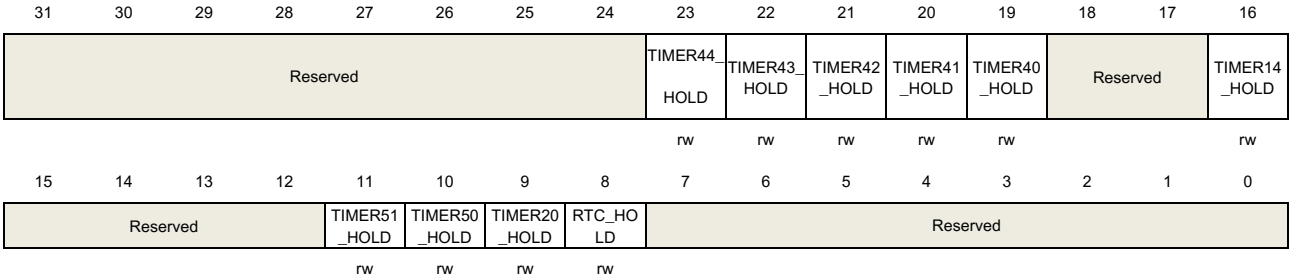
		1: Hold the I2C1 SMBus timeout for debug when core halted.
15	I2C0_HOLD	I2C0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the I2C0 SMBus timeout for debug when core halted.
14	CAN0_HOLD	CAN0 hold bit This bit is set and reset by software. 0: no effect 1: The receive register of CAN0 stops receiving data when core halted.
13	TIMER3_HOLD	TIMER3 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER3 counter for debug when core halted.
12	TIMER2_HOLD	TIMER2 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER2 counter for debug when core halted.
11	TIMER1_HOLD	TIMER1 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER1 counter for debug when core halted.
10	TIMER0_HOLD	TIMER0 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER0 counter for debug when core halted.
9	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software. 0: no effect 1: Hold the WWDGT counter clock for debug when core halted.
8	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software. 0: no effect 1: Hold the FWDGT counter clock for debug when core halted.
7:6	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	Reserved	Must be kept at reset value.
3	CAN2_HOLD	CAN2 hold bit This bit is set and reset by software. 0: no effect 1: The receive register of CAN2 stops receiving data when core halted.
2	STB_HOLD	Standby mode hold bit This bit is set and reset by software. 0: no effect 1: In the standby mode, the clock of AHB bus and system clock are provided by CK_IRC8M, a system reset is generated when the device exits standby mode.
1	DSLP_HOLD	Deep-sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC8M.
0	SLP_HOLD	Sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the sleep mode, the clock of AHB bus for CPU is not closed.

17.4.3. Control register 1 (DBG_CTL1)

Address offset: 0x08
Reset value: 0x0000 0000, power reset only.

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



Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	TIMER44_HOLD	TIMER44 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER44 counter for debug when core halted.
22	TIMER43_HOLD	TIMER43 hold bit

		This bit is set and reset by software. 0: no effect 1: Hold the TIMER43 counter for debug when core halted.
21	TIMER42_HOLD	TIMER42 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER42 counter for debug when core halted.
20	TIMER41_HOLD	TIMER41 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER41 counter for debug when core halted.
19	TIMER40_HOLD	TIMER40 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER40 counter for debug when core halted.
18:17	Reserved	Must be kept at reset value.
16	TIMER14_HOLD	TIMER14 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER14 counter for debug when core halted.
15:12	Reserved	Must be kept at reset value.
11	TIMER51_HOLD	TIMER51 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER51 counter for debug when core halted.
10	TIMER50_HOLD	TIMER50 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER50 counter for debug when core halted.
9	TIMER20_HOLD	TIMER20 hold bit This bit is set and reset by software. 0: no effect 1: Hold the TIMER20 counter for debug when core halted.
8	RTC_HOLD	RTC hold bit This bit is set and reset by software. 0: no effect 1: Hold the RTC counter clock for debug when core halted.

7:0	Reserved	Must be kept at reset value.

17.4.4. Control register 2 (DBG_CTL2)

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

Bits	Fields	Descriptions
31	LK	DBG register unlock 0: DBG register locked, registers are read-only 1: DBG register unlocked, registers are configurable
30:0	Reserved	Must be kept at reset value.

18. Analog-to-digital converter (ADC)

18.1. Overview

A 12-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip. ADC0 has 16 external channels, 5 internal channel (temperature sensor V_{SENSE}, internal reference voltage V_{REFINT}, external battery power supply V_{BAT}, positive reference voltage V_{REFP}, negative reference voltage V_{REFN}), ADC1 has 18 external channels, 3 internal channel (internal reference voltage V_{REFINT}, positive reference voltage V_{REFP}, negative reference voltage V_{REFN}), ADC2 has 18 external channels, 3 internal channel (internal reference voltage V_{REFINT}, positive reference voltage V_{REFP}, negative reference voltage V_{REFN}).

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.

18.2. 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, 18 external analog inputs for ADC2.
  - Internal channel (ADC0_CH16) for internal temperature sensor output voltage (V_{SENSE}).
  - Internal channel (ADC0_CH17, ADC1_CH18, ADC2_CH17) for internal reference voltage (V_{REFINT}).
  - Internal channel (ADC0_CH18) for external battery power supply pin (V_{BAT}).
  - Internal channel (ADC0_CH19, ADC1_CH19, ADC2_CH19) for positive reference voltage (V_{REFP}).
  - Internal channel (ADC0_CH20, ADC1_CH20, ADC2_CH20) for negative reference voltage (V_{REFN}).
- 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.
  - Overflow event.
  - Trigger error event.
- Oversampler.
  - 16-bit data register in ADC0/1/2.
  - Adjustable oversampling rate: from 2x to 256x in ADC0/1/2
  - Programmable data shift: up to 8-bit in ADC0/1/2.
- Channel input voltage range: V_{REFN} <=V_{IN} <=V_{REFP}

18.3. Pins and internal signals

Figure 18-1. ADC module block diagram shows the ADC block diagram. *Table 18-1. ADC internal input* gives the ADC internal signals and *Table 18-2. ADC input pins definition* gives the ADC pin description.

Table 18-1. ADC internal input channels

Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{REFINT}	Internal voltage reference
V_{BAT}	external battery power supply voltage
V_{REFP}	The positive reference voltage for the ADC.
V_{REFN}	The negative reference voltage for the ADC.

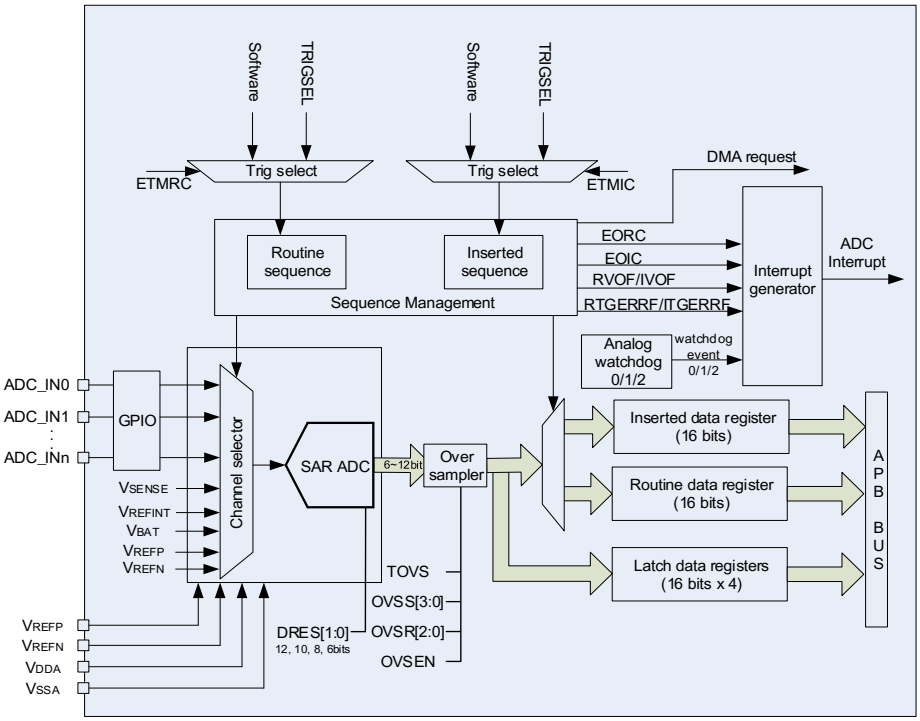
Table 18-2. ADC input pins definition

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.
ADC0_INn(n=0…15)	ADC0: Up to 16 external channels
ADC1_INn(n=0…17)	ADC1: Up to 18 external channels
ADC2_INn(n=0…16,18)	ADC2: Up to 18 external channels

Note: V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS} respectively.

18.4. Function overview

Figure 18-1. ADC module block diagram



18.4.1. 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 RCU chapter for more information on ADC clock generation.

18.4.2. 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).

18.4.3. 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.

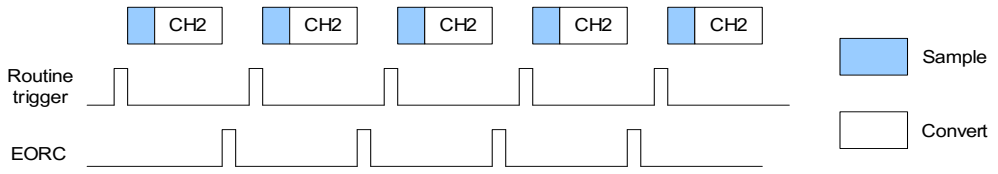
The inserted sequence supports up to 4 channels, and each channel is called inserted channel. The IL[1:0] bits in the ADC_ISQ register specify the total conversion sequence length. The ADC_ISQ register specifies the selected channels of the inserted sequence.

18.4.4. 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 18-2. 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.

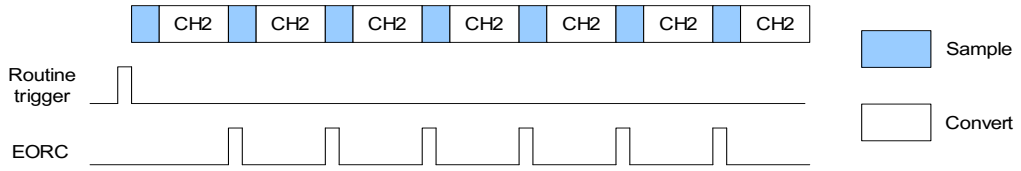
Software procedure for single operation mode of inserted sequence:

1. Make sure the DISIC, SM in the ADC_CTL0 register are reset.
2. Configure ISQ3[4:0] in the ADC_ISQ register 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 18-3. 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:

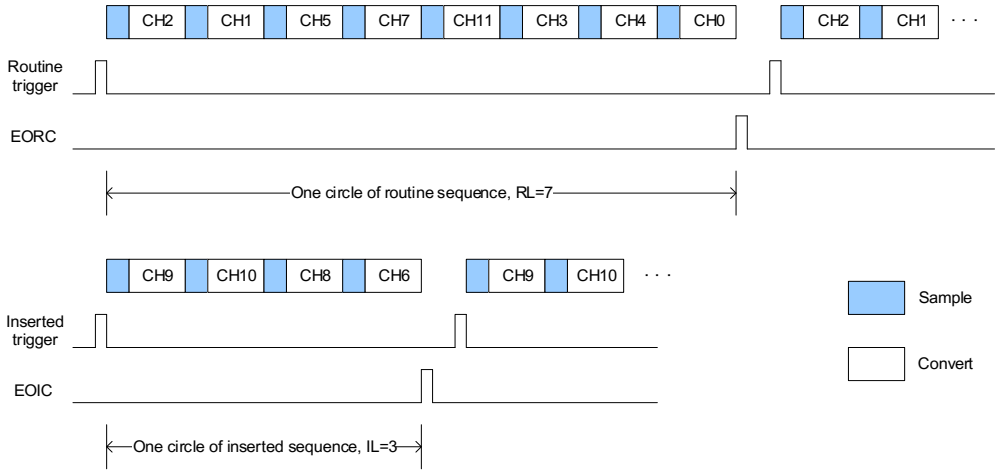
1. Set the CTN and RDMA bit in the ADC_CTL1 register.
2. Configure RSQ0[4:0] in ADC_RSQ2 register with the analog channel number.
3. Configure ADC_SAMPTx register.
4. Configure ETMRC[1:0] bits in the ADC_CTL1 register if in need.
5. Prepare the DMA module to transfer data from the ADC_RDATA.
6. Set the 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 TRIGSEL 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 18-4. Scan operation mode, continuous disable



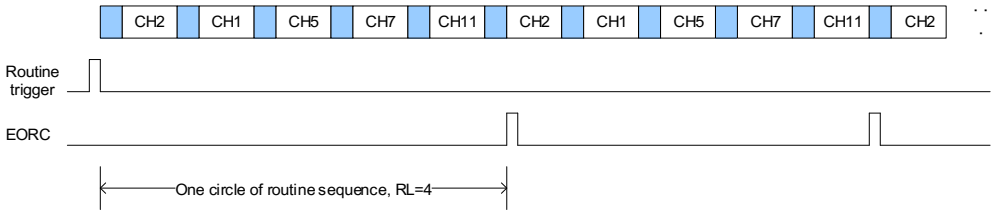
Software procedure for scan operation mode on a routine sequence:

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 18-5. Scan operation mode, continuous enable



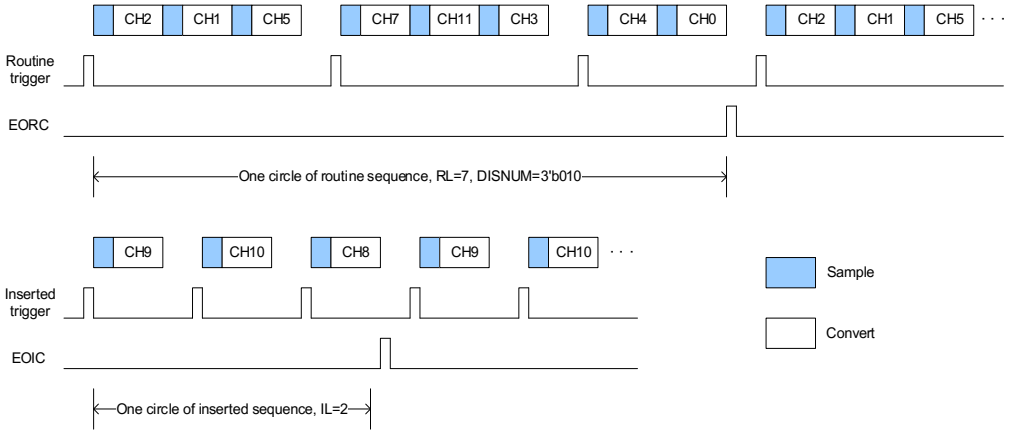
Discontinuous operation mode

For routine sequence, the discontinuous operation mode will be enabled when DISRC bit in the ADC_CTL0 register is set. In this mode, the ADC performs a short sequence of n conversions (n does not exceed 8) which is a part of the conversions selected in the ADC_RSQ0~ADC_RSQ2 registers. The value of n is configured by the DISNUM[2:0] bits in the ADC_CTL0 register. When the corresponding software trigger or TRIGSEL trigger is active, the ADC samples and converts the next n channels configured in the ADC_RSQ0~ADC_RSQ2 registers until all the channels of routine sequence are done. The 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 TRIGSEL 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 18-6. 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:

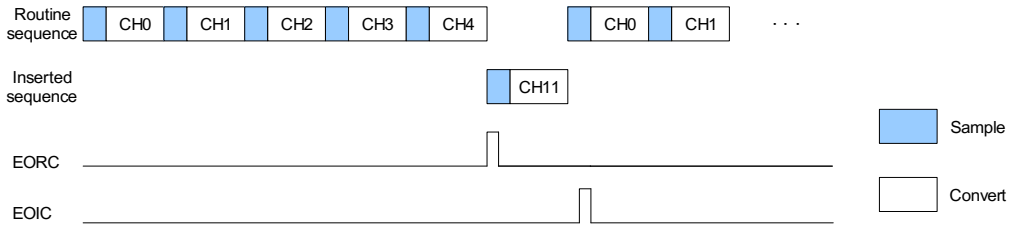
1. Set the DISIC bit in the ADC_CTL0 register and the IDMA bit in the ADC_CTL1 register.
2. Configure ADC_ISQ and ADC_SAMPTx registers.
3. Configure ETMIC[1:0] bits in the ADC_CTL1 register if in need.
4. Prepare the DMA module to transfer data from the ADC_IDATA;
5. Set the SWICST bit, or generate an TRIGSEL trigger for the inserted sequence.
6. Repeat step5 if in need.

18.4.5. 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 18-7. Auto-insertion, CNT = 1

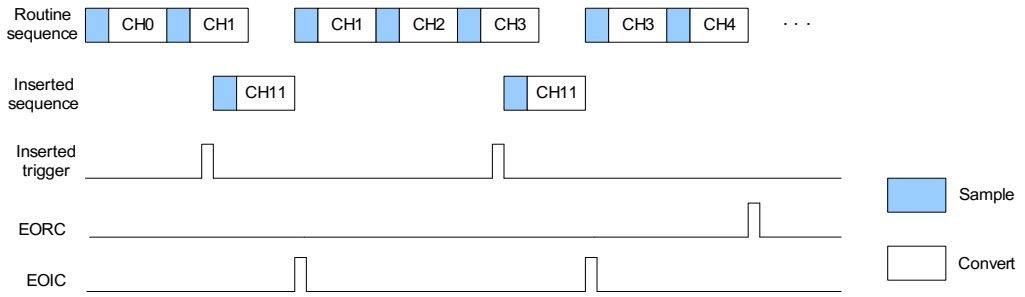


The auto insertion mode cannot be enabled when the discontinuous operation mode is set.

Triggered insertion

If the ICA bit is cleared, the triggered insertion occurs if a software or TRIGSEL 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.

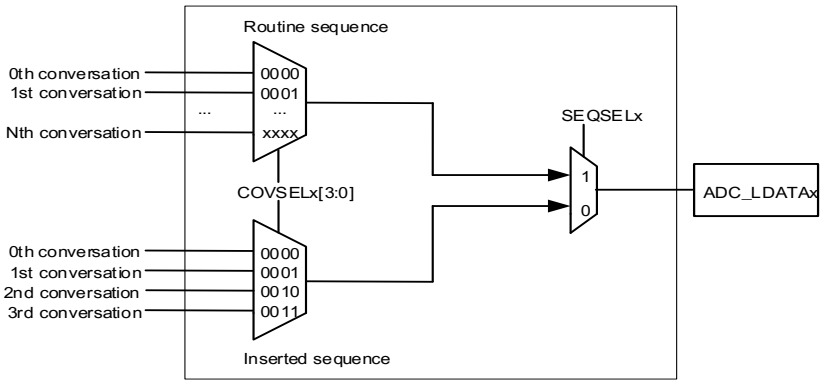
Figure 18-8. Triggered insertion



18.4.6. 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 18-9. Data latch***.

Figure 18-9. Data latch

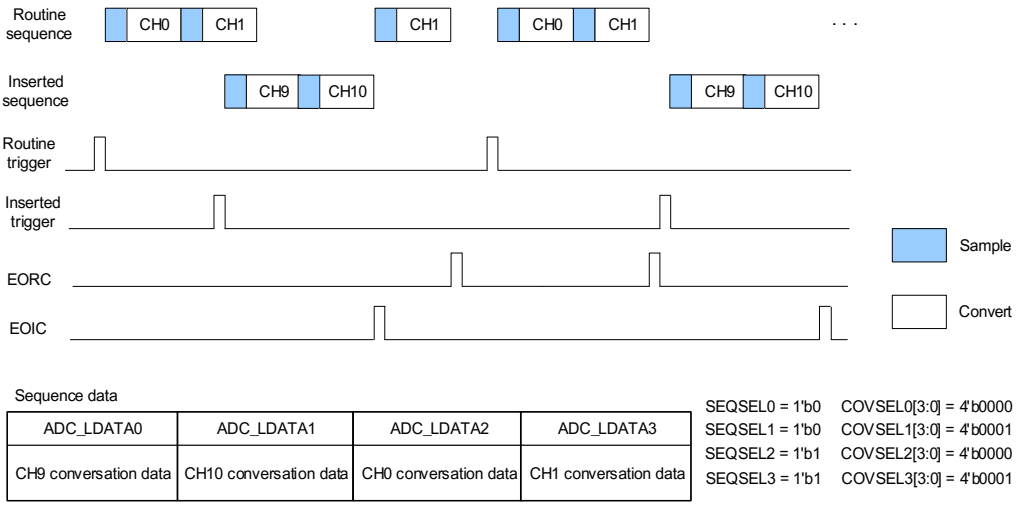


The SEQSELx (x=0…3) bit in the ADC_LDCTL register is used to select whether to latch data from a routine or insert sequence, while the COVSELx[3:0] (x=0…3) bits selects which conversion result in the sequence to latch.

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 18-10. 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.

18.4.7. 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 WD0LE or WD0HE bit in ADC_STAT register will be set if the conversion result crosses the low threshold or the high threshold. An interrupt will be generated if the WD0LEIE or WD0HEIE 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.

Analog watchdog 1/2

The analog watchdog 1/2 are more flexible, and can configure the watchdog function of single or several channels.

The analog watchdog 1 function can be enabled by configuring the corresponding bits in the AWD1CS[21:0] bits in the ADC_WD1SR register. Similarly, the watchdog 2 function can be configured. The high / low threshold of the analog watchdog 1/2 can be configured in the ADC_WD1LT, ADC_WD1HT, ADC_WD2LT and ADC_WD2HT registers.

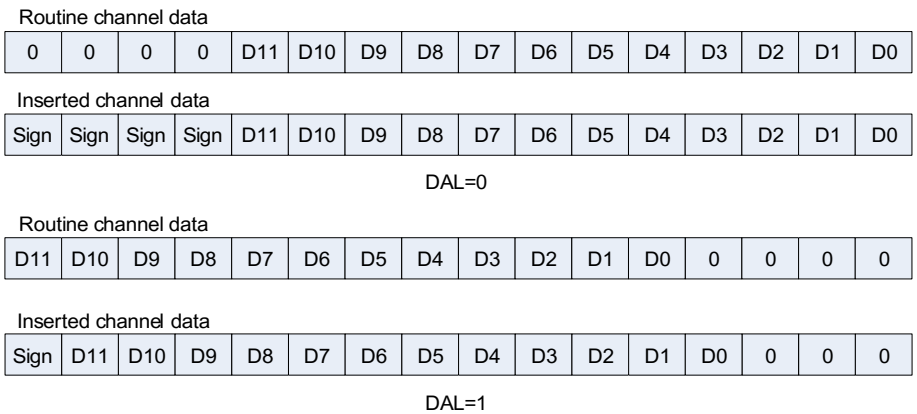
Note: If OVSEN = 1, analog watchdog 0/1/2 can compare the analog voltage converted (after oversample) with a low threshold or a high threshold. If OVSEN = 0, analog watchdog 0/1/2 can compare the analog voltage converted (before oversample) with a low threshold or a high threshold.

18.4.8. Data storage mode

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

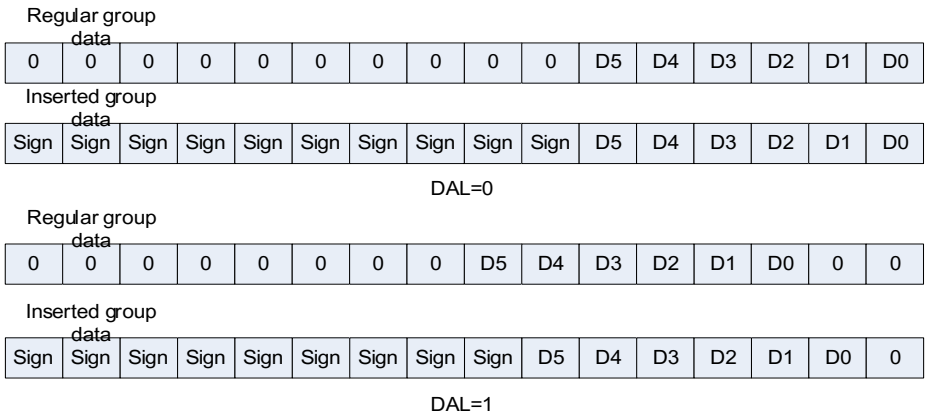
After being decreased by the user-defined offset written in the ADC_IOFFx registers, the inserted sequence data value may be a negative value. The sign bit is extended.

Figure 18-11. 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 18-12. Data storage mode of 6-bit resolution***.

Figure 18-12. Data storage mode of 6-bit resolution



18.4.9. 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_SAMPTx(x=0⋯2) 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.

18.4.10. 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. The external trigger source of routine sequence and inserted sequence is controlled by TRIGSEL module.

Table 18-3. External trigger modes and trigger type

ETMRC[1:0]/ETMIC[1:0]	Trigger mode	Trigger Type
00	Rising edge of external trigger enable	Hardware trigger: TRIGSEL
01		
10		
11	External trigger disable	Software trigger: SWRCST/SWICST

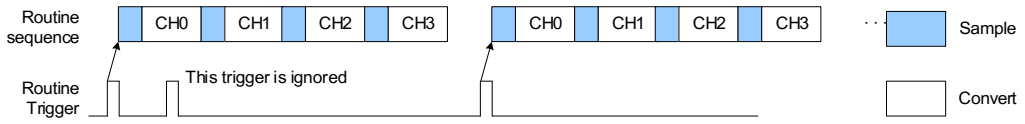
The selection of the external triggers can be changed on the fly, while no trigger event occurs due to this change.

■ ITGRECEN = 0, RTGRECEN = 0

During the inserted sequence conversion, if a new inserted or routine sequence trigger

request occurs, or during the routine sequence conversion, if a new routine sequence trigger request occurs, the subsequent trigger requests will be ignored, as shown in ***Figure 18-13. Routine trigger occurs during routine conversion (ITGRECEN = 0, RTGRECEN = 0)***.

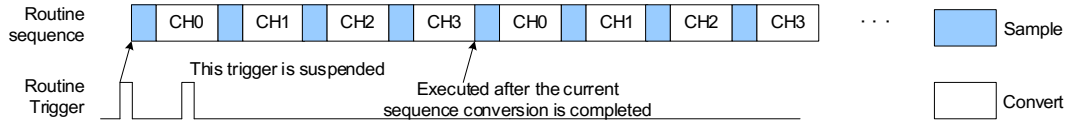
Figure 18-13. Routine trigger occurs during routine conversion (ITGRECEN = 0, RTGRECEN = 0)



■ RTGRECEN = 1

During the routine sequence conversion, if a new routine sequence trigger request occurs, the trigger request will be suspended and executed after the current conversion is completed, as shown in ***Figure 18-14. Routine trigger occurs during routine conversion (RTGRECEN = 1)***.

Figure 18-14. Routine trigger occurs during routine conversion (RTGRECEN = 1)



■ ITGRECEN = 1, RTGRECEN = 1

During the inserted sequence conversion, if new inserted sequence or routine sequence trigger requests occur, these trigger requests will also be suspended and executed after the current conversion is completed. When both inserted sequence and routine sequence trigger requests are suspended, the ADC will prioritize the inserted sequence conversion. After the inserted sequence conversion is completed, the routine sequence conversion will then be executed.

Each sequence (inserted sequence or routine sequence) can have at most one suspended trigger request. Any subsequent trigger requests will be ignored.

Note: In ADC sync mode, this function is only configurable in three modes: routine parallel mode, inserted parallel mode, combined routine parallel & inserted parallel mode.

The trigger error detection is enabled when the RTGERREN and ITGERREN bits in the ADC_CTL1 register are set for routine and inserted sequences respectively.

The RTGERRF bit in ADC_STAT register will be set if one trigger signal of routine sequence occurs when routine sequence is in the process of conversion. An interrupt will be generated if the RTGERRIE bit is set. The ITGERRF bit in ADC_STAT register will be set if one trigger signal of inserted sequence occurs when inserted sequence is in the process of conversion. An interrupt will be generated if the ITGERRIE bit is set.

18.4.11. 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.

18.4.12. Overflow detection

Overflow detection is enabled when DMA is enabled. An overflow event occurs when a routine or inserted conversion is done before the prior routine or inserted data has been read out. The ROVF or IOVF bit of the ADC_STAT is set. Overflow interrupt is generated if the ROVFIE or IOVFIE bit in the ADC_CTL0 is set.

It is recommended to reinitialize the DMA module to recover the ADC from overflow state. To ensure the routine converted data are transferred correctly, the internal state machine is reset. The ADC conversion will be stalled until the overflow status flag bit is cleared.

Note: If the ACST bit in the ADC_CTL1 register is set to 1, ADC conversion will stop when an overflow event occurs.

Software procedure for recovering the ADC from overflow state:
1. Clear RDMA or IDMA bit of ADC_CTL1 to 0
2. Clear ADCON bit of ADC_CTL1 to 0
3. Clear CHEN bit of DMA_CHxCTL to 0 with reinit DMA module
4. Clear ROVF or IOVF bit of ADC_STAT to 0, clear EORC or EOIC bit of ADC_STAT to 0
5. Set CHEN bit of DMA_CHxCTL to 1
6. Set RDMA or IDMA bit of ADC_CTL1 to 1
7. Set ADCON bit of ADC_CTL1 to 1
8. Wait T(setup)
9. Start conversion with software or trigger.

18.4.13. ADC internal channels

When the INREFEN bit in the ADC_CTL1 register is set, the internal reference voltage V_{REFINT} channel (ADC0_CH17,ADC1_CH18,ADC2_CH17) is enabled. The internal reference voltage (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and Comparators. V_{REFINT} is internally connected to the input channel(ADC0_CH17,ADC1_CH18,ADC2_CH17).

When the TSVEN bit of ADC_CTL1 register is set, the temperature sensor channel (ADC0_CH16) 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. Enable the ADC by setting the ADCON bit, and start the ADC conversion by software or TRIGSEL trigger.
4. Read and calculate internal temperature sensor output voltage (V_{SENSE}) in the ADC data register, and get the temperature using the following equation:

$$\text{Temperature (°C)} = \{(V_{25} - V_{SENSE}) / \text{Avg_Slope}\} + 25 \quad (18\text{-}1)$$

V_{25}: The voltage of the internal temperature sensor at 25°C, typical value and the factory calibration value address can be found in the Datasheet (see the Temperature sensor characteristics section).

Avg_Slope: The mean slope of the temperature versus the internal temperature sensor voltage curve, typical values can be found in the Datasheet (refer to the Temperature sensor characteristics section).

18.4.14. Battery voltage monitoring

The V_{BAT} channel can be used to measure the backup battery voltage on the V_{BAT} pin. When the VBATEN bit of the ADC_CTL1 register is set, V_{BAT} channel (ADC0_CH18) is enabled and a bridge divider by 4 integrated on the V_{BAT} pin is also enabled automatically with it. As V_{BAT} may be higher than V_{DDA}, this bridge is used to ensure the ADC correct operation. And it connects $V_{BAT}/4$ to the ADC0_CH18 input channel. So, the converted digital value is $V_{BAT}/4$. In order to prevent unnecessary battery energy consumption, it is recommended that the bridge will be enabled only when it is required.

18.4.15. 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 18-4. tCONV timings depending on resolution* .

Table 18-4. 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

18.4.16. On-chip hardware oversampling

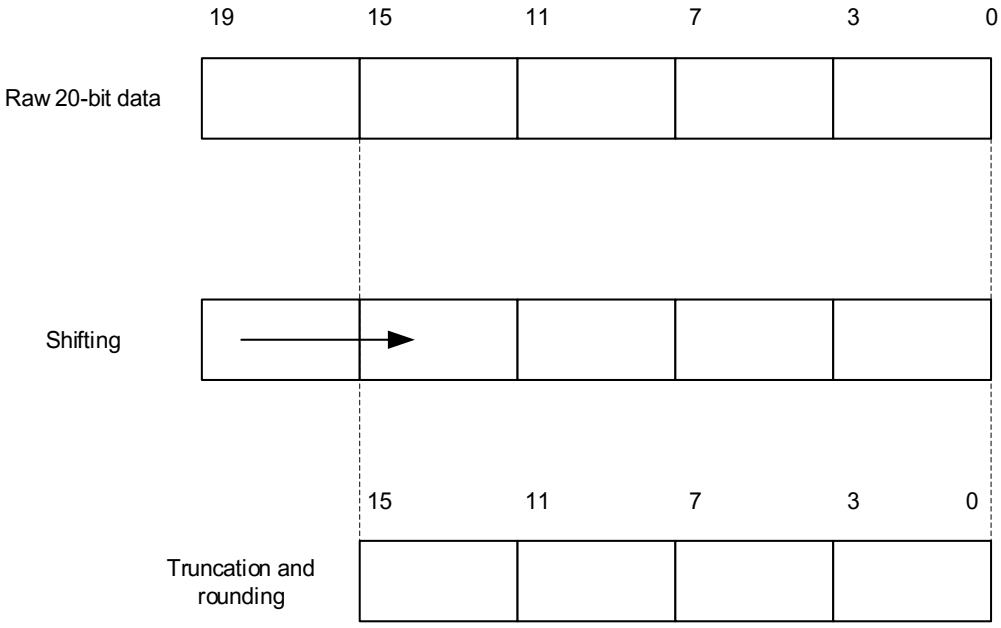
The on-chip hardware oversampling circuit performs data preprocessing to offload the CPU. It can handle multiple conversions and average them into a single data with increased data width, up to 16-bit. The on-chip hardware oversampling circuit is enabled by OVSEN bit in the ADC_OVSAMPCTL register.It provides a result with the following form, where N and M can be adjusted, and $D_{out}(n)$ is the n-th output digital signal of the ADC:

$$Result=\frac{1}{M}*\sum_{n=0}^{N-1} D_{out}(n) \quad (18\text{-}2)$$

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 18-15. 20-bit to 16-bit result truncation*.

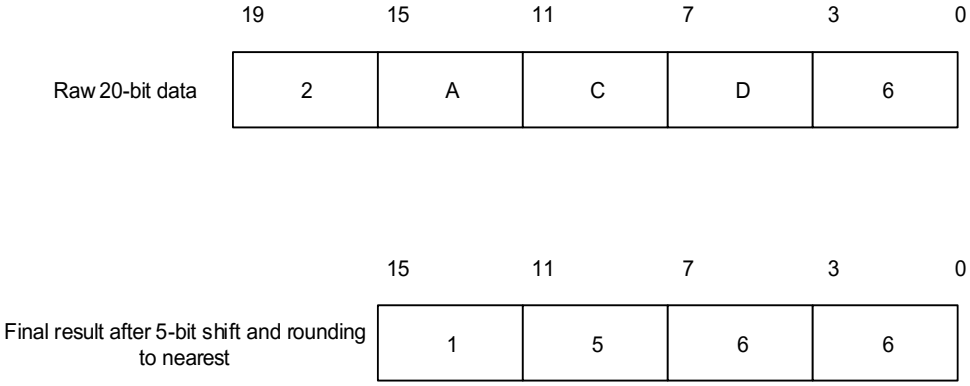
Figure 18-15. 20-bit to 16-bit result truncation



Note: If the intermediate result after the shifting exceeds 16 bits, the upper bits of the result are simply truncated.

Figure 18-16. 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 18-16. Numerical example with 5-bits shift and rounding



The *Table 18-5. 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 18-5. 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:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (18\text{-}3)$$

18.5. 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 ADC sync mode, when the conversion is configured to be triggered by an external event, the external trigger must be disabled for ADC1.

The modes in ***Table 18-6. 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).

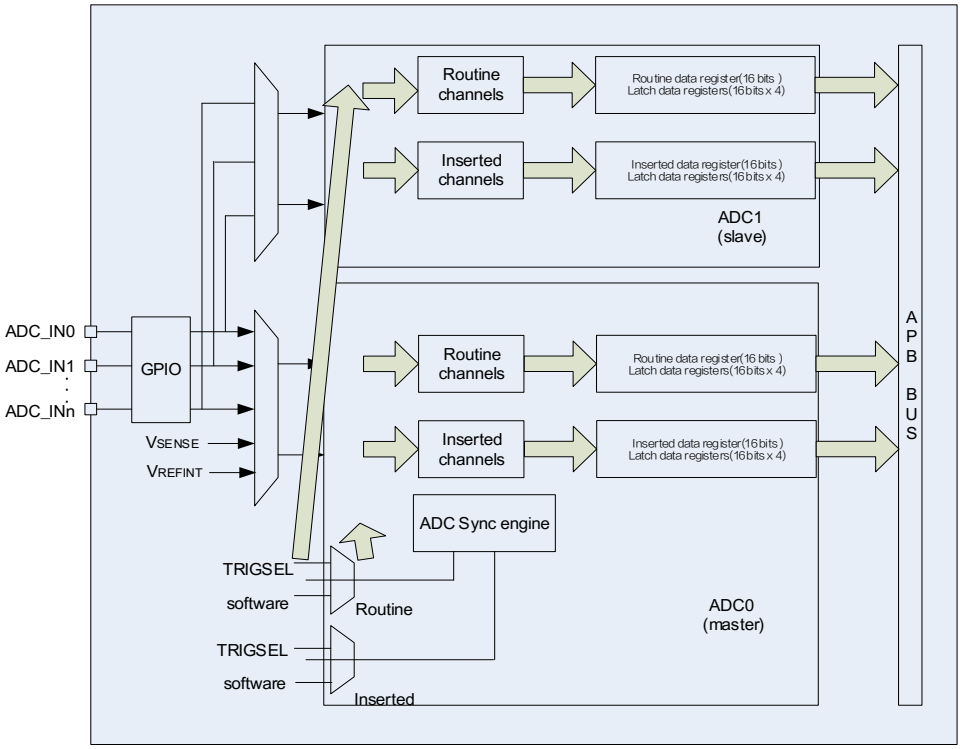
Table 18-6. ADC sync mode table

SYNCM[3: 0]	Mode
0000	Free mode. All ADCs work independently.
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 fast follow-up mode.
0100	ADC0 and ADC1 work in combined inserted parallel & routine slow follow-up 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 fast follow-up mode.
1000	ADC0 and ADC1 work in routine slow follow-up 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. Inserted data register (ADC_IDATA) should not be used for DMA transfer.

The ADC sync scheme is shown in ***Figure 18-17. ADC sync block diagram*** .

Figure 18-17. ADC sync block diagram



18.5.1. Free mode

In this mode, each ADC works independently and does not interfere with each other.

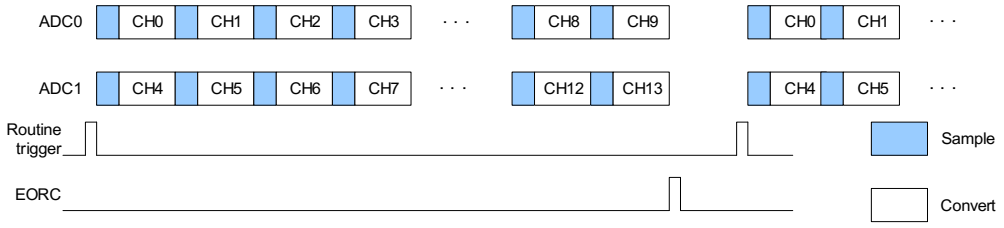
18.5.2. Routine parallel mode

The routine parallel mode is enabled by setting the SYNCM[3:0] bits in the ADC0_CTL0 register to 4'b0110. In the routine parallel mode, all of the ADCs convert the routine sequence in parallel at the selected external trigger of ADC0. The trigger is selected by configuring the ETMRC[1:0] bits in the ADC_CTL1 register of ADC0.

At the end of ADC routine sequence conversion, an EORC interrupt is generated (if any ADC interrupt is enabled for EORC). The behavior of routine parallel mode is shown in the ***Figure 18-18. Routine parallel mode on 10 channels*** .

A 32-bit DMA is used, which transfers the 32-bit ADC_RDATA register (the 32-bit ADC_RDATA register containing the ADC1 converted data in the upper half-word and the ADC0 converted data in the lower half-word) to SRAM.

Figure 18-18. Routine parallel mode on 10 channels



Note:

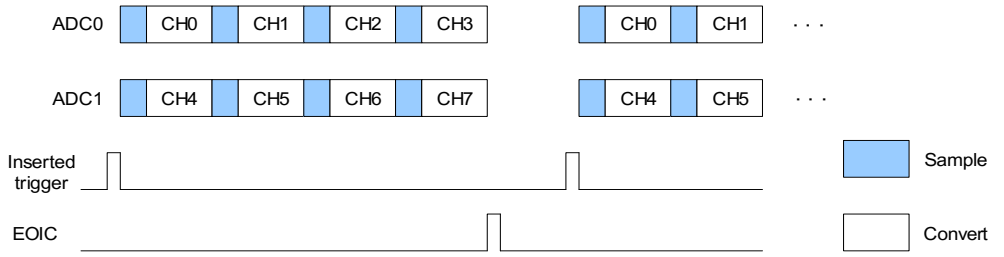
1. If two ADCs use the same sampling channel, it should be ensured that the channel is not used at the same time.
2. Two channels sampled by two ADCs at the same time should be configured with the same sampling time.
3. Make sure to do ADC peripheral reset before ADCON is set to 1.

18.5.3. 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 18-19. Inserted parallel mode on 4 channels*** .

Figure 18-19. 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.
3. Inserted data register (ADC_IDATA) should not be used for DMA transfer.

18.5.4. 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.

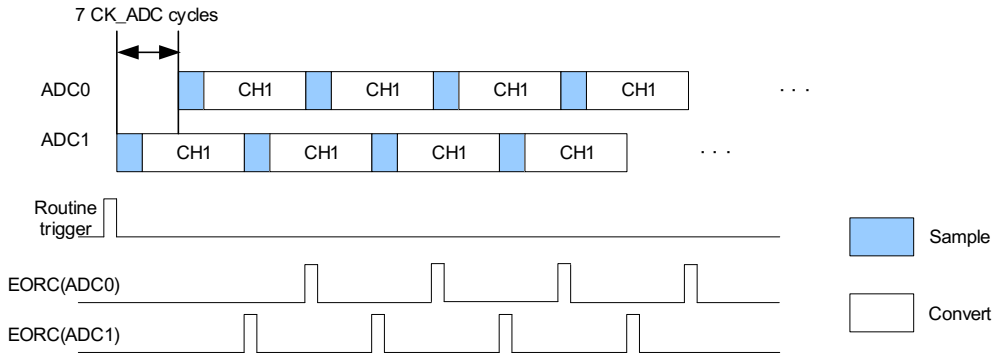
If the continuous mode is enabled for both ADC0 and ADC1, the selected routine channels of two ADCs are continuously converted. The behavior of follow-up fast mode shows in the ***Figure 18-20. Routine follow-up fast mode (the CTN bit of ADCs are set)***.

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 18-20. Routine follow-up fast mode (the CTN bit of ADCs are set)



18.5.5. 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.

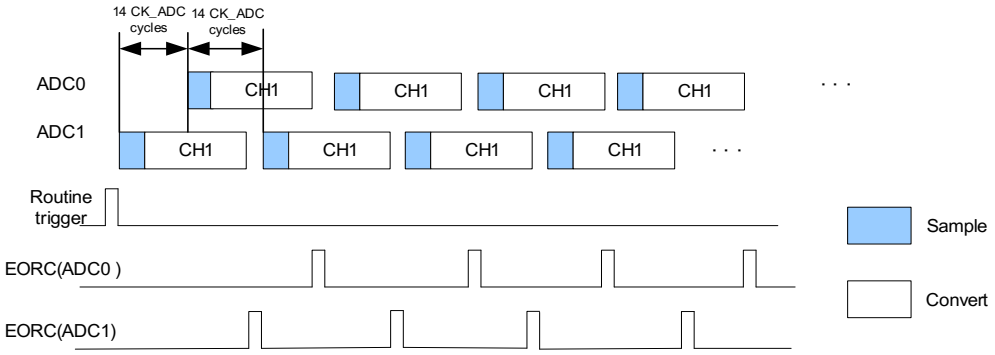
Continuous mode can't be used in this mode, because it continuously converts the routine channel. The behavior of follow-up slow mode shows in the ***Figure 18-21. Routine follow-up slow mode***.

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 18-21. Routine follow-up slow mode

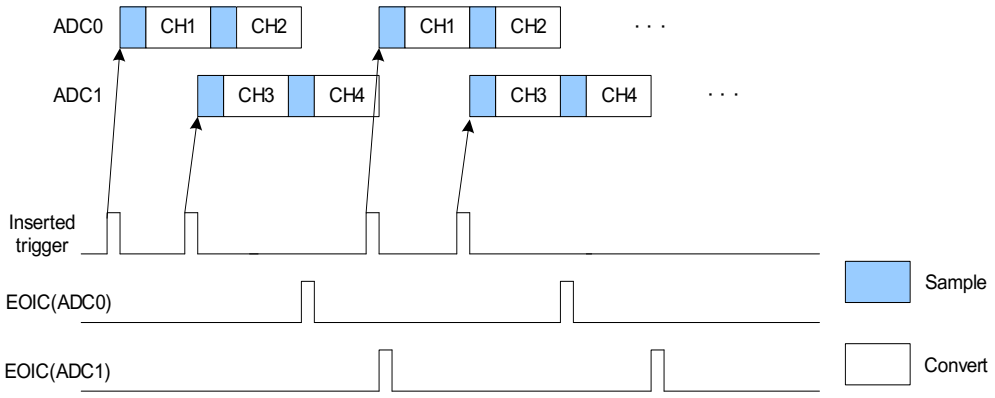


18.5.6. Inserted trigger rotation mode

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 18-22. Inserted trigger rotation: DISIC=0, IL=1* when DISIC bit is 0.

Figure 18-22. Inserted trigger rotation: DISIC=0, IL=1

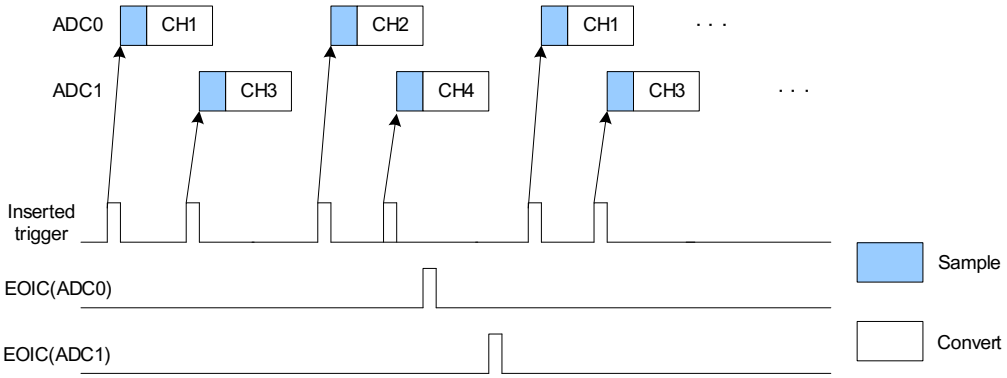


If the discontinuous operation mode is enabled for both ADC0 and ADC1, when the first

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 18-23. Inserted trigger rotation: DISIC=1, IL=1* when DISIC bit is 1.

Figure 18-23. 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).
2. Inserted data register (ADC_IDATA) should not be used for DMA transfer.

18.5.7. 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.
3. Inserted data register (ADC_IDATA) should not be used for DMA transfer.

18.5.8. Combined routine parallel & inserted trigger rotation mode

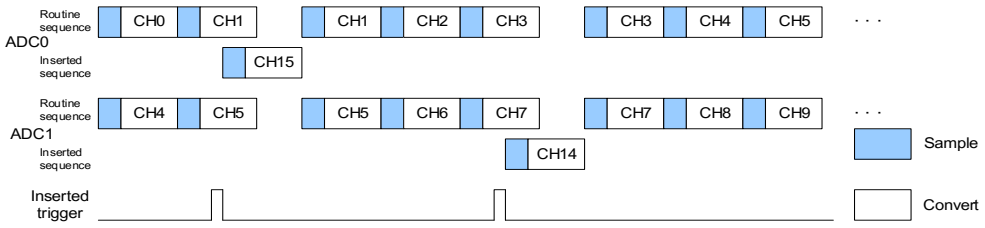
The combined routine parallel & inserted trigger rotation mode is enabled by setting the

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 18-24. 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 .

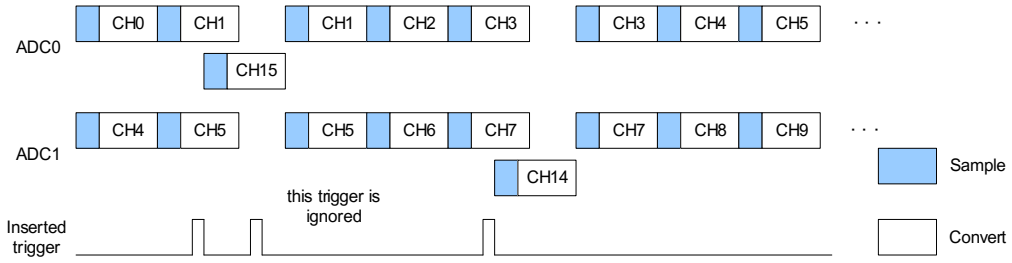
EOIC interrupts (if enabled on the ADC interfaces) are generated at the end of the inserted sequences.

Figure 18-24. 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 18-25. Trigger occurs during inserted conversion: SYNCM = 4'b0010*.

Figure 18-25. Trigger occurs during inserted conversion: SYNCM = 4'b0010



Note:

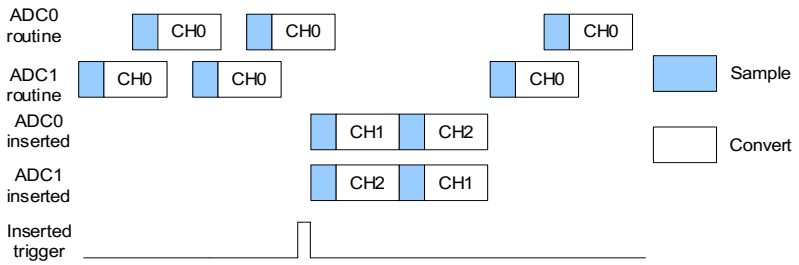
1. Make sure to do ADC peripheral reset before ADCON is set to 1.
2. Inserted data register (ADC_IDATA) should not be used for DMA transfer.

18.5.9. 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 18-26. Follow-*

up single channel with inserted sequence CH1, CH2 shows the behavior of this mode.

Figure 18-26. Follow-up single channel with inserted sequence CH1, CH2



Note:

1. Make sure to do ADC peripheral reset before ADCON is set to 1.
2. Inserted data register (ADC_IDATA) should not be used for DMA transfer.

18.6. 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.

18.7. Register definition

ADC0 base address: 0x4001 2400

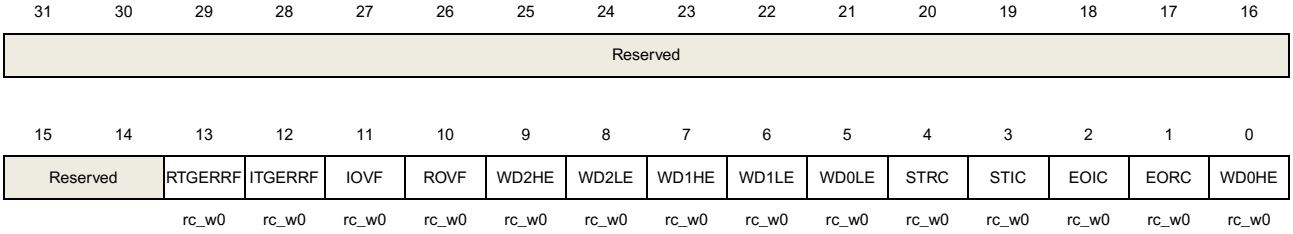
ADC1 base address: 0x4001 2800

ADC2 base address: 0x4001 3C00

18.7.1. Status register (ADC_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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		RTGERRF	ITGERRF	IOVF	ROVF	WD2HE	WD2LE	WD1HE	WD1LE	WD0LE	STRC	STIC	EOIC	EORC	WD0HE
		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	rc_w0

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	RTGERRF	Routine sequence trigger error flag 0: No trigger error 1: Trigger error occurred This bit is set by hardware when a false trigger of the routine sequence occurred. The routine sequence trigger error detection is valid only during the routine sequence is working. Cleared by software writing 0 to it.
12	ITGERRF	Inserted sequence trigger error flag 0: No trigger error 1: Trigger error occurred This bit is set by hardware when the a false trigger of inserted sequence occurred. The Inserted sequence trigger error detection is valid only during the inserted sequence is working. Cleared by software writing 0 to it.
11	IOVF	Inserted data register overflow flag 0: Inserted data register not overflow 1: Inserted data register overflow This bit is set by hardware when the inserted data registers are overflow. This flag is only set when DMA is enabled(IDMA=1). The recent inserted data is lost when this bit is set.

		Cleared by software writing 0 to it.
10	ROVF	Routine data register overflow flag 0: Routine data register not overflow 1: Routine data register overflow This bit is set by hardware when the routine data registers are overflow. This flag is only set when DMA is enabled(RDMA=1). The recent routine data is lost when this bit is set. Cleared by software writing 0 to it.
9	WD2HE	Analog watchdog 2 high threshold event flag 0: Analog watchdog 2 high threshold event is not happened 1: Analog watchdog 2 high threshold event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD2HT registers. Cleared by software writing 0 to it.
8	WD2LE	Analog watchdog 2 low threshold event flag 0: Analog watchdog 2 low threshold event is not happened 1: Analog watchdog 2 low threshold event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD2LT registers. Cleared by software writing 0 to it.
7	WD1HE	Analog watchdog 1 high threshold event flag 0: Analog watchdog 1 high threshold event is not happened 1: Analog watchdog 1 high threshold event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD1HT registers. Cleared by software writing 0 to it.
6	WD1LE	Analog watchdog 1 low threshold event flag 0: Analog watchdog 1 low threshold event is not happened 1: Analog watchdog 1 low threshold event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD1LT registers. Cleared by software writing 0 to it.
5	WD0LE	Analog watchdog 0 low threshold event flag 0: Analog watchdog 0 low threshold event is not happened 1: Analog watchdog 0 low threshold event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD0LT registers. Cleared by software writing 0 to it.
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	WD0HE	Analog watchdog 0 high threshold event flag 0: Analog watchdog 0 high threshold event is not happened 1: Analog watchdog 0 high threshold event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WD0HT registers. Cleared by software writing 0 to it.

18.7.2. 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
IOVFIE	ROVFIE	WD2HEIE	WD2LEIE	WD1HEIE	WD1LEIE	WD0HEIE	WD0LEIE	RWD0EN	IWD0EN	ITGERRIE	RTGERRIE	SYNCM[3:0]			
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
DISNUM[2:0]			DISIC	DISRC	ICA	WD0SC	SM	EOICIE	Reserved	EORCIE	WD0CHSEL[4:0]				
rw			rw	rw	rw	rw	rw	rw		rw	rw				

Bits	Fields	Descriptions
31	IOVFIE	Interrupt enable for IOVF

		0: IOVF interrupt disable 1: IOVF interrupt enable
30	ROVFIE	Interrupt enable for ROVF 0: ROVF interrupt disable 1: ROVF interrupt enable
29	WD2HEIE	Interrupt enable for WD2HE 0: WD2HE interrupt disable 1: WD2HE interrupt enable
28	WD2LEIE	Interrupt enable for WD2LE 0: WD2LE interrupt disable 1: WD2LE interrupt enable
27	WD1HEIE	Interrupt enable for WD1HE 0: WD1HE interrupt disable 1: WD1HE interrupt enable
26	WD1LEIE	Interrupt enable for WD1LE 0: WD1LE interrupt disable 1: WD1LE interrupt enable
25	WD0HEIE	Interrupt enable for WD0HE 0: WD0HE interrupt disable 1: WD0HE interrupt enable
24	WD0LEIE	Interrupt enable for WD0LE 0: WD0LE interrupt disable 1: WD0LE interrupt enable
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	ITGERRIE	Interrupt enable for ITGERRF 0: ITGERRF interrupt disable 1: ITGERRF interrupt enable
20	RTGERRIE	Interrupt enable for RTGERRF 0: RTGERRF interrupt disable 1: RTGERRF interrupt enable
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	Reserved	Must be kept at reset value.
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 10010: ADC channel 18 10011: ADC channel 19 10100: ADC channel 20 Other values are reserved. **Note:** 1. ADC0 analog inputs channel16,17,18,19,20 are internally connected to the temperature sensor and analog inputs V_{REFINT},V_{BAT},V_{REFP},V_{REFN}, respectively. 2. ADC1 analog inputs channel18,19,20 are internally connected to V_{REFINT}, V_{REFP}, V_{REFN}, respectively. 3. ADC2 analog inputs channel17,19,20 are internally connected to V_{REFINT}, V_{REFP}, V_{REFN}, respectively.

18.7.3. 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		ITGERREN	RTGERREN	ITGRECEN	RTGRECEN	VBATEN	TSVEN	INREFEN	SWRCST	SWICST	Reserved	ETMRC[1:0]		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
Reserved	ETMIC[1:0]		Reserved	DAL	Reserved		RDMA	IDMA	Reserved				ACST	CTN	ADCON
	rw			rw			rw	rw					rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	ITGERREN	Inserted sequence trigger error detection enable 0: Inserted sequence trigger error detection disable 1: Inserted sequence trigger error detection enable **Note:** 1. The Inserted sequence trigger error detection is valid only during the inserted sequence is working. 2. In ADC sync mode, it is sufficient to set ITGERREN=1 of ADC1 and ITGERREN=0 of ADC2. After a trigger error occurs, ITGERRF of ADC1 is set to 1, and ITGERRF remains 0 of ADC2.
28	RTGERREN	Routine sequence trigger error detection enable 0: Routine sequence trigger error detection disable 1: Routine sequence trigger error detection enable **Note:** 1. The routine sequence trigger error detection is valid only during the routine sequence is working. 2. In ADC sync mode, it is sufficient to set RTGERREN=1 of ADC1 and RTGERREN=0 of ADC2. After a trigger error occurs, RTGERRF of ADC1 is set to 1, and RTGERRF remains 0 of ADC2.
27	ITGRECEN	Inserted sequence trigger recover enable 0: Inserted sequence trigger will not be stored and recover. 1: Inserted sequence trigger will be stored and recover. **Note:** 1. The Inserted sequence trigger recover is valid only during the inserted sequence is working. 2. Only one trigger can be stored. 3. In ADC sync mode, this function is only configurable in three modes: routine parallel mode, inserted parallel mode, combined routine parallel & inserted parallel mode.
26	RTGRECEN	Routine sequence trigger recover enable 0: Routine sequence trigger will not be stored and recover. 1: Routine sequence trigger will be stored and recover. **Note:** 1. The Routine sequence trigger recover is valid only during the routine sequence is working. 2. Only one trigger can be stored. 3. In ADC sync mode, this function is only configurable in three modes: routine parallel mode, inserted parallel mode, combined routine parallel & inserted parallel mode.

		4. In continuous mode (CTN=1), this bit must be set to 0, the trigger suspend function for the routine sequence is not supported, and RTGERREN is required to be 0.
25	VBATEN	This bit can be set or cleared by software in ADC0. Channel 18 (battery power supply voltage V_{BAT}) enable of ADC0. 0: Battery power supply voltage channel disable 1: Battery power supply voltage channel enable
24	TSVEN	This bit can be set or cleared by software in ADC0. Channel 16 (temperature sensor) enable of ADC0. 0: Temperature sensor channel disable 1: Temperature sensor channel enable
23	INREFEN	This bit can be set or cleared by software in ADC0. Channel17 (internal reference voltage V_{REFINT}) enable of ADC0 Channel18 (internal reference voltage V_{REFINT}) enable of ADC1 Channel17 (internal reference voltage V_{REFINT}) enable of ADC2 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 **Note:** In sync mode, this bit remains 0, and DMA request for inserted sequence is disabled.
6:3	Reserved	Must be kept at reset value.
2	ACST	ADC conversion stop when the inserted or routine data registers is overflow. 0: The ADC conversion does not stop when the inserted or routine data registers is overflow. 1: The ADC conversion stop when the inserted or routine data registers is overflow. **Note:** 1. If ACST = 0， user can handle overflow event in the overflow interrupt. 2. When ACST = 1, after the ADC conversion stops, if the overflow flag ROVF or IOVF is not cleared, all subsequent triggers are blocked. Once ROVF or IOVF is cleared, the ADC continues to be triggered. 3. In ADC sync mode, if an overflow event occurs in the routine sequence (ROVF=1 in the ADC1_STAT register) and ACST=0, the ADC conversion is not affected. 4. In ADC sync mode, if an overflow event occurs in the routine sequence (ROVF=1 in the ADC1_STAT register) and ACST=1, both ADC0 and ADC1 stop conversion. 5. In ADC sync mode, if an overflow event occurs in the inserted sequence (IOVF=1 in the ADC1_STAT register) and ACST=0, the ADC conversion is not affected. 6. In ADC sync mode, if an overflow event occurs in the inserted sequence (IOVF=1 in the ADC1_STAT register) and ACST=1, both ADC0 and ADC1 stop conversion.
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. 0: ADC disable and power down 1: ADC enable

18.7.4. 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					SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:1]	
					rw			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
26:24	SPT18[2:0]	Refer to SPT10[2:0] description
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

18.7.5. 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

18.7.6. 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).

31	30	29	28	27	26	25	24	23	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				IOFF[11:0]											
				rw											

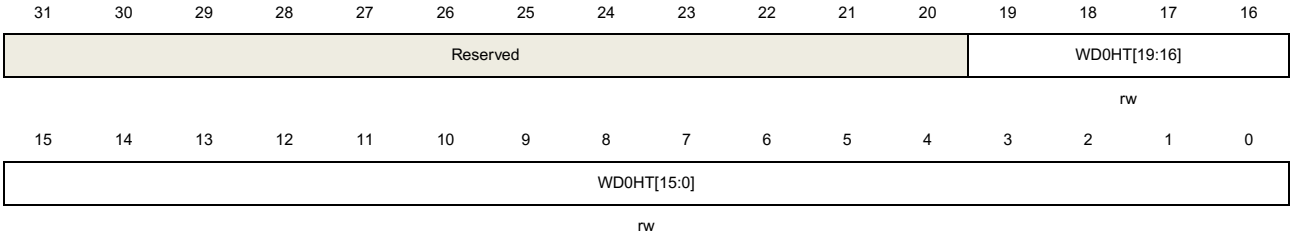
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	IOFF[11:0]	Data offset for inserted channel x These bits will be subtracted from the raw converted data when converting inserted channels. The conversion result can be read from in the ADC_IDATA register.

18.7.7. Watchdog 0 high threshold register (ADC_WD0HT)

Address offset: 0x24

Reset value: 0x000F FFFF

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



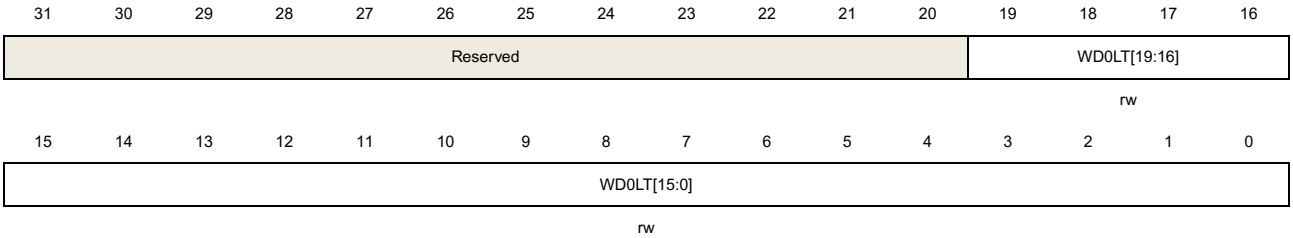
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD0HT[19:0]	High threshold for analog watchdog 0 These bits define the high threshold for the analog watchdog 0.

18.7.8. Watchdog 0 low threshold register (ADC_WD0LT)

Address offset: 0x28

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD0LT[19:0]	Low threshold for analog watchdog 0

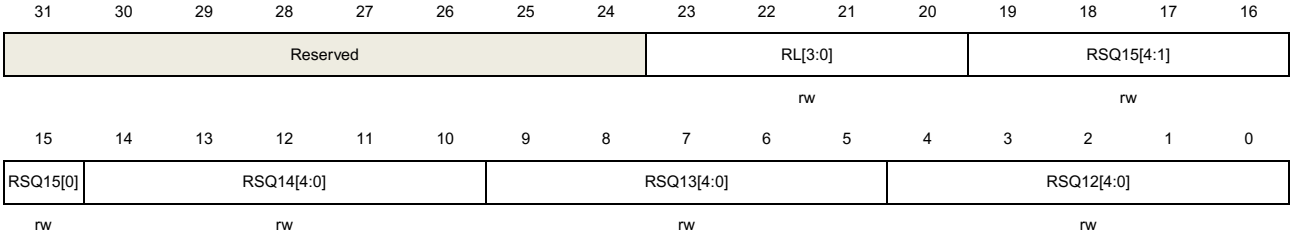
These bits define the low threshold for the analog watchdog 0.

18.7.9. Routine sequence register 0 (ADC_RSQ0)

Address offset: 0x2C

Reset value: 0x0000 0000

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



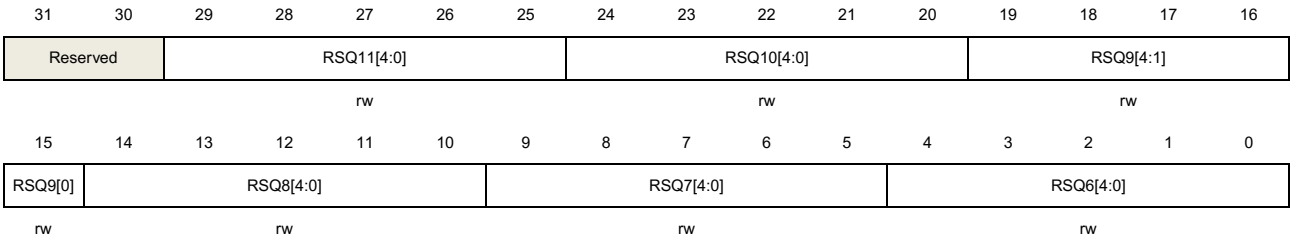
Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	RL[3:0]	Routine sequence length. The total number of conversion in routine sequence equals to RL[3:0]+1.
19:15	RSQ15[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ14[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ13[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ12[4:0]	Refer to RSQ0[4:0] description

18.7.10. Routine sequence register 1 (ADC_RSQ1)

Address offset: 0x30

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ11[4:0]	Refer to RSQ0[4:0] description

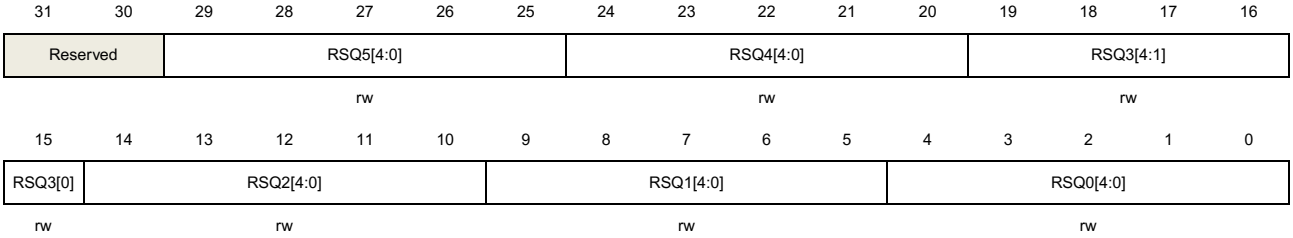
24:20	RSQ10[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ9[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ8[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ7[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ6[4:0]	Refer to RSQ0[4:0] description

18.7.11. Routine sequence register 2 (ADC_RSQ2)

Address offset: 0x34

Reset value: 0x0000 0000

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



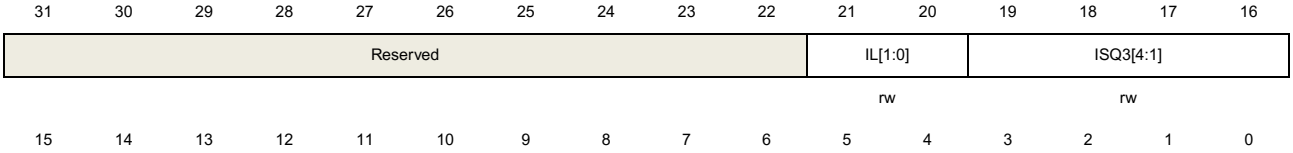
Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ5[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ4[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ3[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ2[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ1[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ0[4:0]	The channel number (0..20) is written to these bits to select a channel as the nth conversion in the routine sequence.

18.7.12. Inserted sequence register (ADC_ISQ)

Address offset: 0x38

Reset value: 0x0000 0000

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



ISQ3[0]	ISQ2[4:0]	ISQ1[4:0]	ISQ0[4:0]
rw	rw	rw	rw

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..20) 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

18.7.13. 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															

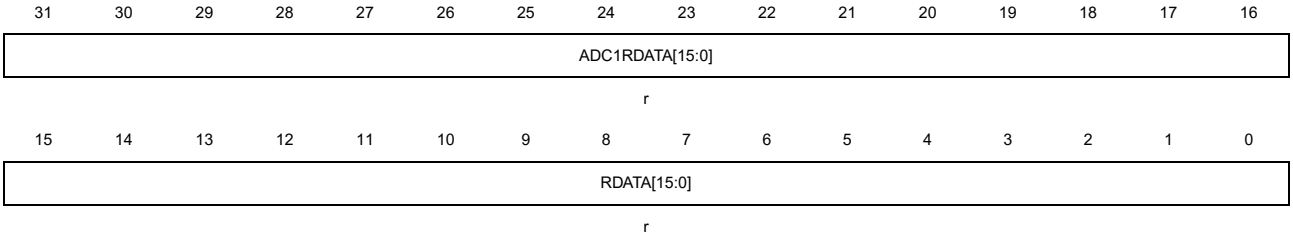
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	LDATAn[15:0]	Inserted or routine number n conversion data. These bits contain the number n conversion result, which is read only.

18.7.14. Routine data register (ADC_RDATA)

Address offset: 0x4C

Reset value: 0x0000 0000

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



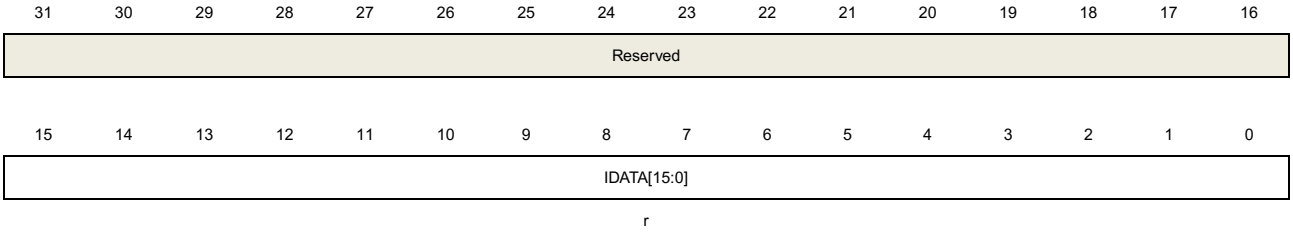
Bits	Fields	Descriptions
31:16	ADC1RDATA[15:0]	ADC1 routine channel data In ADC0: In sync mode, these bits contain the routine data of ADC1. In ADC1: These bits are reserved.
15:0	RDATA[15:0]	Routine channel data These bits contain routine channel conversion value, which is read only.

18.7.15. Inserted data register (ADC_IDATA)

Address offset: 0x50

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	IDATA[15:0]	Inserted channel data These bits contain the inserted channel conversion value, which is read only.

18.7.16. 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.

18.7.17. Watchdog 1 high threshold register (ADC_WD1HT)

Address offset: 0x60

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												WD1HT[19:16]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD1HT[15:0]															
rw															

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD1HT[19:0]	High threshold for analog watchdog 1 These bits define the high threshold for the analog watchdog 1.

18.7.18. Watchdog 1 low threshold register (ADC_WD1LT)

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												WD1LT[19:16]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WD1LT[15:0]															
rw															

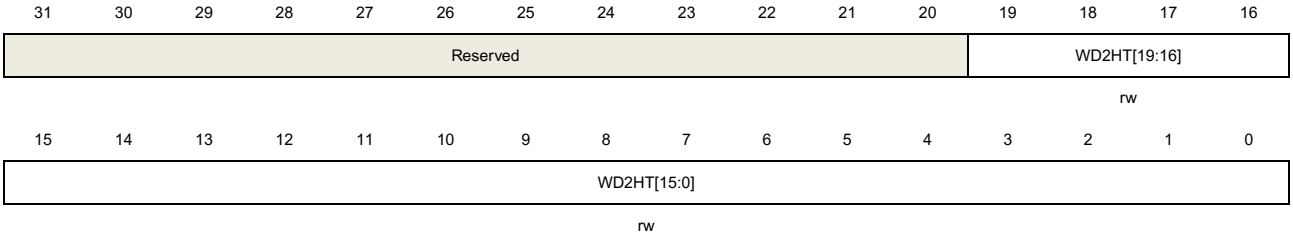
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD1LT[19:0]	Low threshold for analog watchdog 1 These bits define the low threshold for the analog watchdog 1.

18.7.19. Watchdog 2 high threshold register (ADC_WD2HT)

Address offset: 0x68

Reset value: 0x000F FFFF

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



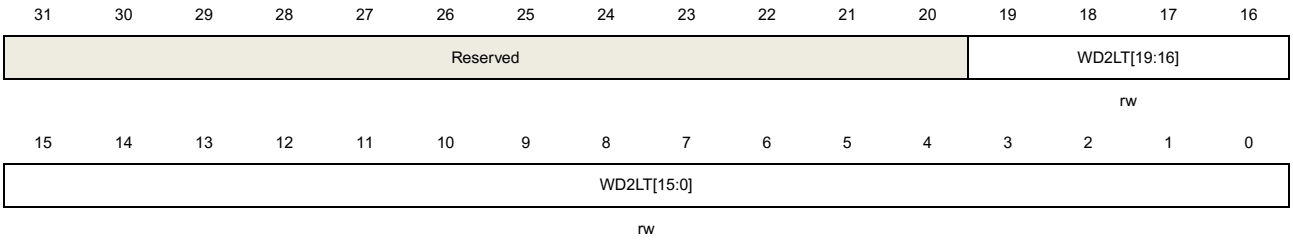
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD2HT[19:0]	High threshold for analog watchdog 2 These bits define the high threshold for the analog watchdog 2.

18.7.20. Watchdog 2 low threshold register (ADC_WD2LT)

Address offset: 0x6C

Reset value: 0x0000 0000

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



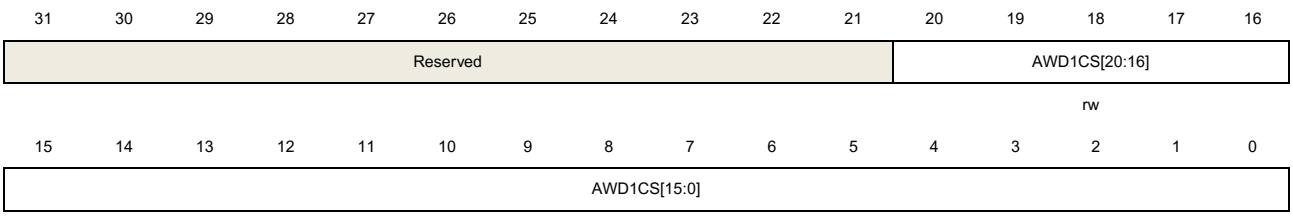
Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	WD2LT[19:0]	Low threshold for analog watchdog 2 These bits define the low threshold for the analog watchdog 2.

18.7.21. Watchdog 1 Channel Selection Register (ADC_WD1SR)

Address offset: 0x70

Reset value: 0x00000000

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



rw

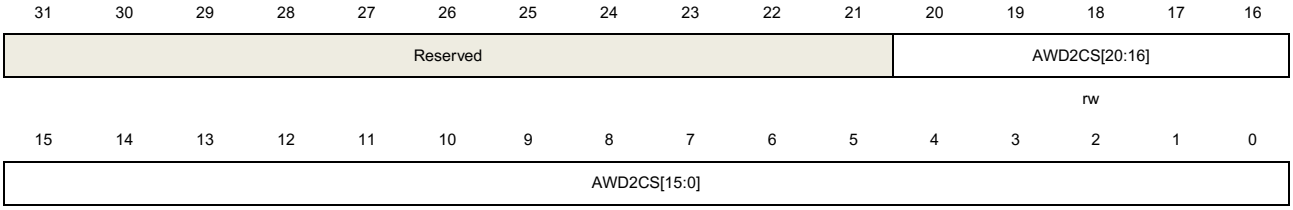
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:0	AWD1CS[20:0]	Analog watchdog 1 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 1. AWD1CS[n] = 0: ADC analog input channel n is not monitored by analog watchdog 1. AWD1CS[n] = 1: ADC analog input channel n is monitored by analog watchdog 1. When AWD1CS[20:0] = 000..0, the analog Watchdog 1 is disabled **Note:** 1) The channels selected by AWD1CS must be also selected into the ADC_RSQn or ADC_ISQ registers. 2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

18.7.22. Watchdog 2 Channel Selection Register (ADC_WD2SR)

Address offset: 0x74

Reset value: 0x00000000

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



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:0	AWD2CS[20:0]	Analog watchdog 2 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 2. AWD2CS[n] = 0: ADC analog input channel n is not monitored by analog watchdog 2. AWD2CS[n] = 1: ADC analog input channel n is monitored by Analog watchdog 2. When AWD2CS[20:0] = 000..0, the analog Watchdog 2 is disabled **Note:** 1) The channels selected by AWD2CS must be also selected into the ADC_RSQn or ADC_ISQ registers.

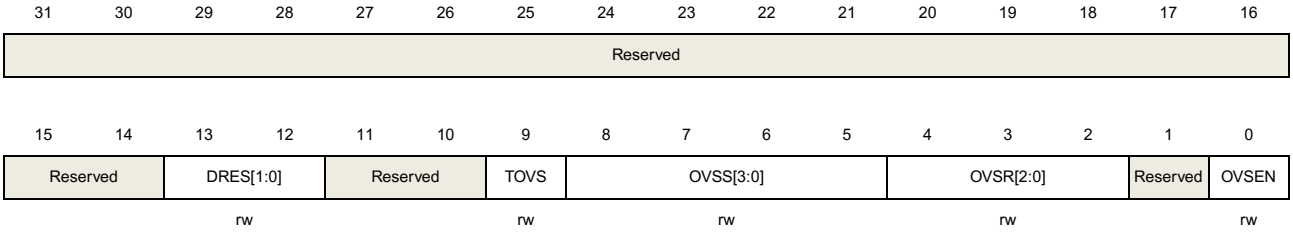
2) Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

18.7.23. 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

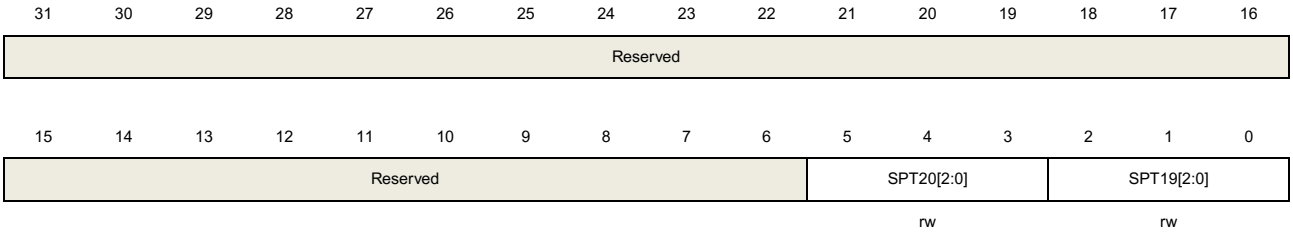
		Other values are reserved. **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
4:2	OVSR[2:0]	Oversampling ratio This bit filed defines the number of oversampling ratio. 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 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).

18.7.24. Sample time register 2 (ADC_SAMPT2)

Address offset: 0x84

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value
5:3	SPT20[2:0]	Refer to SPT19[2:0] description
2:0	SPT19[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

19. Digital-to-analog converter (DAC)

19.1. 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.

The DAC channels can work independently or concurrently.

19.2. Characteristics

The main features of DAC are as follows:

- 8-bit or 12-bit resolution.
- Left or right data alignment.
- DMA capability for each channel and underrun function.
- Conversion update synchronously.
- Conversion triggered by external triggers.
- Configurable internal buffer.
- Extern voltage reference, V_{REFP}.
- Noise wave generation (LFSR noise mode and triangle noise mode).
- Two DAC channels in concurrent mode.

Figure 19-1. DAC block diagram and *Table 19-1. DAC I/O description* show the block diagram of DAC and the pin description of DAC, respectively.

Figure 19-1. DAC block diagram

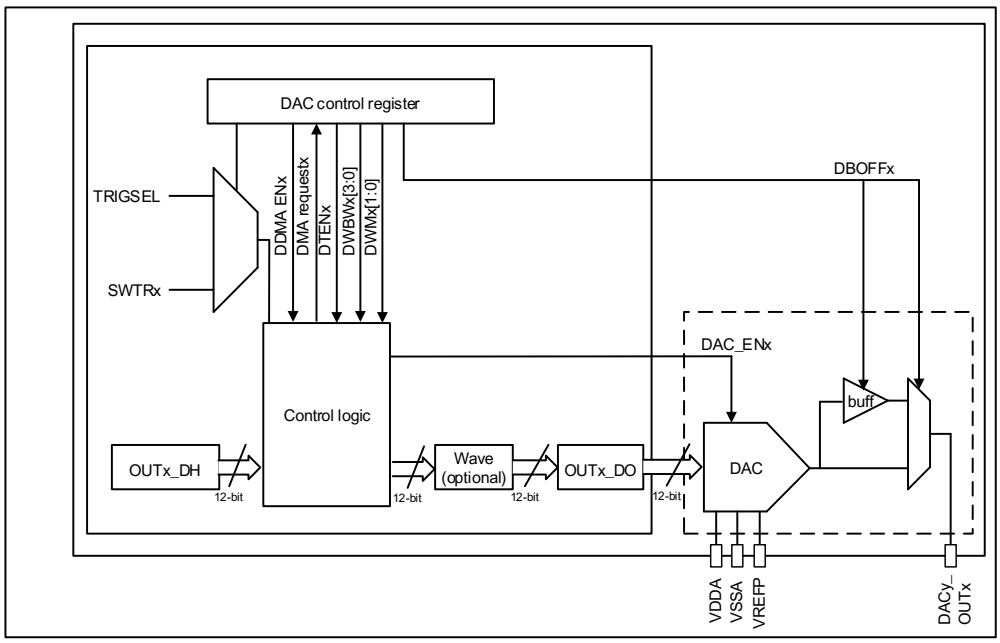


Table 19-1. DAC I/O description

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 19-2. DAC triggers and outputs summary

	DAC0	
Channel	**Channel0**	**Channel1**
DAC outputs connected to I / Os	PA4	PA5
DAC output buffer	•	•
DAC trigger signals from TRIGSEL	•	
DAC software trigger	•	

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

19.3. Function overview

19.3.1. 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.

19.3.2. 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.

19.3.3. 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.

19.3.4. 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 19-3. Triggers of DAC***.

Table 19-3. Triggers of DAC

DTSELx[1:0]	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 (DENx=1 in the DAC_CTL0 register).

19.3.5. 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.

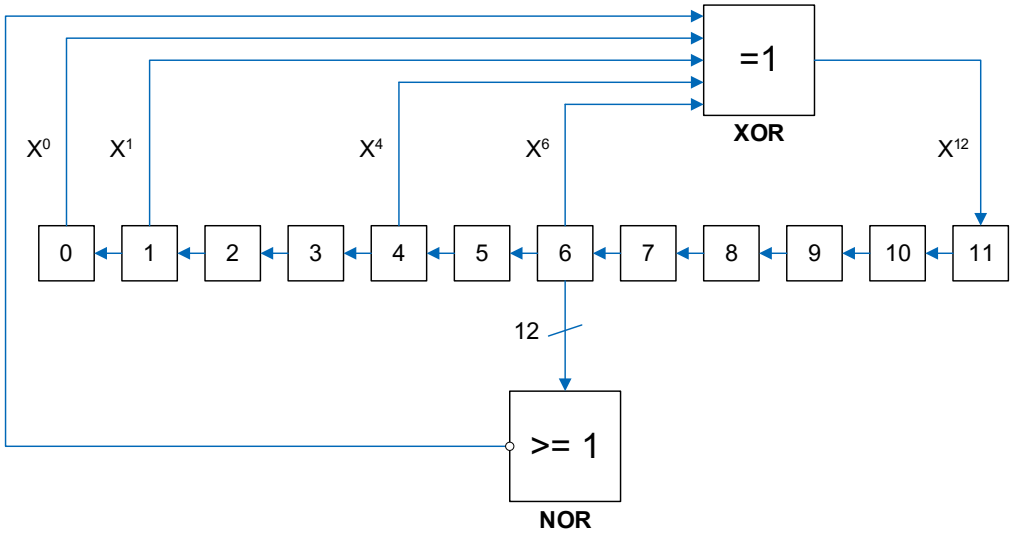
When the DAC holding data (OUTx_DH) is loaded into the DAC_OUTx_DO register, after the time $t_{SETTLING}$ which is determined by the analog output load and the power supply voltage, the analog output is valid.

19.3.6. 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.

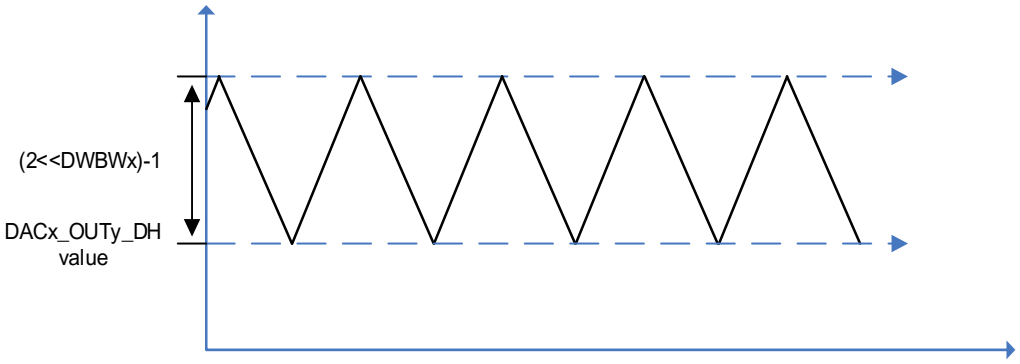
LFSR noise wave mode: there is a Linear Feedback Shift Register (LFSR) in the DAC control logic, it controls the LFSR noise signal which is added to the OUTx_DH value, and then the result is stored into the DAC_OUTx_DO register. When the configured DAC noise wave bit width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

Figure 19-2. 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 19-3. DAC triangle noise wave



19.3.7. DAC output voltage

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

$$V_{DAC_OUT}=V_{REFP}*OUTx_DO/4096 \quad (19\text{-}1)$$

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

19.3.8. 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.

19.3.9. DAC concurrent conversion

When the two output channels work at the same time, for maximum bus bandwidth utilization in specific applications, two output channels can be configured in concurrent mode. In concurrent mode, the OUTx_DH and OUTx_DO value will be updated at the same time.

There are three concurrent registers that can be used to load the OUTx_DH value: DACC_R8DH, DACC_R12DH and DACC_L12DH. User just need to access a unique register to realize driving two DAC channels at the same time.

When using DMA function, please ensure both DTENx bits be set, DTSEL0/DTSEL1 bits be same to guarantee the simultaneous trigger. and any DDMAENx bit in one DAC setting to enable DMA request.

The noise mode and noise bit width can be configured either the same or different, depending on the application scenario.

19.3.10. DAC output buffer calibration

The output voltage may be offset when DAC use buffer, so it is necessary to compensate

output voltage.

The DAC calibration transfer function is:

$$V_{out}=(D/2^N)*G*V_{REFP}+V_{of} \quad (19\text{-}2)$$

Where N is the significant digit of the DAC, D is the digital input of the DAC, and G is the gain, V_{REFP} is the positive reference voltage of the DAC, V_{of} is the offset voltage. The G is 1 and V_{of} is 0 for an ideal DAC.

Calibration will be effective when buffer is enabled. During the calibration:

- Buffer is disconnected from on chip peripherals and enters tri-stated.
- The buffer will be used as a comparator,and the internal bridge is configured to use V_{REFP}*255/4095 and V_{REFP}*3584/4095 as inputs. The CALFx bit in DAC_CALR register will be changed to 0 or 1 according to the comparison result.

Two calibration techniques are available:

- Factory calibration (always enable)
  - DAC buffer offset is adjusted at the factory. And the default value of OTVx_P[4:0] bits and OTVx_N[4:0] bits in DAC_CALR register are loaded when DAC reset.
- User calibration
  - When the operating conditions are different from the nominal factory calibration conditions, especially when the VDDA, V_{REFP} or temperature change, the user can calibrate at any point in the application process through the software.

The contents of OTVx_P[4:0] bits and OTVx_N[4:0] bitsc can be calculated in a faster way by using successive approximation or dichotomy techniques. The following process is successive approximation method:

- Writing 0 to DENx bit in DAC_CTL0 register to disable DAC output.
- Writing 0 to DBOFFx bit in DAC_CTL0 register to enable DAC output buffer.
- Writing 1 to CALENx bit in DAC_CALR register to enable DAC calibration.
- process of calibration algorithm
  - Writing 0xFF(255) to OUTx_DH[11:0] bit in DAC_OUTx_R12DH register.
  - Writing a code (staring by 0'b00000) into OTVx_P[4:0].
  - Waiting for t_{TRIM} delay.
  - Checking the flag CALFx in DAC_CALR register.
  - If the CALFx set to 1, it proves that the correct calibration value has been found; otherwise, add 1 to the code until the correct calibration value P is found.
  - Writing 0xE00(3584) to OUTx_DH[11:0] bit in DAC_OUTx_R12DH register.
  - Writing a code (staring by 0'b00000) into OTVx_N[4:0].
  - Waiting for t_{TRIM} delay.
  - Checking the flag CALFx in DAC_CALR register.
  - If the CALFx set to 1, it proves that the correct calibration value has been found; otherwise, add 1 to the code until the correct calibration value N is found.

Note: CALENx should write back to 0 after the calibration process, and then wirte DENx to 1 to use DAC at normal mode. It is forbidden to set DENx and CALENx to 1 at the same time.

19.4. Register definition

DAC0 base address: 0x4000 7400

19.4.1. DACx control register 0 (DAC_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	DCSEL1	DDUDRIE1	DDMAEN1	DWBW1[3:0]				DWM1[1:0]		Reserved	DTSEL1[1:0]		DTEN1	DBOFF1	DEN1
	rw	rw	rw	rw				rw			rw		rw	rw	rw

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

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30	DCSEL1	Connection selection of DACx_OUT1 with GPIO: 0: DACx_OUT1 is connected to GPIO 1: DACx_OUT1 is not connected to GPIO
29	DDUDRIE1	DACx_OUT1 DMA underrun interrupt enable 0: DACx_OUT1 DMA underrun interrupt disabled 1: DACx_OUT1 DMA underrun interrupt enabled
28	DDMAEN1	DACx_OUT1 DMA enable 0: DACx_OUT1 DMA mode disabled 1: DACx_OUT1 DMA mode enabled
27:24	DWBW1[3:0]	DACx_OUT1 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT1. 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 (DEN1=1).
23:22	DWM1[1:0]	DACx_OUT1 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT1 when external trigger of DACx_OUT1 is enabled (DTEN1=1). 00: Wave disabled 01: LFSR noise mode 1x: Triangle noise mode
21	Reserved	Must be kept at reset value
20:19	DTSEL1[1:0]	DACx_OUT1 trigger selection These bits are only used if bit DTEN1 = 1 and select the external event used to trigger DAC. 00: EXTRIG (external trigger from TRIGSEL) 11: Software trigger All other values: reserved.
18	DTEN1	DACx_OUT1 trigger enable 0: DACx_OUT1 trigger disabled 1: DACx_OUT1 trigger enabled
17	DBOFF1	DACx_OUT1 output buffer turn off 0: DACx_OUT1 output buffer turns on to reduce the output impedance and improve the driving capability 1: DACx_OUT1 output buffer turns off
16	DEN1	DACx_OUT1 enable 0: DACx_OUT1 disabled 1: DACx_OUT1 enabled
15	Reserved	Must be kept at reset value
14	DCSEL0	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 DTEN0 = 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

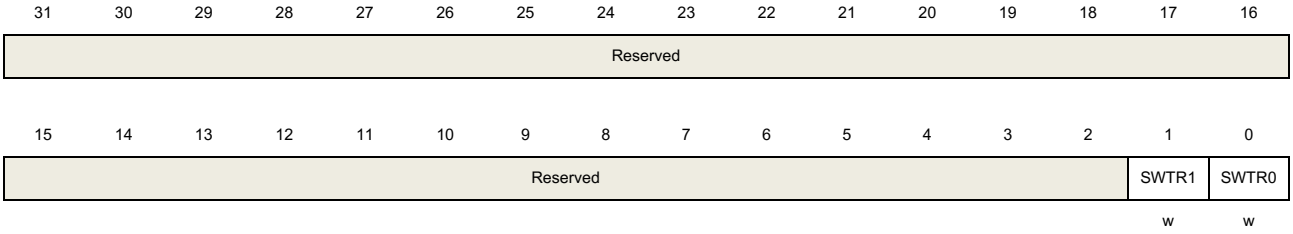
1: DACx_OUT0 enabled

19.4.2. DACx software trigger register (DAC_SWT)

Address offset: 0x04

Reset value: 0x0000 0000

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



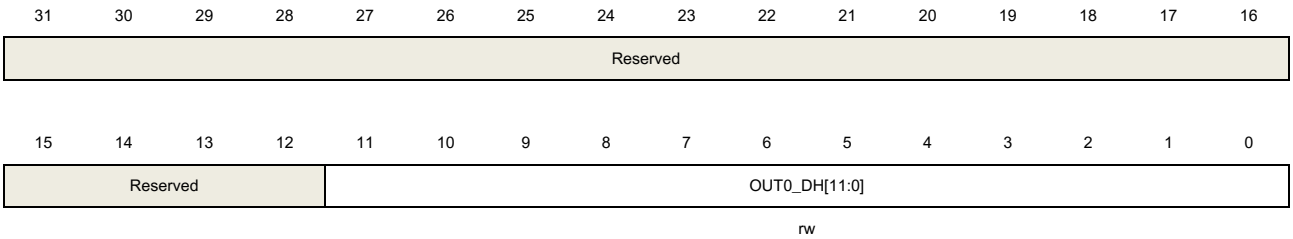
Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	SWTR1	DACx_OUT1 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled
0	SWTR0	DACx_OUT0 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled

19.4.3. 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.

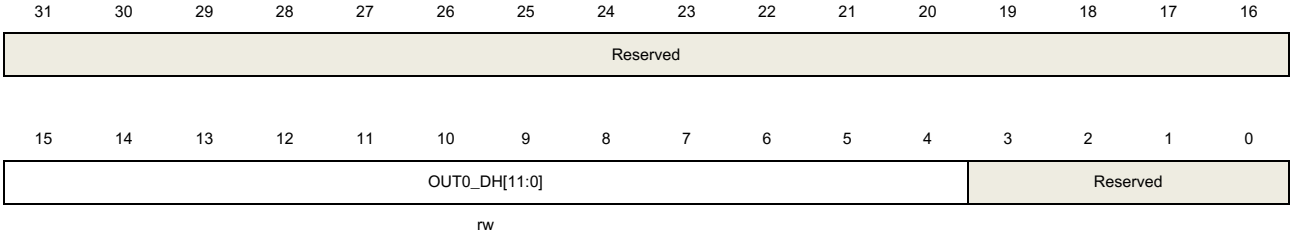
These bits specify the data that is to be converted by DACx_OUT0.

19.4.4. 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).



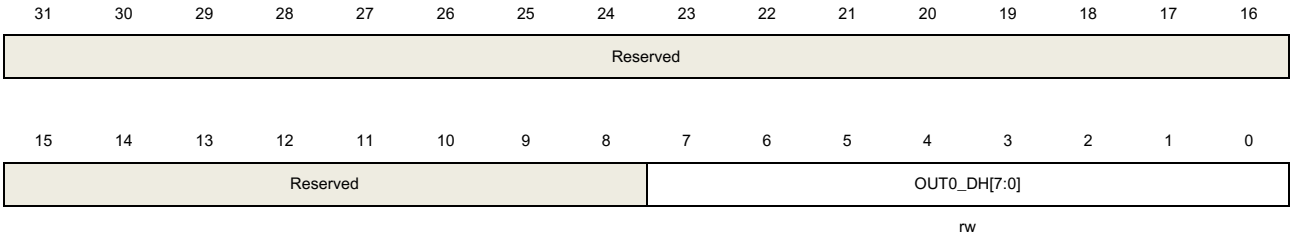
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

19.4.5. 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).



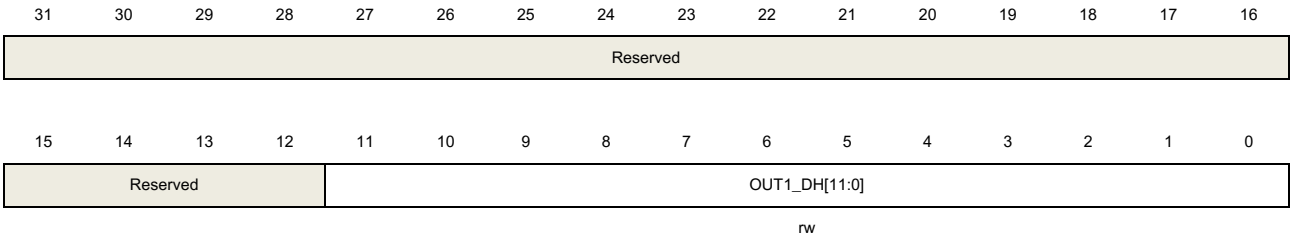
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

19.4.6. DACx_OUT1 12-bit right-aligned data holding register (DAC_OUT1_R12DH)

Address offset: 0x14

Reset value: 0x0000 0000

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



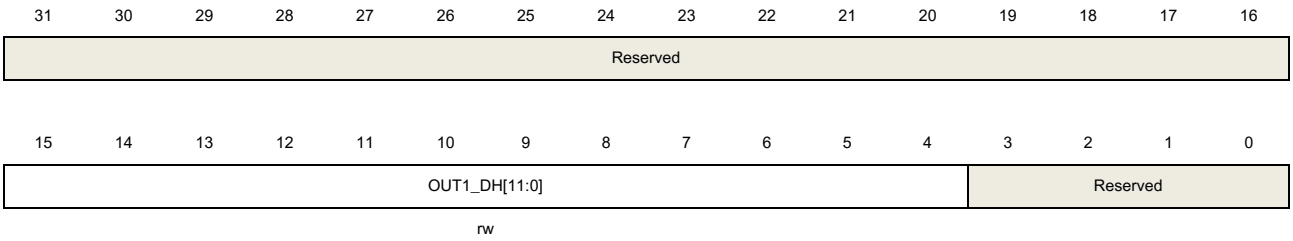
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DH[11:0]	DACx_OUT1 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT1.

19.4.7. DACx_OUT1 12-bit left-aligned data holding register (DAC_OUT1_L12DH)

Address offset: 0x18

Reset value: 0x0000 0000

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



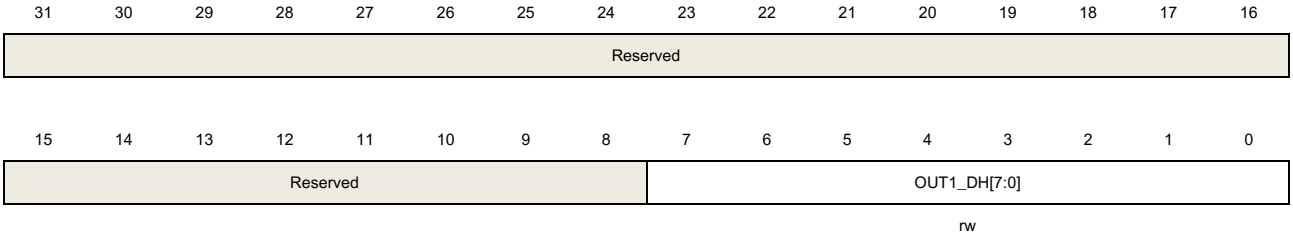
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT1_DH[11:0]	DACx_OUT1 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT1.
3:0	Reserved	Must be kept at reset value.

19.4.8. DACx_OUT1 8-bit right-aligned data holding register (DAC_OUT1_R8DH)

Address offset: 0x1C

Reset value: 0x0000 0000

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



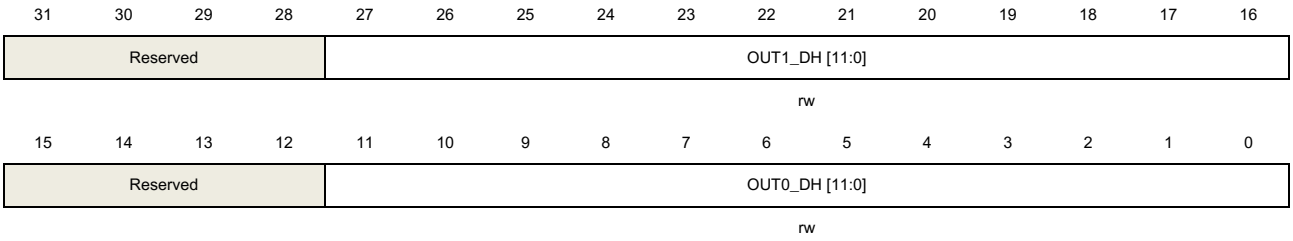
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT1_DH[7:0]	DACx_OUT1 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT1.

19.4.9. DACx concurrent mode 12-bit right-aligned data holding register (DACC_R12DH)

Address offset: 0x20

Reset value: 0x0000 0000

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



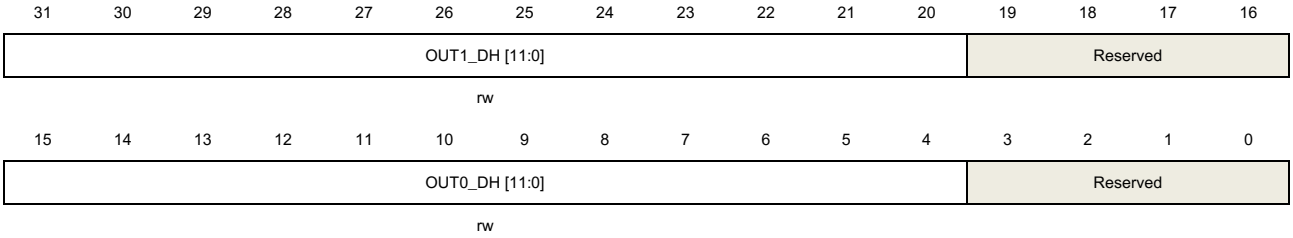
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
27:16	OUT1_DH [11:0]	DACx_OUT1 12-bit right-aligned data These bits specify the data that is to be converted by DACx_OUT1.
12	Reserved	Must be kept at reset value.
11:0	OUT0_DH [11:0]	DACx_OUT0 12-bit right-aligned data These bits specify the data that is to be converted by DACx_OUT0.

19.4.10. DACx concurrent mode 12-bit left-aligned data holding register (DACC_L12DH)

Address offset: 0x24

Reset value: 0x0000 0000

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



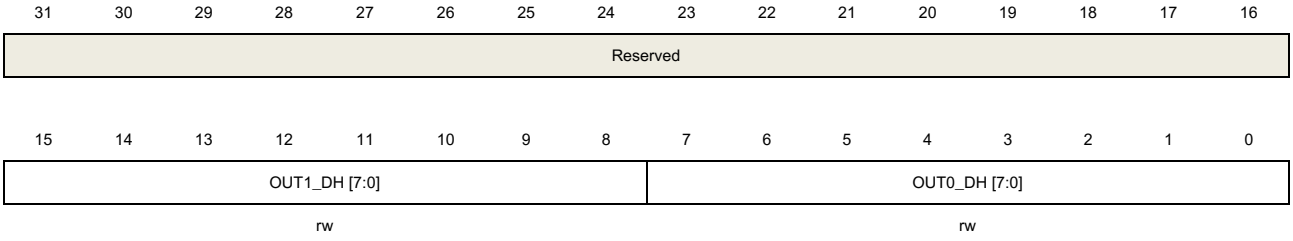
Bits	Fields	Descriptions
31:20	OUT1_DH [11:0]	DACx_OUT1 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT1.
12	Reserved	Must be kept at reset value.
15:4	OUT0_DH [11:0]	DACx_OUT0 12-bit left-aligned data These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

19.4.11. DACx concurrent mode 8-bit right-aligned data holding register (DACC_R8DH)

Address offset: 0x28

Reset value: 0x0000 0000

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



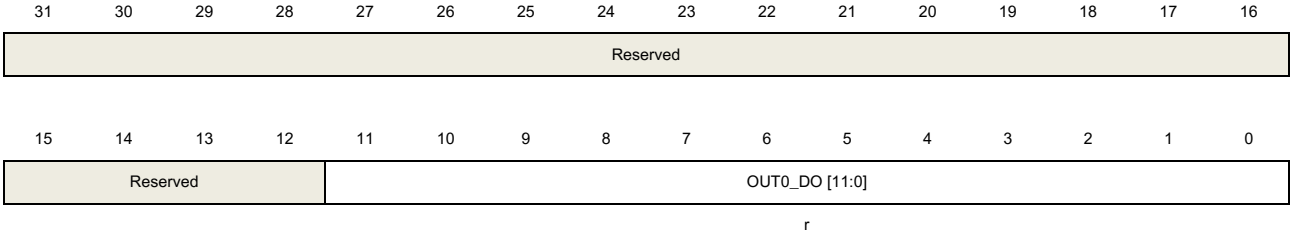
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	OUT1_DH [7:0]	DACx_OUT1 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT1.
7:0	OUT0_DH [7:0]	DACx_OUT0 8-bit right-aligned data These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

19.4.12. 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).



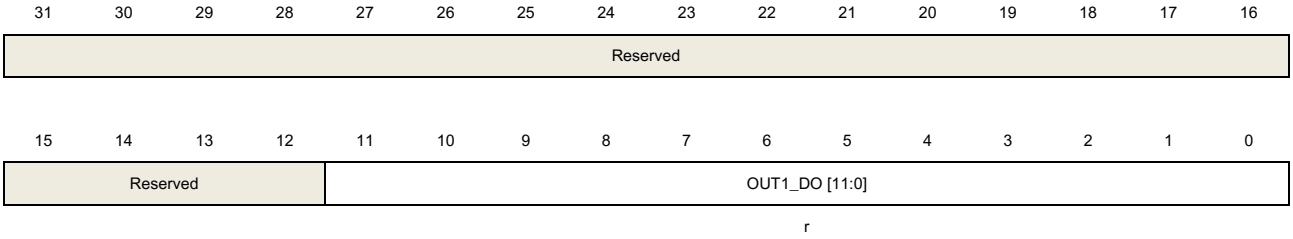
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DO [11:0]	DACx_OUT0 12-bit output data These bits, which are read only, storage the data that is being converted by DACx_OUT0.

19.4.13. DACx_OUT1 data output register (DAC_OUT1_DO)

Address offset: 0x30

Reset value: 0x0000 0000

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



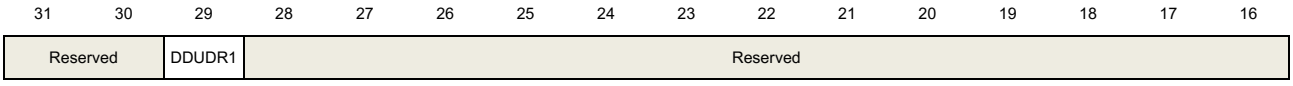
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT1_DO [11:0]	DACx_OUT1 12-bit output data These bits, which are read only, storage the data that is being converted by DACx_OUT1.

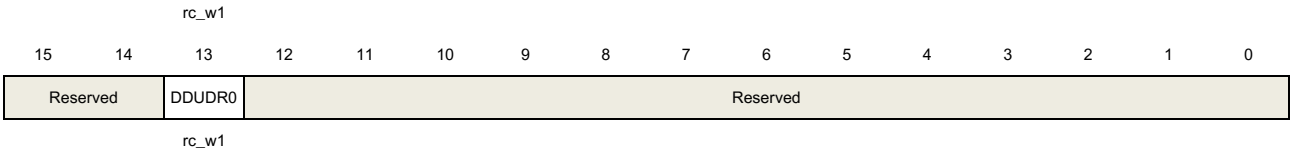
19.4.14. DACx_OUT0 status register 0 (DAC_STAT0)

Address offset: 0x34

Reset value: 0x0000 0000

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





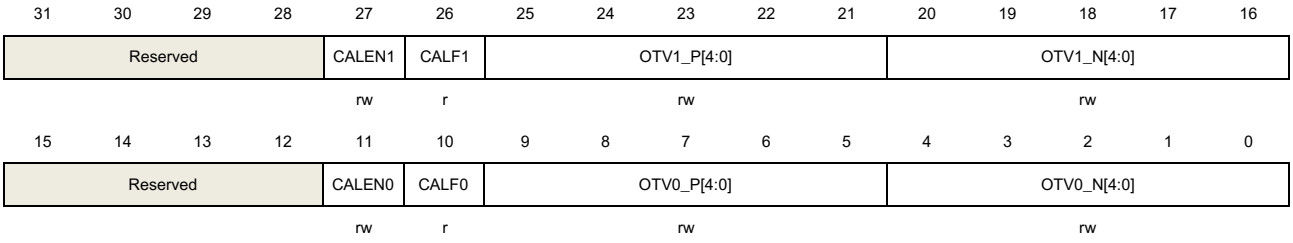
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
29	DDUDR1	DACx_OUT1 DMA underrun flag. This bit is set by hardware and cleared by software (by writing it to 1). 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is higher than the DMA transfer).
28:14	Reserved	Must be kept at reset value.
13	DDUDR0	DACx_OUT0 DMA underrun flag. This bit is set by hardware and cleared by software (by writing it to 1). 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is higher than the DMA transfer).
12:0	Reserved	Must be kept at reset value.

19.4.15. DACx calibration register (DAC_CALR)

Address offset: 0x38

Reset value: 0x0210 0210

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27	CALEN1	DACx_OUT1 calibration enable 0: DACx_OUT1 calibration mode disabled 1: DACx_OUT1 calibration mode enabled **Note:** CALEN1 can be written to 1 only when DEN1 is 0 in the DAC_CTL0 register.
26	CALF1	DACx_OUT1 calibration offset flag. This bit is set and cleared by hardware.

		0: The offset correction value is lower than the calibration value. 1: The offset correction value is higher than or equal to the calibration value.
25:21	OTV1_P[4:0]	DACx_OUT1 offset calibration value P. The default is the factory calibration value. If this register is configured, it will switch to user calibration mode as follows: 00000: offset calibration value 0 00001: offset calibration value 1 … 11111: offset calibration value 31
20:16	OTV1_N[4:0]	DACx_OUT1 offset calibration value N. The default is the factory calibration value. If this register is configured, it will switch to user calibration mode as follows: 00000: offset calibration value 0 00001: offset calibration value 1 … 11111: offset calibration value 31
15:12	Reserved	Must be kept at reset value.
11	CALEN0	DACx_OUT0 calibration enable 0: DACx_OUT0 calibration mode disabled 1: DACx_OUT0 calibration mode enabled **Note:** CALEN0 can be written to 1 only when DEN0 is 0 in the DAC_CTL0 register.
10	CALF0	DACx_OUT0 calibration offset flag. This bit is set and cleared by hardware. 0: The offset correction value is lower than the calibration value. 1: The offset correction value is higher than or equal to the calibration value.
9:5	OTV0_P[4:0]	DACx_OUT0 offset calibration value P. The default is the factory calibration value. If this register is configured, it will switch to user calibration mode as follows: 00000: offset calibration value 0 00001: offset calibration value 1 … 11111: offset calibration value 31
4:0	OTV0_N[4:0]	DACx_OUT0 offset calibration value N. The default is the factory calibration value. If this register is configured, it will switch to user calibration mode as follows: 00000: offset calibration value 0 00001: offset calibration value 1 …

11111: offset calibration value 31

20. Comparator (CMP)

20.1. 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.

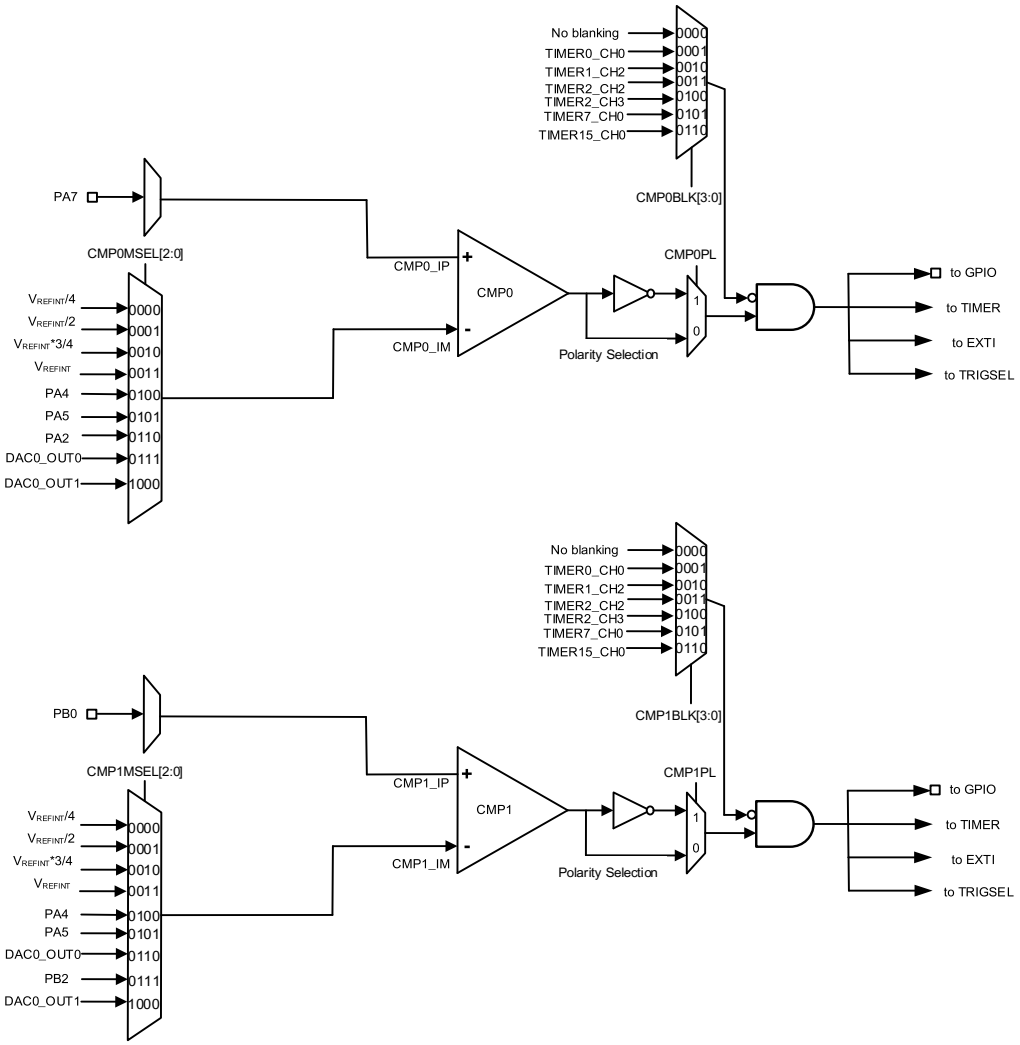
20.2. Characteristic

- 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.

20.3. Function overview

The block diagrams of CMP are shown in ***Figure 20-1. CMP block diagram***.

Figure 20-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

20.3.1. 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.

20.3.2. 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 20-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 20-1. CMP inputs and outputs summary

	CMP0	CMP1
CMP non inverting inputs connected to I/Os	PA7	PB0
CMP inverting inputs connected to I/Os	PA4 PA5 PA2	PA4 PA5 PB2
CMP inverting inputs connected to internal signals	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0 DAC0_OUT1	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC0_OUT0 DAC0_OUT1
CMP outputs connected to I/Os	PA12 PA2 PB9 PE13 PE8	PA10 PB1 PE12 PE9
CMP outputs connected to EXTI	•	•
CMP outputs connected to TRIGSEL	•	•
CMP outputs connected to NVIC	•	•
CMP outputs connected to TIMER break	•	•

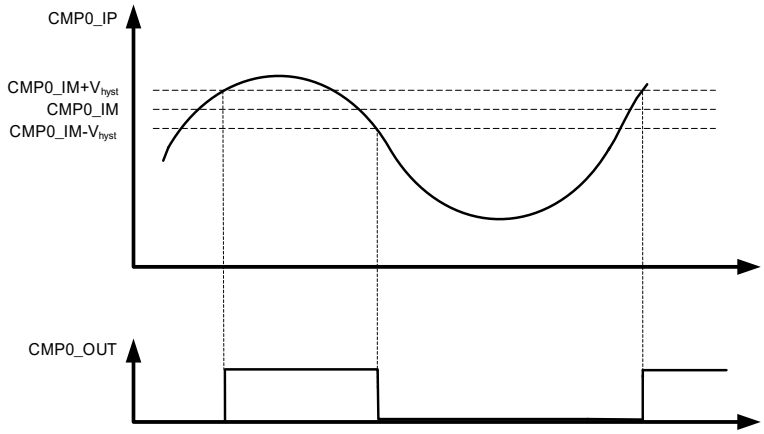
Note:

1. Refer to the TIMER module for details on CMP outputs connected to TIMER break.

20.3.3. 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 CMPx_CS register. This function could be shut down if it is unnecessary.

Figure 20-2. CMP hysteresis



20.3.4. CMP register write protection

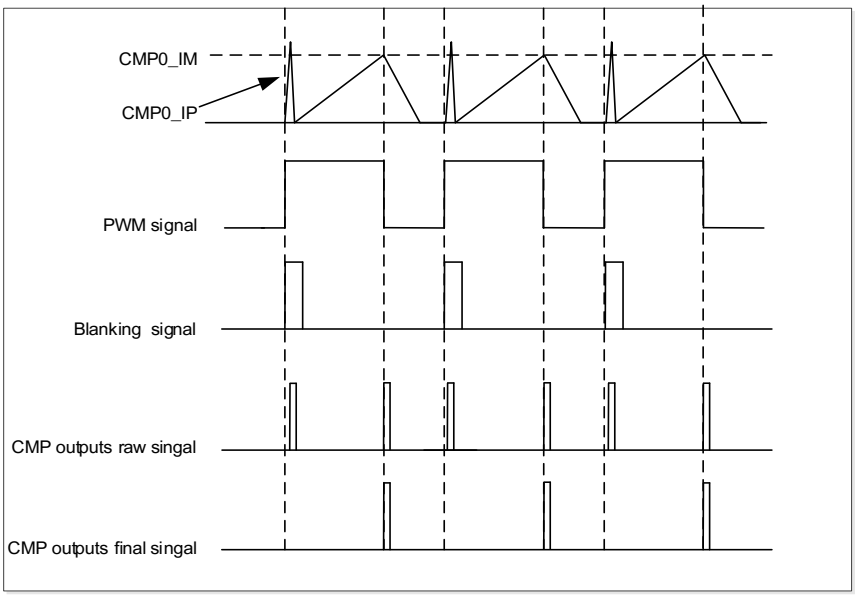
The CMP control and status register (CMPx_CS) can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the MCU reset.

20.3.5. 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 CMPxBLK[3:0] bits in the CMPx_CS register are setting to an available value, the CMP output final signal is obtained by ANDing the complementary signal of the selected blanking signal with the raw output of the comparator. The blanking function can be used for false overcurrent detection in motor control applications.

Figure 20-3. The CMP outputs signal blanking shows the comparator output blank function.

Figure 20-3. The CMP outputs signal blanking



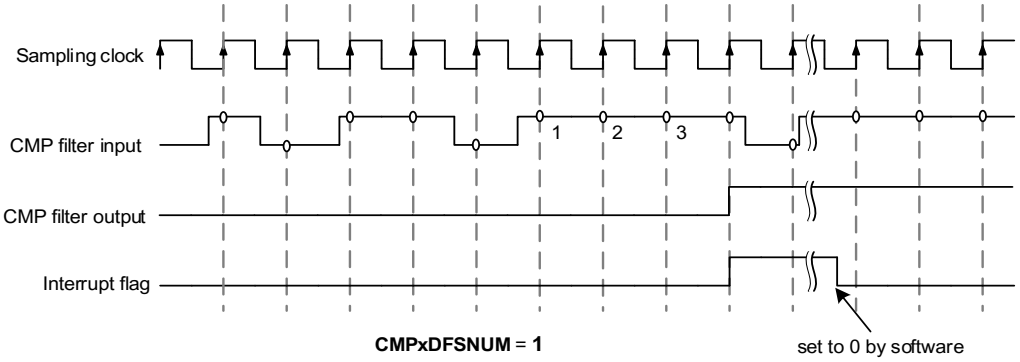
20.3.6. CMP digital filter

The sampling clock of digital filter can be selected by the CMPxDFSCDIV[1:0] bits and the number of matches can be selected by the CMPxDFSNUM bit of CMPx_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, POE source output, and GPT 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 20-4. CMP noise filter and interrupt operation***.

Figure 20-4. CMP noise filter and interrupt operation



Note:

The above operation example is to enable digital noise filtering, and applies when the CMPxDFSCDIV[1:0] bits are 2'b01,2'b10 and 2'b11 and CMPxDFSNUM bits is 1. The noise canceling width can be changed by setting the value of the CMPxDFSCDIV[1:0] bits to select the sampling timing.

20.3.7. 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 CMPxSEN and CMPxBEN bits in CMPx control / status register. The CMPxSEN and CMPxBEN bits are used to enable the VREFINT voltage output and the divider circuit, respectively, to generate the selected voltage.

20.3.8. 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.

20.4. Register definition

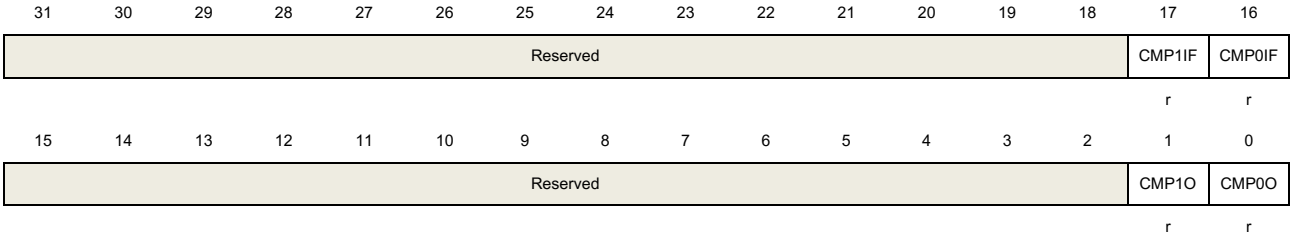
CMP base address: 0x4001 7C00

20.4.1. Status register (CMP_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

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



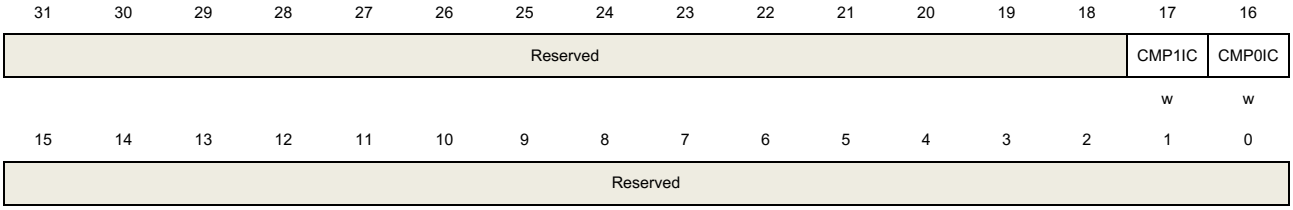
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	CMP1IF	CMP1 interrupt flag 0: No CMP1 output interrupt 1: CMP1 output interrupt Set by hardware when the CMP1 output is set. Cleared by software writing 1 to CMP1IC bit in the CMP_IFC register.
16	CMP0IF	CMP0 interrupt flag 0: No 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.
15:2	Reserved	Must be kept at reset value.
1	CMP1O	CMP1 output state This is a copy of CMP1 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
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

20.4.2. Interrupt flag clear register (CMP_IFC)

Address offset: 0x04

Reset value: 0x0000 0000

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



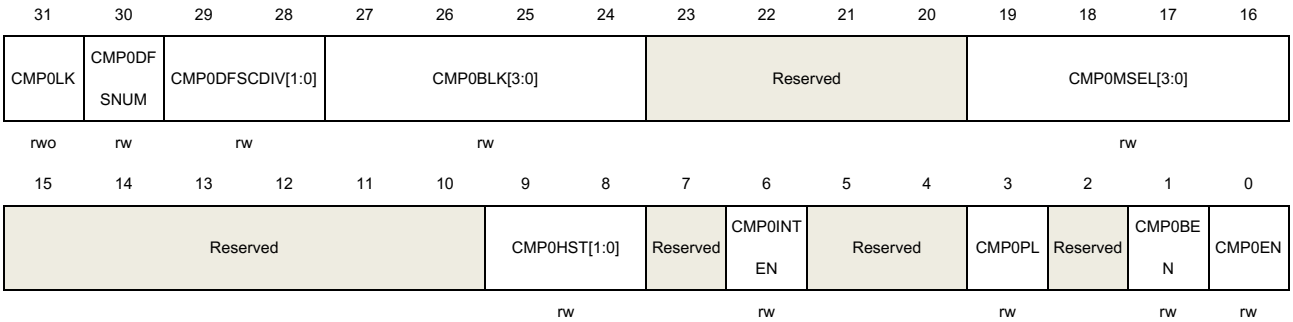
Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	CMP1IC	CMP1 interrupt flag clear 0: Not clear CMP1 interrupt flag 1: Clear CMP1 interrupt flag
16	CMP0IC	CMP0 interrupt flag clear 0: Not clear CMP0 interrupt flag 1: Clear CMP0 interrupt flag
15:0	Reserved	Must be kept at reset value.

20.4.3. CMP0 control/status register (CMP0_CS)

Address offset: 0x08

Reset value: 0x8000 0000

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



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	Noise filter mode 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 noise sampling selection. 00: Noise filter not used 01: Sampling frequency is PCLK / 8. 10: Sampling frequency is PCLK / 16. 11: Sampling frequency is 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 TIMER14_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 1000: DAC0_OUT1 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	Reserved	Must be kept at reset value.
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

20.4.4. CMP1 control/status register (CMP1_CS)

Address offset: 0x10

Reset value: 0x8000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP1LK	CMP1DFSNUM	CMP1DFSCDIV[1:0]		CMP1BLK[3:0]				Reserved				CMP1MSEL[3:0]			
rwo	rw	rw		rw								rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CMP1HST[1:0]		Reserved	CMP1INTEN	Reserved		CMP1PL	Reserved	CMP1BEN	CMP1EN
						rw			rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31	CMP1LK	CMP1 lock This bit allows to have all control bits of CMP1 as read-only. This bit is write once. It can only be cleared by a system reset once it is set by software. 0: CMP1_CS bits are read-write

		1: CMP1_CS bits are read-only
30	CMP1DFSNUM	Noise filter mode 0: Output changes when the same value is sampled 2 times. 1: Output changes when the same value is sampled 3 times
29:28	CMP1DFSCDIV[1:0]	CMP1 digital noise sampling selection. 00: Noise filter not used 01: Sampling frequency is PCLK / 8. 10: Sampling frequency is PCLK / 16. 11: Sampling frequency is PCLK / 32.
27:24	CMP1BLK[3:0]	CMP1 output blanking source This bit is used to select which timer output controls the CMP1 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 TIMER14_CH0 as blanking source All other values: reserved
23:20	Reserved	Must be kept at reset value.
19:16	CMP1MSEL[3:0]	CMP1_IM internal input selection These bits are used to select the source connected to the CMP1_IM input of the CMP1. 0000: V_{REFINT} / 4 0001: V_{REFINT} / 2 0010: V_{REFINT}*3 / 4 0011: V_{REFINT} 0100: PA4 0101: PA5 0110: DAC0_OUT0 0111: PB2 1000: DAC0_OUT1 All other values: reserved
15:10	Reserved	Must be kept at reset value.
9:8	CMP1HST[1:0]	CMP1 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis

		11: High hysteresis
7	Reserved	Must be kept at reset value.
6	CMP1INTEN	CMP1 interrupt enable 0: Disabled 1: Enabled
5:4	Reserved	Must be kept at reset value.
3	CMP1PL	Polarity of CMP1 output This bit is used to select the polarity of CMP1 output. 0: Output is not inverted 1: Output is inverted
2	Reserved	Must be kept at reset value.
1	CMP1BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge disable in case that CMP1BEN bit of CMP1_CS is also reset 1: Enable scaler resistor bridge
0	CMP1EN	CMP1 enable 0: CMP1 disabled 1: CMP1 enabled

21. 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.

21.1. Free watchdog timer (FWDGT)

21.1.1. 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 or the counter is refreshed when the value of the counter is greater than the window register value. The register write protection function in free watchdog can be enabled to prevent it from changing the configuration unexpectedly.

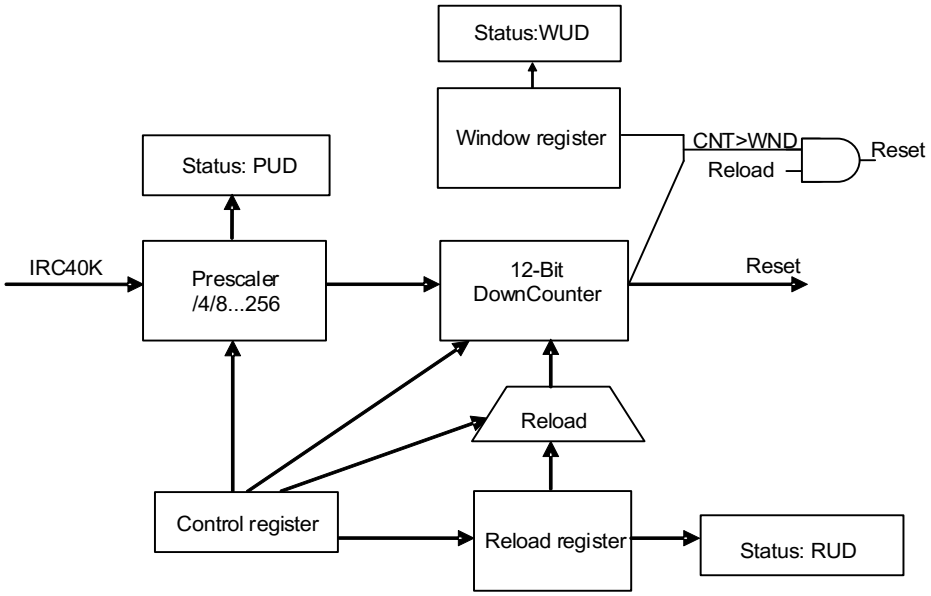
21.1.2. Characteristics

- Free-running 12-bit down counter.
- Generate reset in two conditions when FWDGT is enabled:
  - Reset when the counter reached 0.
  - The counter is refreshed when the value of the counter is greater than the window register value.
- Free clock source, FWDGT can operate even if the main clock fails such as in standby, Deep-sleep1 and Deep-sleep modes.
- Hardware free watchdog bit, automatically start the FWDGT or not when power on.
- FWDGT debug mode, the FWDGT can stop or continue to work in debug mode.

21.1.3. Function overview

The free watchdog consists of an 8-stage prescaler and a 12-bit down counter. ***Figure 21-1. Free watchdog block diagram*** shows the functional block of the free watchdog module.

Figure 21-1. Free watchdog block diagram



The free watchdog is enabled by writing the value (0xCCCC) to the control register (FWDGT_CTL), then counter starts counting down. When the counter reaches the value (0x000), there will be a reset. The counter can be reloaded by writing the value (0xAAAA) to the FWDGT_CTL register at any time. The reload value comes from the FWDGT_RLD register. The software can prevent the watchdog reset by reloading the counter before the counter reaches the value (0x000).

By setting the appropriate window in the FWDGT_WND register, the FWDGT can also work as a window watchdog timer. A reset will occur if the reload operation is performed while the counter is greater than the value stored in the window register (FWDGT_WND). The default value of the FWDGT_WND is 0x0000 0FFF, so if it is not updated, the window option is disabled. A reload operation is performed in order to reset the downcounter to the FWDGT_RLD value and the prescaler counter to generate the next reload, as soon as the window value is changed.

The free watchdog can automatically start at power on when the hardware free watchdog nWDG_HW bit in the device option bytes 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), status register (FWDGT_STAT) 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 21-1. Min/max FWDGT timeout period at 40 kHz (IRC40K)

Prescaler divider	PSC[3:0] bits	Min timeout (ms) RL[11:0]=0x000	Max timeout (ms) RL[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
1 / 256	0110	6.4	26214.4
1 / 512	0111	12.8	52428.8
1 / 1024	1000	25.6	104857.6
1 / 2048	1001	51.2	209715.2
1 / 4096	1010	102.4	419430.4
1 / 8192	1011	204.8	838860.8
1 / 16384	1100	409.6	1677721.6
1 / 32768	1101	819.2	3355443.2
1 / 65536	1110	1638.4	6710886.4
1 / 131072	1111	3276.8	13421772.8

The FWDGT timeout can be more accurately by calibrating the IRC40K.

Note:

1. When after the execution of watchdog reload operation, if the MCU needs enter the deep-sleep / deep-sleep 1 / standby mode immediately, more than 3 IRC40K clock intervals must be inserted in the middle of reload and deep-sleep / deep-sleep 1 / standby mode commands by software setting.

2. Seven or more IRC40K time intervals should be inserted between two consecutive dog feedings.

21.1.4. Register definition

FWDGT base address: 0x4000 3000

Control register (FWDGT_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different functions are realized by writing these bits with different values. 0x5555: Disable the FWDGT_PSC, FWDGT_RLD and FWDGT_WND write protection. 0xCCCC: Start the free watchdog timer counter. When the counter reduces to 0, the free watchdog generates a reset. 0xAAAA: Reload the counter.

Prescaler register (FWDGT_PSC)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	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[3:0]			
												rw			

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

0101: 1/128

0110: 1/256

……

1100: 1/16384

1101: 1/32768

1110: 1/65536

1111: 1/131072

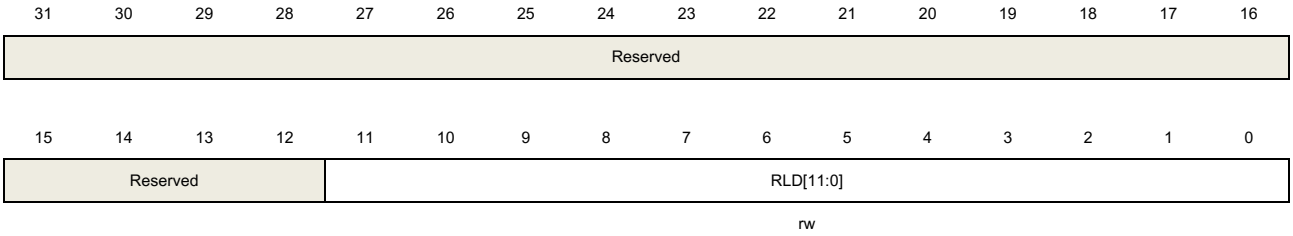
If several prescaler values are used by the application, it is mandatory to wait until PUD bit has been reset before changing the prescaler value. If the prescaler value has been updated, it is not necessary to wait until PUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until PUD is reset).

Reload register (FWDGT_RLD)

Address offset: 0x08

Reset value: 0x0000 0FFF

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



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT conter with the RLD value. These bits are write-protected. Write 0X5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit has been reset before changing the reload value. If the reload value has been updated, it is not necessary to wait until RUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until RUD is

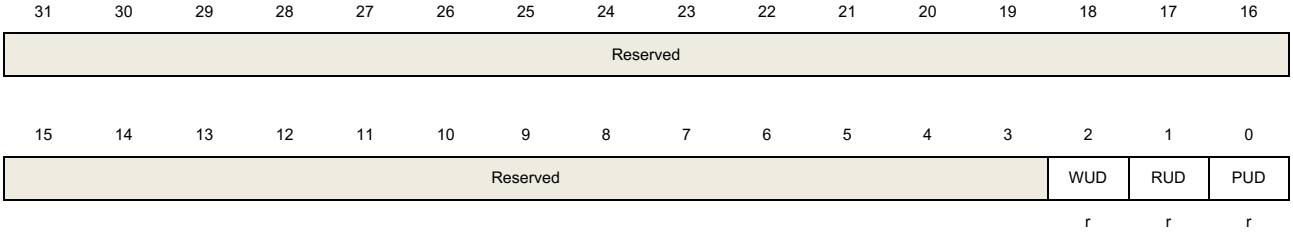
reset).

Status register (FWDGT_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

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



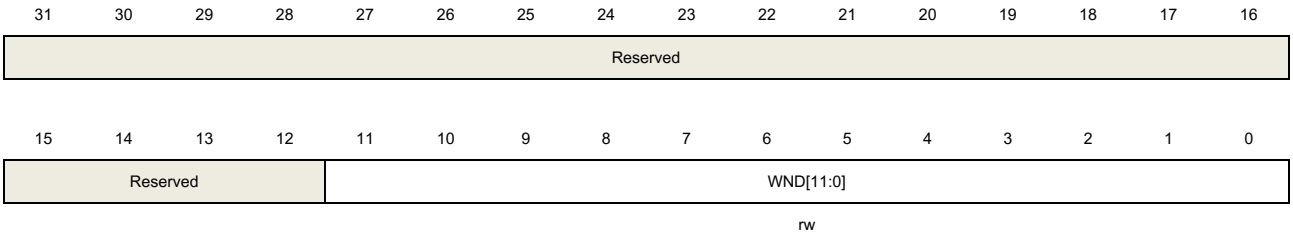
Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	WUD	Watchdog counter window value update When a write operation to FWDGT_WND register ongoing, this bit is set and the value read from FWDGT_WND register is invalid.
1	RUD	Free watchdog timer counter reload value update During a write operation to FWDGT_RLD register, this bit is set and the value read from FWDGT_RLD register is invalid.
0	PUD	Free watchdog timer prescaler value update During a write operation to FWDGT_PSC register, this bit is set and the value read from FWDGT_PSC register is invalid.

Window register (FWDGT_WND)

Address offset: 0x10

Reset value: 0x0000 0FFF

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



Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WND[11:0]	Watchdog counter window value. These bits are used to contain the high limit of the

window value to be compared to the downcounter. A reset will occur if the reload operation is performed while the counter is greater than the value stored in this register. The WUD bit in the FWDGT_STAT register must be reset in order to be able to change the reload value.

These bits are write protected. Write 0x5555 in the FWDGT_CTL register before writing these bits.

If several window values are used by the application, it is mandatory to wait until WUD bit has been reset before changing the window value. However, after updating the window value it is not necessary to wait until WUD is reset before continuing code execution except in case of low-power mode entry (Before entering low-power mode, it is necessary to wait until WUD is reset).

21.2. Window watchdog timer (WWDGT)

21.2.1. 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. Interrup occurs if it is enable.

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.

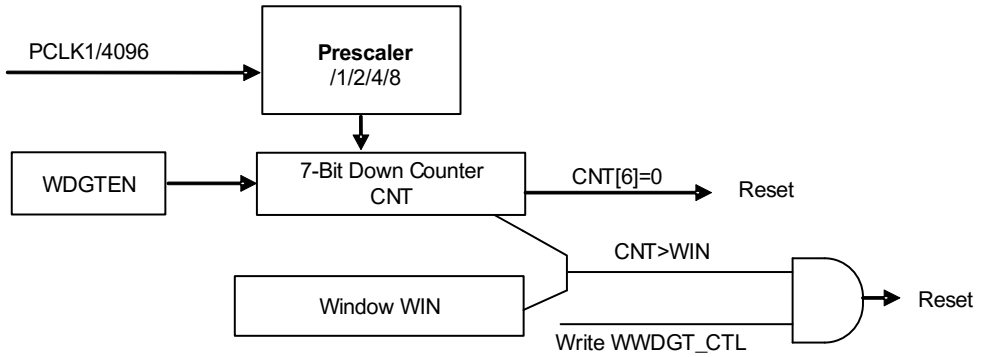
21.2.2. Characteristics

- Programmable free-running 7-bit down counter.
- Generate reset in two conditions when WWDGT is enabled:
  - Reset when the counter reached 0x3F.
  - The counter is refreshed when the value of the counter is greater than the window register value.
- Early wakeup interrupt (EWI): the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x40 or refreshes before it reaches the window value.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

21.2.3. Function overview

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

Figure 21-2. Window watchdog timer block diagram



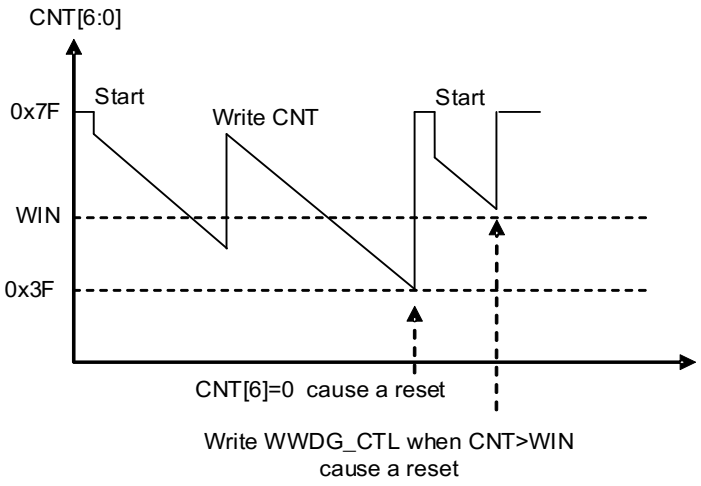
The window watchdog timer is always disabled after power on reset. The software starts the watchdog by setting the WDGTEN bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F (it implies 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[4:0] 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 or the counter is refreshed before it reaches the window value. The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyse the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things.

The EWI interrupt flag is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 21-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (21\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB1 clock period measured in ms

The table below shows the minimum and maximum values of the t_{WWDGT} for PCLK1 at 100MHZ.

Table 21-2. Min-max timeout value at 100 MHz (f_{PCLK1})

Prescaler divider	PSC[4:0]	Min timeout value(ms) CNT[5:0] =0x00	Max timeout value(ms) CNT[5:0]=0x3F
1/1	00000	0.041	2.621
1/2	00001	0.082	5.243
1/4	00010	0.164	10.486
1/8	00011	0.328	20.972
1/16	00100	0.655	41.943
1/32	00101	1.311	83.886
1/64	00110	2.621	167.772
1/128	00111	5.243	335.544
1/256	01000	10.486	671.089
1/512	01001	20.972	1342.177
1/1024	01010	41.943	2684.355
1/2048	01011	83.886	5368.709
1/4096	01100	167.772	10737.418
1/8192	01101	335.544	21474.836
1/16384	01110	671.089	42949.673
1/32768	01111	1342.177	85899.346
1/65536	10000	2684.355	171798.692
1/131072	10001	5368.709	343597.384
1/262144	>=10010	10737.418	687194.767

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.

21.2.4. Register definition

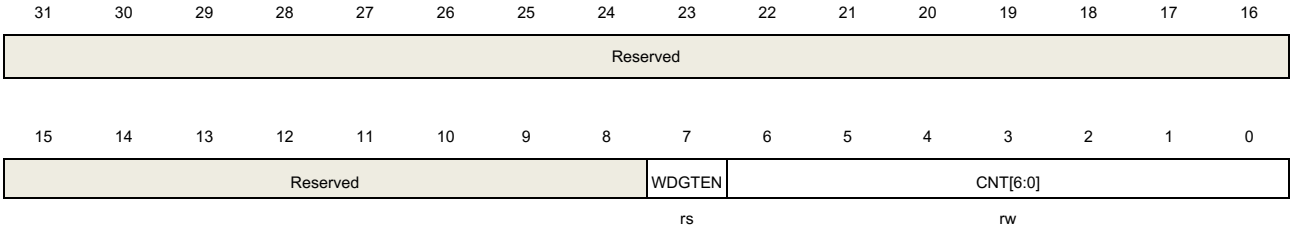
WWDGT base address: 0x4000 2C00

Control register (WWDGT_CTL)

Address offset: 0x00

Reset value: 0x0000 007F

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



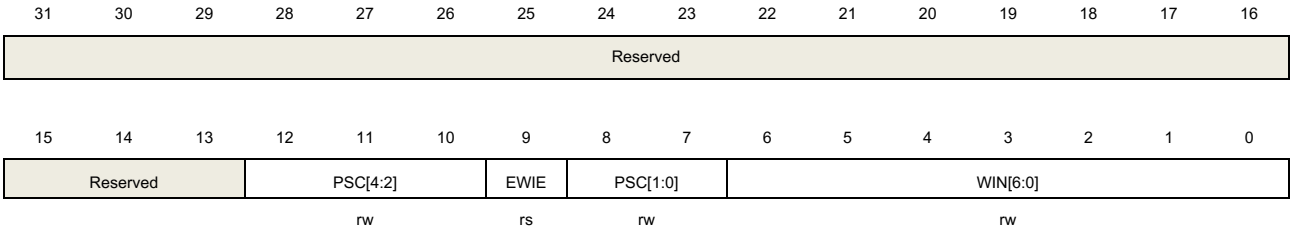
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	WDGTEN	Start the Window watchdog timer. It can be cleared by a hardware reset or software clock reset (refer to ***APB1 enable register (RCU_APB1EN)***). Writing 0 has no effect. 0: Window watchdog timer disabled 1: Window watchdog timer enabled
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset occur when the value of this counter decreases from 0x40 to 0x3F. When the value of this counter is greater than the window value, writing this counter also causes a reset.

Configuration register (WWDGT_CFG)

Address offset: 0x04

Reset value: 0x0000 007F

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



Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:10	PSC[4:2]	Prescaler. The time base of the watchdog counter

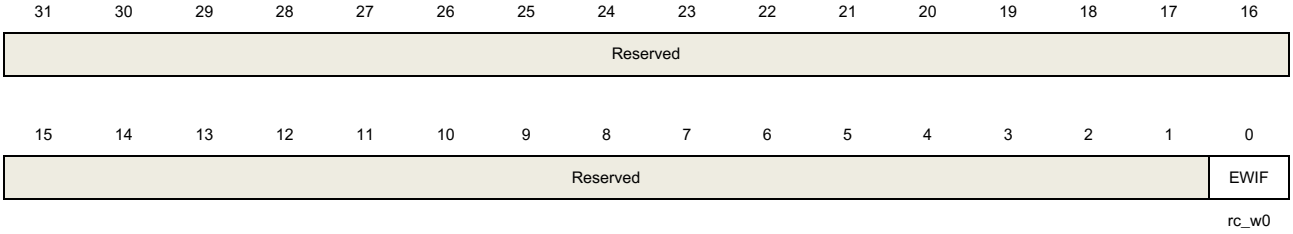
00000: (PCLK1 / 4096) / 1
00001: (PCLK1 / 4096) / 2
00010: (PCLK1 / 4096) / 4
00011: (PCLK1 / 4096) / 8
……
10010: (PCLK1 / 4096) / 262144
>=10010: (PCLK1 / 4096) / 262144

9	EWIE	Early wakeup interrupt enable. If the bit is set, an interrupt occurs when the counter reaches 0x40 or refreshes before it reaches the window value. It can be cleared by a hardware reset or software reset by setting the WWDGTRST bit of the RCU module. A write operation of 0 has no effect.
8:7	PSC[1:0]	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 occur if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is greater than the Window value.

Status register (WWDGT_STAT)

Address offset: 0x08
Reset value: 0x0000 0000

This register can be accessed by 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. There is no effect when writing 1.

22. Real time clock (RTC)

22.1. Overview

The RTC provides time that includes hours / minutes / seconds / sub-seconds and a calendar that includes year / month / day / weekday. The time and calendar are expressed in BCD code except for sub-seconds. Sub-seconds are expressed in binary code. Hour adjustment for daylight saving time. Working in power saving mode and smart wakeup are software configurable. Support improving calendar accuracy using external accurate low frequency clock.

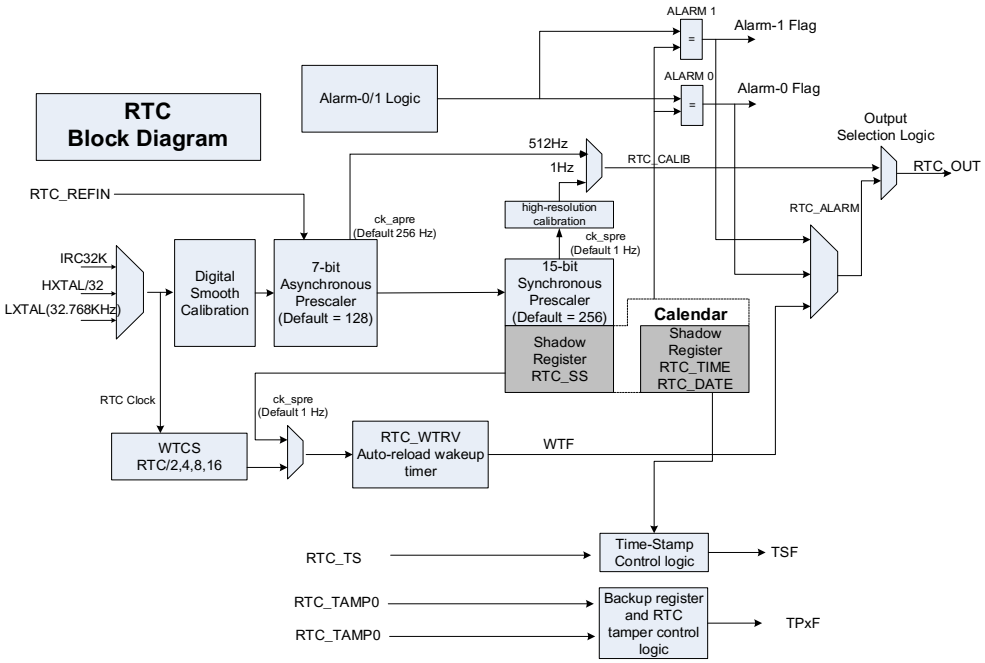
22.2. Characteristics

- Support calendar function, which can support year, month, date(day of month), day(day of week), hours, minutes, seconds and subseconds.
- Daylight saving compensation supported, which is realized through software.
- External high-accuracy low frequency (50Hz or 60Hz) clock can be used to achieve higher calendar accuracy through the reference clock detection option function.
- Atomic clock adjustment for calendar calibration performed by digital calibration function. Maximum adjustment accuracy is 0.95PPM for smooth digital calibration, 0.0625PPM for high-resolution calibration.
- Sub-second adjustment by shift function.
- Two separate Time-stamp function for saving event time.
- Two tamper sources can be chosen and tamper type is configurable (RTC_TAMP0 and RTC_TAMP1).
- Programmable calendar and two field maskable alarms.
- Maskable interrupt source:
  - Alarm 0 and Alarm 1.
  - Time-stamp detection.
  - Tamper detection.
  - Auto wakeup event.
- Four 32-bit (128 bytes total) universal backup registers which can keep data under power saving mode. Backup register will be reset if tamper event detected.

22.3. Function overview

22.3.1. Block diagram

Figure 22-1. Block diagram of RTC



The RTC unit includes:

- Two alarm event / interrupt and support two tamper event / interrupt from I/Os.
- When tamper detection event happen, there will generate a time-stamp event.
- When tamper detection event happen, the backup registers will be erased.
- When power switch is switched to the V_{BAT}, there will generate a time-stamp event.
- Four 32-bit backup registers.
- Optional RTC output function:
  - 512Hz (default prescaler): RTC_OUT(PC13 or PB2).
  - 1Hz(default prescaler): RTC_OUT(PC13 or PB2).
  - Alarm event(polarity is configurable): RTC_OUT(PC13 or PB2).
  - Automatic wakeup event(polarity is configurable): RTC_OUT(PC13 or PB2).
- Optional RTC input function:
  - Time-stamp 0 event detection: RTC_TS0(PC13).
  - Time-stamp 1 event detection: RTC_TS1(PD4).
  - Tamper 0 event detection: RTC_TAMP0(PC13).
  - Tamper 1 event detection: RTC_TAMP1(PD4).
  - Reference clock input: RTC_REFIN(PB15 or PB13).

22.3.2. Clock source and prescalers

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

128.

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

The frequency formula of two prescalers is shown as below:

$$f_{ck_apre} = \frac{f_{rtcclk}}{FACTOR_A + 1} \quad (22\text{-}1)$$

$$f_{ck_spre} = \frac{f_{ck_apre}}{FACTOR_S + 1} = \frac{f_{rtcclk}}{(FACTOR_A + 1)*(FACTOR_S + 1)} \quad (22\text{-}2)$$

The ck_apre clock is used to driven the RTC_SS down counter which stands for the time left to next second in binary format and when it reaches 0 it will automatically reload FACTOR_S[14:0] value. The ck_spre clock is used to driven the calendar registers. Each clock will make second plus one.

Note: When the RTC uses the divided clock of HXTAL as the working clock source, a system reset will cause the HXTAL clock enable to reset, resulting in the RTC being unable to function properly. The clock needs to be reconfigured for the RTC to work normally.

22.3.3. Shadow registers introduction

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

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

System reset will reset the shadow calendar registers.

22.3.4. Configurable and field maskable alarm

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

RTC alarm function can be enabled or disabled by ALRMxEN(x=0,1) bit in RTC_CTL. If all the alarm fields value match the corresponding calendar value when ALRMxEN=1(x=0,1), the Alarm flag will be set.

Note: FACTOR_S[14:0] in the RTC_PSC register must be larger than 3 if MSKS bit reset in

RTC_ALRMxTD(x=0,1).

If a field is masked, the field is considered as matched in logic. If all the fields have been masked, the Alarm Flag will assert 3 RTC clock later after ALRMxEN(x=0,1) is set.

22.3.5. RTC initialization and configuration

RTC register write protection

The BKPWEN bit in the PMU_CTL0 register is cleared by default, so writing to RTC registers needs setting BKPWEN bit ahead of time.

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

Following below steps will unlock the write protection:

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

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

RTC_TIME, RTC_DATE, RTC_CTL, RTC_STAT, RTC_PSC, RTC_WUT, RTC_ALRM0TD, RTC_ALRM1TD, RTC_SHIFTCTL, RTC_HRFC, RTC_ALRM0SS, RTC_ALRM1SS.

Calendar initialization and configuration

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

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

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

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

Note: AM/PM(PM bit in RTC_TIME register) can be configured under 24-hour mode, but after exiting the initialization mode, real calendar registers will not load the value of AM/PM. The value of AM/PM will be reset two RTC clock cycles after exiting the initialization mode and the RSYNF bit in RTC_STAT register is set.

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

Daylight saving Time

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

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

Note: S1H operation can not be excuted at the time of 0:00~1:00 in 24-hour format nor 12:00~1:00 AM(PM=0) in 12-hour format. The operation excuted at this time will not take effect.

Alarm function operation process

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

1. Disable Alarm (by resetting ALRMxEN(x=0,1) in RTC_CTL)
2. Set the Alarm registers needed(RTC_ALRMxTD/RTC_ALRMxSS(x=0,1))
3. Enable Alarm function (by setting ALRMxEN(x=0,1) in the RTC_CTL)

22.3.6. Configurable periodic auto-wakeup counter

In the RTC block, there is a 16-bit down counter designed to generate periodic wakeup flag. This function is enabled by set the WTEN to 1 and can be running in power saving mode. Two clock sources can be chose for the down counter:

1) RTC clock divided by 2 / 4 / 8 / 16.
   Assume RTC clock comes from LXTAL (32.768 KHz), this can periodically assert wakeup interrupt from 122us to 32s under the resolution down to 61us.

2) Internal clock ck_spre
   Assume ck_spre is 1Hz, this can periodically assert wakeup interrupt from 1s to 36 hours under the resolution down to 1s.

   - WTCS[2:1] = 0b10.This will make period to be 1s to 18 hours.
   - WTCS[2:1] = 0b11.This will make period to be 18 to 36 hours.

When this function is enabled, the down counter is running. When it reaches 0, the WTF flag is set and the wakeup counter is automatically reloaded with RTC_WUT value.

When WTF asserts, software must then clear it.

If WTIE is set and this counter reaches 0, a wakeup interrupt will make system exit from the power saving mode. System reset has no influence on this function.

WTF is also can be output to RTC_OUT from RTC_ALARM channel.

22.3.7. Calendar reading

Reading calendar registers under BPSHAD=0

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

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

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

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

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

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

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

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

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

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

Reading calendar registers under BPSHAD=1

When BPSHAD=1, RSYNF is cleared and maintains as 0 by hardware so reading calendar registers does not care about RSYNF bit. Current calendar value is read from real-time calendar counter directly. The benefit of this configuration is that software can get the real

current time without any delay after wakeup from power saving mode (Deep-sleep /Standby Mode).

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

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

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

22.3.8. Resetting the RTC

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

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

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

- RTC current real-time calendar registers.
- RTC Control register (RTC_CTL).
- RTC Prescaler register (RTC_PSC).
- RTC Wakeup timer register (RTC_WUT).
- RTC High resolution frequency compensation register (RTC_HRFC).
- RTC Shift control register (RTC_SHIFTCTL).
- RTC Time-stamp registers (RTC_SSTSxF / RTC_SSTSxR / RTC_TTSxF / RTC_TTSxR / RTC_DTSxF / RTC_DTSxR).
- RTC Tamper register (RTC_TAMP).
- RTC Backup registers (RTC_BKPx).
- RTC Configuration registers (RTC_CFG).
- RTC Alarm registers (RTC_ALRMxSS / RTC_ALRMxTD(x=0,1)).

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

22.3.9. RTC shift function

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

RTC_SS register indicates the fraction of a second in binary format and is down counting when RTC is running. Therefore by adding the SFS[14:0] value to the synchronous prescaler

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

The maximal RTC_SS value depends on the FACTOR_S[14:0] value in RTC_PSC. The higher FACTOR_S[14:0], the higher adjust precision.

Because of the 1Hz clock (ck_spre) is generated by FACTOR_A[6:0] and FACTOR_S[14:0], the higher FACTOR_S[14:0] means the lower FACTOR_A[6:0], then more power consuming.

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

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

Shift operation only works correctly when REFEN=0.

Software must not write to RTC_SHIFTCTL if REFEN=1.

22.3.10. RTC reference clock detection

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

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

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

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

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

When reference detection function is running while the external reference clock is removed (no reference clock edge found in 3 ck_apre window), the calendar updating still can be performed by LXTAL clock only. If the reference clock is recovered later, detection function will use 7 ck_apre window to identify the reference clock and use 3 ck_apre window to adjust the 1Hz clock (ck_spre) edge.

Note: Software must configure the FACTOR_A=0x7F and FACTOR_S[14:0]=0xFF before enabling reference detection function (REFEN=1).

Reference detection function does not work in Standby Mode.

22.3.11. RTC smooth digital calibration

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

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

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

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

So using CMSK[8:0] can mask clock cycles from 0 to 511 and thus the RTC frequency can be reduced by up to 487.1PPM.

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

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

The combined using of CMSK[8:0] and FREQI can adjust the RTC cycles from -511 to +512 cycles in the period time which means the calibration range is -487.1PPM to +488.5PPM with a resolution of about 0.954PPM.

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

$$f_{cal}=f_{rtcclk} \times (1+\frac{FREQI\times 512-CMSK}{2^{N}+CMSK-FREQI\times 512}) \quad (22\text{-}3)$$

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

Calibration when FACTOR_A < 3

When asynchronous prescaler value (FACTOR_A[6:0]) is set to less than 3, software should not set FREQI bit to 1 when using calibration function. FREQI setting will be ignored when FACTOR_A<3.

When the FACTOR_A[6:0] is less than 3, the FACTOR_S[14:0] value should be set to a value less than the nominal value. Assuming that RTC clock frequency is nominal 32.768 KHz, the corresponding FACTOR_S[14:0] should be set as following rule:

FACTOR_A[6:0] = 2: 2 less than nominal FACTOR_S[14:0] (8189 with 32.768 KHz).
FACTOR_A[6:0] = 1: 4 less than nominal FACTOR_S[14:0] (16379 with 32.768 KHz).
FACTOR_A[6:0] = 0: 8 less than nominal FACTOR_S[14:0] (32759 with 32.768 KHz).

When the FACTOR_A[6:0] is less than 3, CMSK[8:0] is 0x100, the formula of calibration frequency is as follows:

$$f_{cal}=f_{rtcclk} \times (1+\frac{256-CMSK}{2^N+CMSK-256}) \quad (22\text{-}4)$$

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

Verifying the RTC calibration

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

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

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

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

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

  In this configuration, CMSK[8:0] is fixed to 0 by hardware. Using exactly 16s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 0.954PPM (0.5 RTCCLK cycles over 16s).

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

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

Re-calibration on-the-fly

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

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

Multiple calibaration window

Software can select single window or cwindows mode of calibration. If MWE in RTC_HRFC is not set, only one calibration window will be active when software write RTC_HRFC. Otherwise, calibration will keep running according to the configration in RTC_HRFC. When the current calibration window finishes its work, another calibration window will start

immediately.

Software write 0 to MWE will close the multiple windows mode. Current calibration window on the run will not be affect by the closing action. It is not allowed to change the configuration of the smoothing calibration window and calibration value at the same time when writing 0 to MWE.

22.3.12. RTC high-resolution calibration

Using a high-frequency clock signal, the RTC divided clock can be calibrated with high resolution. The calibration accuracy depends on the frequency of the high-frequency clock in the range of 8MHz to 31MHz. It is recommended to use 16MHz, a calibration step of 0.0625 PPM (calibration accuracy of 0.03125 PPM) can be achieved.

The High-Resolution Clock Frequency Calibration Register (RTC_HRCFC) and the High-Resolution Calibration Register (RTC_HRCR) jointly determine the number of RTCCLK clock cycles and high-frequency clock cycles calibrated within each ck_spre cycle.

By configuring HRC_MASK[14:0], the HRC_MASK[14:0] RTCCLK cycles (up to FACTOR_S[14:0]) can be added within each ck_spre cycle. By configuring HRCP bit in RTC_HRCR register, FACTOR_S+1 RTCCLK cycles can be reduced within each ck_spre cycle. By configuring HRCN[9:0] can be added or reduced within each ck_spre cycle, depending on the value of HRCADD. High-resolution calibration takes effect after two RTCCLK cycles following the configuration of the RTC_HRCR register, and its status can be checked via the HRCPF bit in RTC_STAT register.

High-resolution calibration is achieved by using the high-frequency clock to simulate the last RTCCLK of each ck_spre cycle, requiring the circuit to know the ratio between the adopted high-frequency clock frequency and the RTCCLK frequency. Considering that the high-frequency clock frequency can be derived from system clock division, the frequency is not fixed, and the system clock may have certain errors, two methods are provided to calculate the ratio between the high-frequency clock frequency and the RTCCLK frequency:

Hardware Calculation:

The user directly writes the high-frequency clock frequency value into the circuit via the HRCF[14:0] bit in RTC_HRCFC register (unit: KHz). The hardware circuit calculates the frequency ratio using the formula: HRCF[14:0]/32768.

Software Calculation:

Using Timer or external devices to precisely measure the high-frequency clock and RTCCLK frequency ratio, the ratio value is written into the HRCR[9:0] bit in RTC_HRCFC register, and HRCRV bit in RTC_HRCFC is enabled. When high-resolution compensation is activated, the circuit loads the clock frequency ratio from the register.

Note: Before enabling high-resolution calibration, the configuration of HRCF[14:0] / HRCR[9:0] / HRCRV must be completed.

The calculation formula for the register values required to enable high-resolution calibration is as follows:

$$PREDIV=(FACTOR_A+1)\times(FACTOR_S+1) \quad (22\text{-}5)$$

$$RTCCAL=HRC_MASK-(FACTOR_S+1)\times HRCP \quad (22\text{-}6)$$

$$HRCADD\grave{}=\begin{cases}-1, & HRCADD=0\\ 1, & HRCADD=1\end{cases} \quad (22\text{-}7)$$

$$HIGHCAL=\frac{HRCADD\grave{}\times HRCN}{HRCR\grave{}} \quad (22\text{-}8)$$

$$f_{hcal}=f_{ck_spre}\times\frac{PREDIV}{PREDIV+RTCCAL+HIGHCAL} \quad (22\text{-}9)$$

Formula Explanation:

f_{hcal}: ck_spre frequency after high-resolution calibration

f_{ck_spre}: ck_spre frequency before calibration

HRCR`: The final effective frequency ratio between the high-frequency clock and RTCCLK in the circuit

Note:The actual configuration value of the HRC_MASK[14:0] bit in RTC_HRCR register should not exceed the synchronization division coefficient FACTOR_S. The HRCN[9:0] bit in RTC_HRCR register configuration value should be less than HRCR`. If the configuration value exceeds HRCR`, the effective HRCN[9:0] value will equal HRCR`-1, but the HRCN[9:0] bit in RTC_HRCR register will still read the originally configured value.

High-Resolution Calibration Verification

The ck_spre after high-resolution calibration can be output through the PC13 pin for verification. The verification method is as follows:

Enable high-resolution compensation and check the HRCPF bit in RTC status register RTC_STAT. Once the flag is set, measure the period length and frequency of ck_spre starting from the first rising edge of ck_spre.

Since the high-frequency clock involved in high-resolution calibration may have errors, it is strongly recommended to calculate the frequency ratio between the high-frequency calibration clock and the RTCCLK clock through software.

Dynamic Modification of High-Resolution Calibration

High-resolution calibration supports dynamic modification during calibration. When the chip's operating environment changes, such as operating temperature or fluctuations in external crystal oscillators, the software can modify the calibration registers without disabling high-resolution calibration. During modification, the configuration of the RTC_HRCFC register must be completed before RTC_HRCR. If the current dynamic modification only involves the RTC_HRCFC register, the RTC_HRCR register also needs to be reconfigured, and the configuration value must be the same as the previous RTC_HRCR register configuration. Dynamic modifications also require two RTCCLK cycles to take effect.

Notes: PREDIV_A > 4 is required when high-resolution calibration is enabled.

Notes: High-resolution calibration cannot be used in conjunction with smooth digital calibration.

22.3.13. Time-stamp function

Time-stamp function is performed on RTC_TS0 and RTC_TS1 pin and is enabled by control bit TSxRE / TSxFE. It is also enabled by control bit ITSEN.

When a time-stamp event(rising edge or falling edge) occurs on RTC_TS0 or RTC_TS1 pin (TSxRE = 1 / TSxFE = 1), the calendar value will be saved in time-stamp registers (RTC_SSTSxF / RTC_SSTSxR / RTC_TTSxF / RTC_TTSxR / RTC_DTSxF / RTC_DTSxR) and the time-stamp flag (TSF0R / TSF0F / TSF1R / TSF1F) is set to 1 by hardware. Time-stamp event can generate an interrupt if time-stamp interrupt enable (TSIE) is set.

When an internal time-stamp event detected (ITSEN = 1), the calendar value will be saved in time-stamp registers (RTC_SSTSxF / RTC_SSTSxR / RTC_TTSxF / RTC_TTSxR / RTC_DTSxF / RTC_DTSxR), the time-stamp flag (TSF0R / TSF0F / TSF1R / TSF1F) and internal time-stamp flag (ITSF) is set to 1 by hardware. Time-stamp event can generate an interrupt if internal time-stamp interrupt enable (TSIE) is set.The internal timestamp event is generated by the switch to the V_{BAT} supply.

Time-stamp registers only record the calendar at the first time time-stamp event occurs which means that time-stamp registers will not change when time-stamp flag is set.

To extend the time-stamp event source, one optional feature is provided: tamper function can also be considered as time-stamp function if TPTS is set. The calendar value is saved in time-stamp0 rising register.

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

22.3.14. Tamper detection

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

The purposes of the tamper detect configuration are the following:

- The default configuration will erase the RTC backup registers and BKPSRAM.
- It can wakeup from Deep-sleep, Deep-sleep1 and Standby modes, and generate an interrupt.

RTC backup registers (RTC_BKPx)

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

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

Tamper detection function initialization

RTC tamper detection function can be independently enabled on tamper input pin by setting corresponding TPxEN bit. Tamper detection configuration is set before enable TPxEN bit.

The TPxF flag is set after the tamper event occurs on the pin with the following latency:

- When FLT[1:0] is different from 0x0 (Level detection mode with configurable filtering), there are three ck_apre cycles
- When TPTS is set (time-stamp on tamper event), there are three ck_apre cycles
- When FLT[1:0] is reset (Edge detection mode on tamper input detection) and TPTS is reset, there is no latency.

When TPxF is set during the latency, new tamper cannot be detected occurring on the same pin.

Time-stamp on tamper event

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

Edge detection mode on tamper input detection

When FLT[1:0] bit is set to 0x0, the tamper detection is set to edge detection mode and TPxEG bit determines the rising edge or falling edge is the detecting edge. When tamper detection is under edge detection mode, the internal pull-up resistors on the tamper detection input pin are deactivated.

Because of detecting the tamper event will reset the backup registers (RTC_BKPx), writing to the backup register should ensure that the tamper event reset and the writing operation will not occur at the same time, a recommend way to avoid this situation is disable the tamper detection before writing to the backup register and re-enable tamper detection after finish writing.

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

Level detection mode with configurable filtering on tamper input detection

When FLT[1:0] bit is not reset to 0x0, the tamper detection is set to level detection mode and FLT[1:0] bit determines the consecutive number of samples (2, 4 or 8) needed for valid level. When DISPU is set to 0x0(this is default), the internal pull-up resistance will pre-charge the tamper input pin before each sampling and thus larger capacitance is allowed to connect to the tamper input pin. The pre-charge duration is configured through PRCH[1:0] bit. Higher capacitance needs long pre-charge time.

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

22.3.15. Calibration clock output

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

When the COS bit is set to 0(this is default) and asynchronous prescaler is set to 0x7F(FACTOR_A[6:0]), the frequency of RTC_CALIB is f_{rtcclk}/64.When the RTCCLK is 32.768KHz, RTC_CALIB output is corresponding to 512Hz.It's recommend to using rising edge of RTC_CALIB output for there may be a light jitter on falling edge.

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

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1)\times(FACTOR_S+1)} \quad (12\text{-}10)$$

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

22.3.16. Alarm output

When OS[1:0] control bits are not reset, RTC_ALARM alternate function output is enabled. This function will directly output the content of alarm flag or auto wakeup flag bit in RTC_STAT.

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

22.3.17. RTC pin configuration

RTC_OUT, RTC_TS0 and RTC_TAMP0 use the same pin (PC13). Function of PC13 is controlled by the RTC and regardless of PC13 GPIO configuration. The RTC functions of PC13 are available in all low-power modes and in VBAT only mode.

The priority of the PC13 output shown in ***Table 22-1. RTC pin configuration***.

Table 22-1. RTC pin configuration and function

function configuration and pin function	OS[1:0] (output selection)	COEN (calibration output)	TP0EN (tamper0 enable)	TS0REN / TS0FEN (time-stamp0 enable)	ALRMOUTTYPE (RTC_ALARM output type)
Alarm out output open drain	01 or 10 or 11	-	-	-	0
Alarm out output push-pull	01 or 10 or 11	-	-	-	1
Calibration output push-pull	00	1	-	-	-

function configuration and pin function	OS[1:0] (output selection)	COEN (calibration output)	TP0EN (tamper0 enable)	TS0REN / TS0FEN (time-stamp0 enable)	ALRMOUTTYPE (RTC_ALARM output type)
TAMP0 input floating	00	0	1	0	-
Time-stamp and TAMP0 input floating	00	0	1	1	-
Time-stamp input floating	00	0	0	1	-
Standard GPIO	00	0	0	0	-

The PC13 can be used for the following purposes:

- RTC_ALARM output: this output can be RTC Alarm 0, RTC Alarm 1 or RTC Wakeup depending on the OS[1:0] bits in the RTC_CTL register
- RTC_CALIB output: this feature is enabled by setting the COEN bit in the RTC_CTL register
- RTC_TAMP0: tamper event detection
- RTC_TS0: time-stamp 0 event detection

The ALRMOUTTYPE bit in RTC_CFG is used to select whether the output of RTC_ALARM is in push-pull mode or open-drain mode.

The OUT2EN bit in RTC_CFG can be selected to output RTC_OUT to the PB2 or PC13 pins. This output is not available in VBAT / Standby mode.

22.3.18. RTC power saving mode management

Table 22-2. RTC power saving mode management

Mode	Active in Mode	Exit Mode
Sleep	Yes	RTC Interrupts: Alarm 0 / Alarm 1 / Wakeup / Time-stamp / Tamper 0 / Tamper 1
Deep-sleep / Deep-sleep1	Yes: if clock source is LXTAL or IRC40K	RTC Interrupt: Alarm 0 / Alarm 1 / Wakeup / Time-stamp / Tamper 0 / Tamper 1
Standby	Yes: if clock source is LXTAL or IRC40K	RTC Interrupt: Alarm 0 / Alarm 1 / Wakeup / Time-stamp / Tamper 0 / Tamper 1

Note: In the VBAT power supply mode, the timestamp event on the PD4 pin cannot wakeup the standby mode.

22.3.19. RTC interrupts

All RTC interrupts are connected to the EXTI controller.

Below steps should be followed if you want to use the RTC alarm / tamper / time-stamp / auto

wakeup interrupt:

1. Configure and enable the corresponding interrupt line to RTC alarm / tamper / time-stamp/auto wakeup event of EXTI and set the rising edge for triggering
2. Configure and enable the RTC alarm / tamper / time-stamp / auto wakeup interrupt
3. Configure and enable the RTC alarm / tamper / time-stamp / auto wakeup function

Table 22-3. RTC interrupts control

Interrupt	Event flag	Control Bit	Exit Sleep	Exit Deep-sleep And Standby
Alarm 0	ALRM0F	ALRM0IE	Y	Y(1)
Alarm 1	ALRM1F	ALRM1IE	Y	Y(1)
Wakeup	WTF	WTIE	Y	Y(1)
Time-stamp	TSF0R / TSF0F / TSF1R / TSF1F	TSIE	Y	Y(1)
Tamper 0	TP0F	TPIE	Y	Y(1)
Tamper 1	TP1F	TPIE	Y	Y(1)

(1) Only active when RTC clock source is LXTAL or IRC40K.

22.4. Register definition

RTC base address: 0x4000 2800

22.4.1. Time register (RTC_TIME)

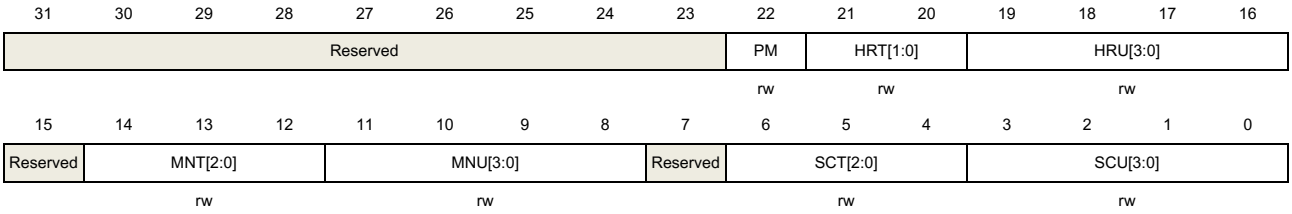
Address offset: 0x00

System reset value: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

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

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



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

22.4.2. Date register (RTC_DATE)

Address offset: 0x04

System reset value: 0x0000 2101 when BPSHAD = 0.

Not affected when BPSHAD = 1.

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

Note: In a non-leap year, February should not be set as the 29th.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								YRT[3:0]				YRU[3:0]			
								rw				rw			

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	YRT[3:0]	Year tens in BCD code
19:16	YRU[3:0]	Year units in BCD code
15:13	DOW[2:0]	Days of the week 000: Reserved 001: Monday … 111: Sunday
12	MONT	Month tens in BCD code
11:8	MONU[3:0]	Month units in BCD code
7:6	Reserved	Must be kept at reset value.
5:4	DAYT[1:0]	Day tens in BCD code
3:0	DAYU[3:0]	Day units in BCD code

22.4.3. Control register (RTC_CTL)

Address offset: 0x08

System reset: not affected

Backup domain reset value: 0x3000 0000

This register is writing protected.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		VCSCAL[1:0]		Reserved	TS1FEN	TS1REN	ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
		rw			rw	rw	rw	rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WTIE	ALRM1IE	ALRM0IE	TS0FEN	WTEN	ALRM1EN	ALRM0EN	Reserved	CS	BPSHAD	REFEN	TS0REN	WTCS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

Bits	Fields	Descriptions

31:30	Reserved	Must be kept at reset value.
29:28	VCSCAL[1:0]	BLDO voltage level selection 00: 0.9V 01: 1.0V 10: 1.1V 11: 1.2V
27	Reserved	Must be kept at reset value.
26	TS1FEN	RTC_TS1 time-stamp event falling edge enable 0: RTC_TS1 time-stamp event falling edge is disable 1: RTC_TS1 time-stamp event falling edge is enable
25	TS1REN	RTC_TS1 time-stamp event rising edge enable 0: RTC_TS1 time-stamp event rising edge is disable 1: RTC_TS1 time-stamp event rising edge is enable
24	ITSEN	Internal time-stamp event enable 0: Disable Internal time-stamp event 1: Enable Internal time-stamp event
23	COEN	Calibration output enable 0: Disable calibration output 1: Enable calibration output
22:21	OS[1:0]	Output selection This bit is used for selecting flag source to output 00: Disable output RTC_ALARM 01: Enable alarm 0 flag output 10: Enable alarm 1 flag output 11: Enable wakeup flag output
20	OPOL	Output polarity This bit is used to invert output RTC_ALARM 0: Disable invert output RTC_ALARM 1: Enable invert output RTC_ALARM
19	COS	Calibration output selection Valid only when COEN=1 and prescalers are at default values 0: Calibration output is 512 Hz 1: Calibration output is 1Hz
18	DSM	Daylight saving mark This bit is flexible used by software. Often can be used to recording the daylight saving hour adjustment.
17	S1H	Subtract 1 hour(winter time change) One hour will be subtracted from current time if it is not 0

		0: No effect 1: 1 hour will be subtracted at next second change time.
16	A1H	Add 1 hour(summer time change) One hour will be added from current time 0: No effect 1: 1 hour will be added at next second change time
15	TSIE	Time-stamp interrupt enable 0: Disable time-stamp interrupt 1: Enable time-stamp interrupt
14	WTIE	Auto-wakeup timer interrupt enable 0: Disable auto-wakeup timer interrupt 1: Enable auto-wakeup timer interrupt
13	ALRM1IE	RTC alarm 1 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
12	ALRM0IE	RTC alarm 0 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
11	TS0FEN	RTC_TS0 time-stamp event falling edge enable 0: RTC_TS0 time-stamp event falling edge is disable 1: RTC_TS0 time-stamp event falling edge is enable
10	WTEN	Auto-wakeup timer function enable 0: Disable function 1: Enable function
9	ALRM1EN	Alarm 1 function enable 0: Disable alarm function 1: Enable alarm function
8	ALRM0EN	Alarm 0 function enable 0: Disable alarm function 1: Enable alarm function
7	Reserved	Must be kept at reset value.
6	CS	Clock System 0: 24-hour format 1: 12-hour format **Note**: Can only be written in initialization state
5	BPSHAD	Shadow registers bypass control 0: Reading calendar from shadow registers

		1: Reading calendar from current real-time calendar **Note**: If frequency of APB clock is less than seven times the frequency of RTCCLK, this bit must set to 1.
4	REFEN	Reference clock detection function enable 0: Disable reference clock detection function 1: Enable reference clock detection function **Note**: Can only be written in initialization state and FACTOR_S[14:0] must be 0x00FF
3	TS0REN	RTC_TS0 time-stamp event rising edge enable 0: RTC_TS0 time-stamp event rising edge is disable 1: RTC_TS0 time-stamp event rising edge is enable
2:0	WTCS[2:0]	Auto-wakeup timer clock selection 000:RTC Clock divided by 16 001:RTC Clock divided by 8 010:RTC Clock divided by 4 011:RTC Clock divided by 2 100,101: ck_spre (default 1Hz) clock 110,111: ck_spre (default 1Hz) clock and 2^{16} is added to wake-up counter.

22.4.4. Status register (RTC_STAT)

Address offset: 0x0C

System reset: Only INITM, INITF and RSYNF bits are set to 0. Others are not affected.

Backup domain reset value: 0x0000 0007

This register is writing protected except RTC_STAT[13:8].

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										HRCPF	TSF1F	TSF1R	TSF0F	ITSF	SCPF
										r	rc_w0	rc_w0	rc_w0	rc_w0	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP1F	Reserved	TP0F	TSOVRF	TSF0R	WTF	ALRM1F	ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	WTWF	ALRM1WF	ALRM0WF
rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	HRCPF	High-resolution calibration pending flag Set to 1 by hardware when software writes to RTC_HRCR without entering initialization mode and set to 0 by hardware when high-resolurtion calibration configuration is taken into account.
20	TSF1F	Time-stamp1 falling edge event flag Set by hardware when time-stamp1 event falling edge is detected.

		Cleared by software writing 0.
19	TSF1R	Time-stamp1 rising edge event flag Set by hardware when time-stamp1 event rising edge is detected. Cleared by software writing 0.
18	TSF0F	Time-stamp 0 falling edge event flag Set by hardware when time-stamp0 event faling edge is detected. Cleared by software writing 0.
17	ITSF	Internal timestamp flag Set by hardware when internal time-stamp event is detected. Cleared by software writing 0, and must be cleared together with TSF0R bit by writing 0 in both bits.
16	SCPF	Smooth calibration pending flag Set to 1 by hardware when software writes to RTC_HRFC without entering initialization mode and set to 0 by hardware when smooth calibration configuration is taken into account.
15	TP1F	RTC_TAMP1 detected flag Set to 1 by hardware when tamper detection is found on tamper1 input pin. Software can clear this bit by writing 0 into this bit.
14	Reserved	Must be kept at reset value.
13	TP0F	RTC_TAMP0 detected flag Set to 1 by hardware when tamper detection is found on tamper0 input pin. Software can clear this bit by writing 0 into this bit.
12	TSOVRF	Time-stamp overflow flag This bit is set by hardware when a time-stamp 0, time-stamp 1 or internal timestamp event is detected if TSF0R, TSF0F, TSF1R, TSF1F or ITSF bit is set before. Cleared by software writing 0.
11	TSF0R	Time-stamp 0 event rising edge flag Set by hardware when time-stamp event is detected. Cleared by software writing 0.
10	WTF	Wakeup timer flag Set by hardware when wakeup timer decreased to 0. Cleared by software writing 0. This flag must be cleared at least 1.5 RTC Clock periods before WTF is set to 1 again.
9	ALRM1F	Alarm 1 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 1 setting value.

		Cleared by software writing 0.
8	ALRM0F	Alarm 0 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 0 setting value. Cleared by software writing 0.
7	INITM	Enter initialization mode 0: Free running mode 1: Enter initialization mode for setting calendar time/date and prescaler. Counter will stop under this mode.
6	INITF	Initialization state flag Set to 1 by hardware and calendar register and prescaler can be programmed in this state. 0: Calendar registers and prescaler register cannot be changed 1: Calendar registers and prescaler register can be changed
5	RSYNF	Register synchronization flag Set to 1 by hardware every 2 RTCCLK which will copy current calendar time/date into shadow register. Initialization mode (INITM), shift operation pending flag (SOPF) or bypass mode (BPSHAD) will clear this bit. This bit is also can be cleared by software writing 0. 0:Shadow register are not yet synchronized 1:Shadow register are synchronized
4	YCM	Year configuration mark Set by hardware if the year field of calendar date register is not the default value 0. 0: Calendar has not been initialized 1: Calendar has been initialized
3	SOPF	Shift function operation pending flag 0: No shift operation is pending 1: Shift function operation is pending
2	WTWF	Wakeup timer write enable flag 0: Wakeup timer update is not allowed 1: Wakeup timer update is allowed
1	ALRM1WF	Alarm 1 configuration can be write flag Set by hardware if alarm register can be wrote after ALRM1EN bit has reset. 0: Alarm registers programming is not allowed 1: Alarm registers programming is allowed
0	ALRM0WF	Alarm 0 configuration can be write flag Set by hardware if alarm register can be wrote after ALRM0EN bit has reset. 0: Alarm registers programming is not allowed.

1: Alarm registers programming is allowed.

22.4.5. Prescaler register (RTC_PSC)

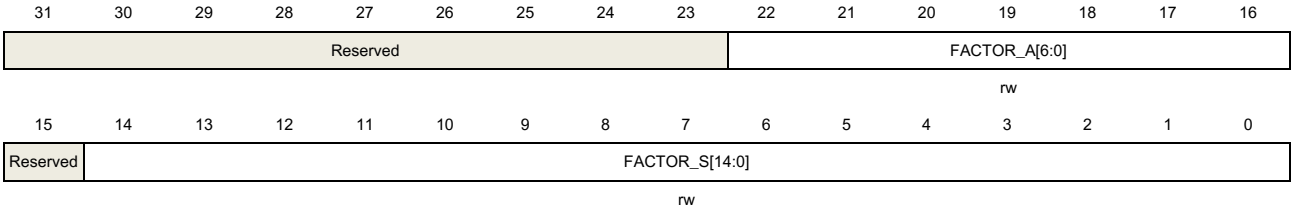
Address offset: 0x10

System reset: not effected

Backup domain reset value: 0x007F 00FF

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

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



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

22.4.6. Wakeup timer register (RTC_WUT)

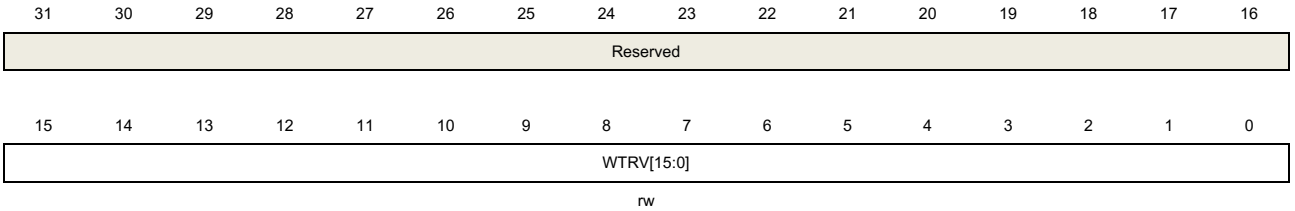
Address offset: 0x14

System reset: not effected

Backup domain reset value: 0x0000 FFFF

This register is writing protected.

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



Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	WTRV[15:0]	Auto-wakeup timer reloads value. Every (WTRV[15:0]+1) ck_wut period the WTF bit is set after WTEN=1.The ck_wut

is selected by WTCS[2:0] bits.

Note: This configure case is forbidden: WTRV=0x0000 with WTCS[2:0]=0b011.

This register can be written only when WTWF=1.

22.4.7. Alarm 0 time and date register (RTC_ALRM0TD)

Address offset: 0x1C

System reset: not effect

Backup domain reset value: 0x0000 0000

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

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

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

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

11:8	MNU[3:0]	Minutes units in BCD code
7	MSKS	Alarm second mask bit 0: Not mask second field 1: Mask second field
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

22.4.8. Alarm 1 time and date register (RTC_ALRM1TD)

Address offset: 0x20

System reset: not effect

Backup domain reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31	MSKD	Alarm date mask bit 0: Not mask date/day field 1: Mask date/day field
30	DOWS	Day of the week selected 0: DAYU[3:0] indicates the date units 1: DAYU[3:0] indicates the week day and DAYT[3:0] has no means.
29:28	DAYT[1:0]	Day tens in BCD code
27:24	DAYU[3:0]	Day units or week day in BCD code
23	MSKH	Alarm hour mask bit 0: Not mask hour field 1: Mask hour field
22	PM	AM/PM flag 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code

15	MSKM	Alarm minutes mask bit 0: Not mask minutes field 1: Mask minutes field
14:12	MNT[2:0]	Minutes tens in BCD code
11:8	MNU[3:0]	Minutes units in BCD code
7	MSKS	Alarm second mask bit 0: Not mask second field 1: Mask second field
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

22.4.9. Write protection key register (RTC_WPK)

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
Reserved								WPK[7:0]							
								w							

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

22.4.10. Sub second register (RTC_SS)

Address offset: 0x28

System reset value: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

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

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

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

Bits	Fields	Descriptions

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

22.4.11. Shift function control register (RTC_SHIFTCTL)

Address offset: 0x2C
System reset: not effect
Backup Reset value: 0x0000 0000
This register is writing protected and can only be wrote when SOPF=0.

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A1S	Reserved														
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SFS[14:0]														
	w														

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

22.4.12. Time of time-stamp 0 rising register (RTC_TTS0R)

Address offset: 0x30
Backup domain reset value: 0x0000 0000
System reset: no effect
This register will record the calendar time when TSF0R is set to 1.
Reset TSF0R bit will also clear this register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			

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

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

22.4.13. Date of time-stamp 0 rising register (RTC_DTS0R)

Address offset: 0x34

Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar date when TSF0R is set to 1.

Reset TSF0R bit will also clear this register.

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

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

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

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

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

22.4.14. Sub second of time-stamp 0 rising register (RTC_SSTS0R)

Address offset: 0x38

Backup domain reset: 0x0000 0000

System reset: no effect

This register will record the sub second of time-stamp 0 when TSF0R is set to 1.

Reset TSF0R bit will also clear this register.

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF0R is set to 1.

22.4.15. High resolution frequency compensation register (RTC_HRFC)

Address offset: 0x3C

Backup domain reset: 0x0000 0000

System Reset: no effect

This register is write protected.

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQI	CWND8	CWND16	MWE	Reserved			CMSK[8:0]								
rw	rw	rw	rw				rw								

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	FREQI	Increase RTC frequency by 488.5PPM 0: No effect 1: One RTCCLK pulse is inserted every 2^{11} pulses. This bit should be used in conjunction with CMSK[8:0] bit. If the input clock frequency is 32.768KHz, the number of RTCCLK pulses added during 32s calibration window is (512 * FREQI) - CMSK[8:0].
14	CWND8	Frequency compensation window 8 second selected 0: No effect 1: Calibration window is 8 second **Note**: When CWND8=1, CMSK[8:0] are stuck at "00".
13	CWND16	Frequency compensation window 16 second selected 0: No effect 1: Calibration window is 16 second **Note**: When CWND16=1, CMSK[8:0] are stuck at "0".
12	MWE	Multiple windows calibration enable. 0: Single window mode 1: Multiple windows mode In multiple windows mode, the calibration will start again at the end of current calibration window. During continuous smoothing calibration, closing the smoothing calibration requires writing 0 to MWE. While writing 0 to MWE, other bit fields in the RTC_HRFC register must not be changed.
11:9	Reserved	Must be kept at reset value.
8:0	CMSK[8:0]	Calibration mask number The number of mask pulse out of 2^{20} RTCCLK pulse. This feature will decrease the frequency of calendar with a resolution of 0.9537 PPM.

22.4.16. Tamper register (RTC_TAMP)

Address offset: 0x40

Backup domain reset: 0x0000 0000

System reset: no effect

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

This register has 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
DISPU	PRCH[1:0]	FLT[1:0]	FREQ[2:0]	TPTS	TP1EG	TP1EN	Reserved	TPIE	TP0EG	TP0EN
rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	DISPU	RTC_TAMPx pull up disable bit 0: Enable inner pull-up before sampling for pre-charge RTC_TAMPx pin 1: Disable pre-charge duration
14:13	PRCH[1:0]	Pre-charge duration time of RTC_TAMPx This setting determines the pre-charge time before each sampling. 00: 1 RTC clock 01: 2 RTC clock 10: 4 RTC clock 11: 8 RTC clock
12:11	FLT[1:0]	RTC_TAMPx filter count setting This bit determines the tamper sampling type and the number of consecutive sample. 00: Detecting tamper event using edge mode. Pre-charge duration is disabled automatically 01: Detecting tamper event using level mode.2 consecutive valid level samples will make an effective tamper event 10: Detecting tamper event using level mode.4 consecutive valid level samples will make an effective tamper event 11: Detecting tamper event using level mode.8 consecutive valid level samples will make an effective tamper event
10:8	FREQ[2:0]	Sampling frequency of tamper event detection 000: Sample once every 32768 RTCCLK(1Hz if RTCCLK=32.768KHz) 001: Sample once every 16384 RTCCLK(2Hz if RTCCLK=32.768KHz) 010: Sample once every 8192 RTCCLK(4Hz if RTCCLK=32.768KHz) 011: Sample once every 4096 RTCCLK(8Hz if RTCCLK=32.768KHz) 100: Sample once every 2048 RTCCLK(16Hz if RTCCLK=32.768KHz) 101: Sample once every 1024 RTCCLK(32Hz if RTCCLK=32.768KHz) 110: Sample once every 512 RTCCLK(64Hz if RTCCLK=32.768KHz) 111: Sample once every 256 RTCCLK(128Hz if RTCCLK=32.768KHz)
7	TPTS	Make tamper function used for time-stamp 0 rising function 0: No effect 1: TSF0R is set when tamper0 or tamper1 event detected even TS0REN=0
6	TP1EG	Tamper 1 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event

		1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
5	TP1EN	Tamper 1 detection enable 0: Disable tamper 1 detection function 1: Enable tamper 1 detection function
4:3	Reserved	Must be kept at reset value.
2	TPIE	Tamper detection interrupt enable 0: Disable tamper interrupt 1: Enable tamper interrupt
1	TP0EG	Tamper 0 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
0	TP0EN	Tamper 0 detection enable 0: Disable tamper 0 detection function 1: Enable tamper 0 detection function

22.4.17. Alarm 0 sub second register (RTC_ALRM0SS)

Address offset: 0x44

Backup domain reset: 0x0000 0000

System reset: no effect

This register is write protected and can only be wrote when ALRM0EN=0 or INITM=1.

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	MSKSSC[3:0]	Mask control bit of SSC 0000: Mask alarm sub second setting. The alarm asserts at every second time point if all the rest alarm fields are matched.

		0001: SSC[0] is to be compared and all others are ignored 0010: SSC[1:0] is to be compared and all others are ignored 0011: SSC[2:0] is to be compared and all others are ignored 0100: SSC[3:0] is to be compared and all others are ignored 0101: SSC[4:0] is to be compared and all others are ignored 0110: SSC[5:0] is to be compared and all others are ignored 0111: SSC[6:0] is to be compared and all others are ignored 100: SSC[7:0] is to be compared and all others are ignored 101: SSC[8:0] is to be compared and all others are ignored 1010: SSC[9:0] is to be compared and all others are ignored 1011: SSC[10:0] is to be compared and all others are ignored 1100: SSC[11:0] is to be compared and all others are ignored 1101: SSC[12:0] is to be compared and all others are ignored 1110: SSC[13:0] is to be compared and all others are ignored 1111: SSC[14:0] is to be compared and all others are ignored **Note**: The bit 15 of synchronous counter (SSC[15] in RTC_SS) is never compared.
23:15	Reserved	Must be kept at reset value.
14:0	SSC[14:0]	Alarm sub second value This value is the alarm sub second value which is to be compared with synchronous prescaler counter SSC. Bit number is controlled by MSKSSC bits.

22.4.18. Alarm 1 sub second register (RTC_ALRM1SS)

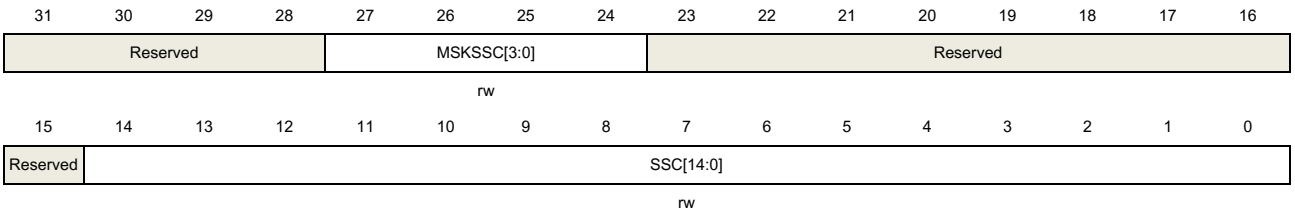
Address offset: 0x48

Backup domain reset: 0x0000 0000

System reset: no effect

This register is write protected and can only be wrote when ALRM1EN=0 or INITM=1.

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



Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	MSKSSC[3:0]	Mask control bit of SSC 0000: Mask alarm sub second setting. The alarm asserts at every second time point if all the rest alarm fields are matched. 0001: SSC[0] is to be compared and all others are ignored 0010: SSC[1:0] is to be compared and all others are ignored

		0011: SSC[2:0] is to be compared and all others are ignored 0100: SSC[3:0] is to be compared and all others are ignored 0101: SSC[4:0] is to be compared and all others are ignored 0110: SSC[5:0] is to be compared and all others are ignored 0111: SSC[6:0] is to be compared and all others are ignored 1000: SSC[7:0] is to be compared and all others are ignored 1001: SSC[8:0] is to be compared and all others are ignored 1010: SSC[9:0] is to be compared and all others are ignored 1011: SSC[10:0] is to be compared and all others are ignored 1100: SSC[11:0] is to be compared and all others are ignored 1101: SSC[12:0] is to be compared and all others are ignored 1110: SSC[13:0] is to be compared and all others are ignored 1111: SSC[14:0] is to be compared and all others are ignored **Note**: The bit 15 of synchronous counter (SSC[15] in RTC_SS) is never compared.
23:15	Reserved	Must be kept at reset value.
14:0	SSC[14:0]	Alarm sub second value This value is the alarm sub second value which is to be compared with synchronous prescaler counter SSC. Bit number is controlled by MSKSSC bits.

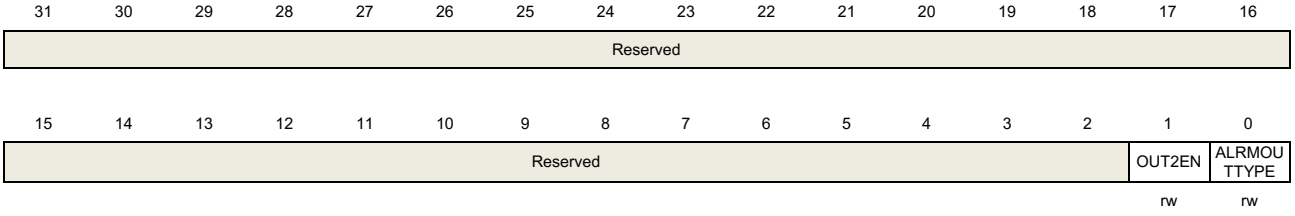
22.4.19. Configuration register (RTC_CFG)

Address offset: 0x4C

Backup domain reset: 0x0000 0000

System reset: no effect

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OUT2EN	RTC_OUT pin select 0: RTC_OUT is output on PC13 1: RTC_OUT is output on PB2
0	ALRMOUTTYPE	RTC_ALARM Output Type 0: Open-drain output type 1: Push-pull output type

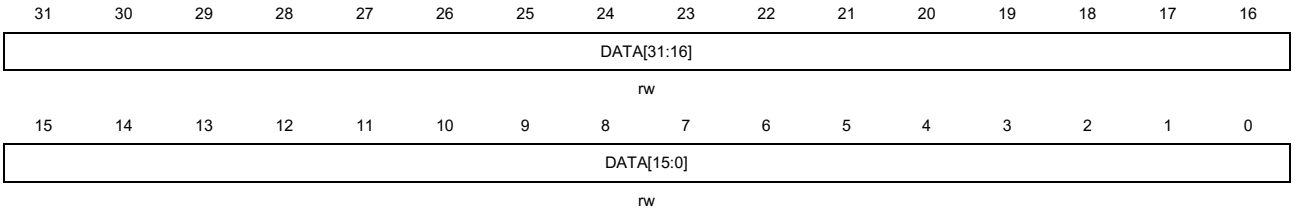
22.4.20. Backup registers (RTC_BKPx) (x=0..3)

Address offset: 0x50~0x5C

Backup domain reset: 0x0000 0000

System reset: no effect

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



Bits	Fields	Descriptions
31:0	DATA[31:0]	Data These registers can be wrote or read by software. The content remains valid even in power saving mode because they can powered-on by VBAT. Tamper detection flag TPxF assertion will reset these registers.

22.4.21. Time of time-stamp 0 falling register (RTC_TTS0F)

Address offset: 0xD0

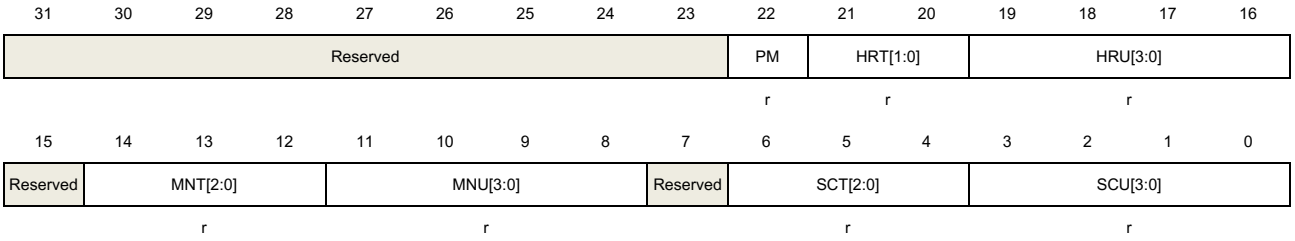
Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar time when TSF0F is set to 1.

Reset TSF0F bit will also clear this register.

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



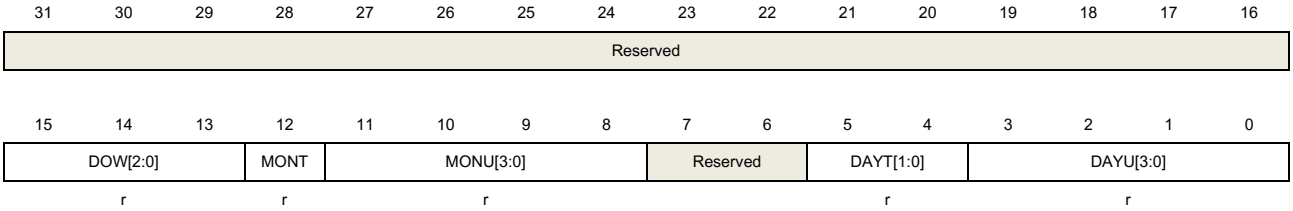
Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PM	AM/PM mark 0:AM or 24-hour format 1:PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code

15	Reserved	Must be kept at reset value.
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code
7	Reserved	Must be kept at reset value.
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

22.4.22. Date of time-stamp 0 falling register (RTC_DTS0F)

Address offset: 0xD4
Backup domain reset value: 0x0000 0000
System reset: no effect
This register will record the calendar date when TSF0F is set to 1.
Reset TSF0F bit will also clear this register.

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



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

22.4.23. Sub second of time-stamp 0 falling register (RTC_SSTS0F)

Address offset: 0xD8
Backup domain reset: 0x0000 0000
System reset: no effect
This register will record the sub second of time-stamp 0 when TSF0F is set to 1.
Reset TSF0F bit will also clear this register.

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF0F is set to 1.

22.4.24. Time of time-stamp 1 rising register (RTC_TTS1R)

Address offset: 0xDC

Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar time when TSF1R is set to 1.

Reset TSF1R bit will also clear this register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			

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

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

6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

22.4.25. Date of time-stamp 1 rising register (RTC_DTS1R)

Address offset: 0xE0

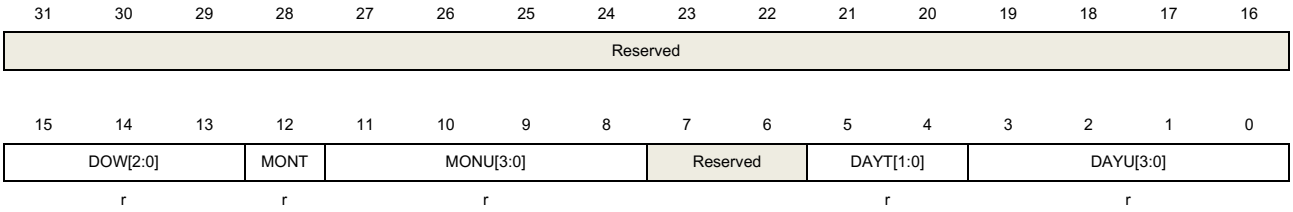
Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar date when TSF1R is set to 1.

Reset TSF1R bit will also clear this register.

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



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

22.4.26. Sub second of time-stamp 1 rising register (RTC_SSTS1R)

Address offset: 0xE4

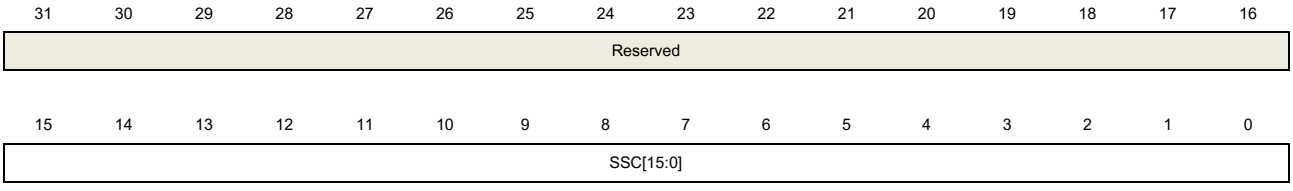
Backup domain reset: 0x0000 0000

System reset: no effect

This register will record the sub second of time-stamp 1 when TSF1R is set to 1.

Reset TSF1R bit will also clear this register.

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



r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF1R is set to 1.

22.4.27. Time of time-stamp 1 falling register (RTC_TTS1F)

Address offset: 0xE8
Backup domain reset value: 0x0000 0000
System reset: no effect
This register will record the calendar time when TSF1F is set to 1.
Reset TSF1F bit will also clear this register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			

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

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

22.4.28. Date of time-stamp 1 falling register (RTC_DTS1F)

Address offset: 0xEC

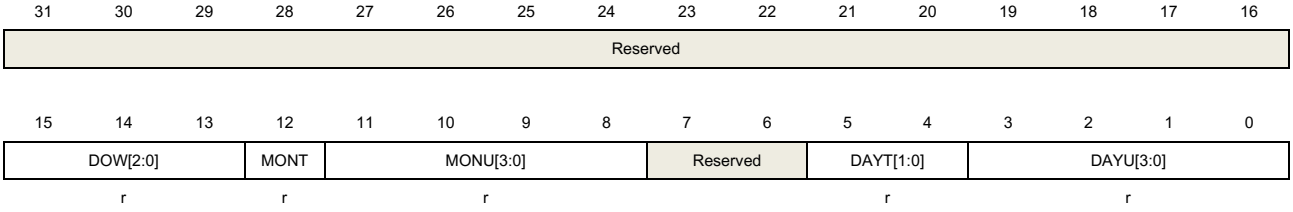
Backup domain reset value: 0x0000 0000

System reset: no effect

This register will record the calendar date when TSF1F is set to 1.

Reset TSF1F bit will also clear this register.

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



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

22.4.29. Sub second of time-stamp 1 falling register (RTC_SSTS1F)

Address offset: 0xF0

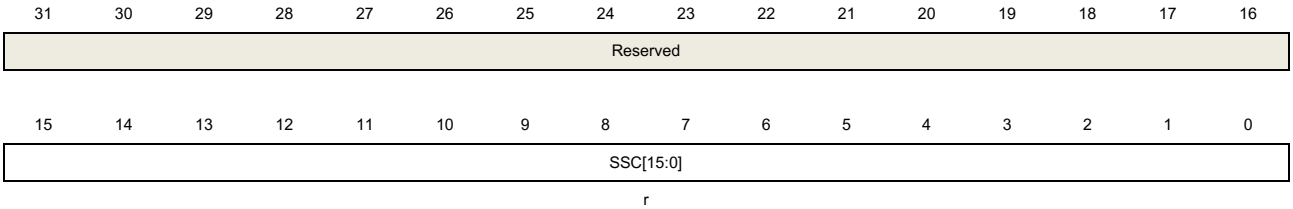
Backup domain reset: 0x0000 0000

System reset: no effect

This register will record the sub second of time-stamp 1 when TSF1F is set to 1.

Reset TSF1F bit will also clear this register.

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



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

15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF1F is set to 1.

22.4.30. High resolution clock frequence calibration register(RTC_HRCFC)

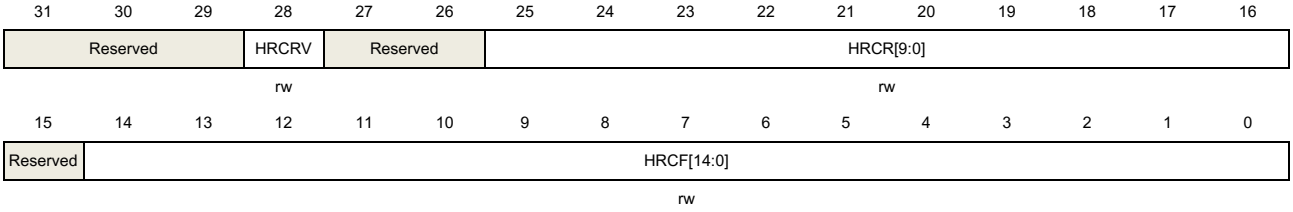
Address offset: 0xF4

Backup domain reset: 0x0000 3E80

System reset: no effect

This register is write protected.

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



Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	HRCRV	HRCR[9:0] valid 0: The ratio value is calculated by the hardware according to the formula: HRCF[14:0]/32768. 1: The ratio value is configured by software
27:26	Reserved	Must be kept at reset value.
25:16	HRCR[9:0]	The ratio of high-resolution calibration clock frequency to the RTC clock frequency. HRCR[9:0] is valid if HRCRV bit is set.
15	Reserved	Must be kept at reset value.
14:0	HRCF[14:0]	High resolution calibration clock frequency The unit of frequency is KHz. HRCF[14:0] must be writtern in correct value before configure RTC_HRFC register.

22.4.31. High-resolution calibration register(RTC_HRCR)

Address offset: 0xF8

Backup domain reset: 0x0000 0000

System reset: no effect

This register is write protected.

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



HRCP	HRC_MASK[14:0]
rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	HRCE	High-resolution calibration enable 0: Disable high-resolution calibration 1: Enable high-resolution calibration
27	Reserved	Must be kept at reset value.
26	HRCADD	The direction of adjusting HRCN[9:0] high-frequency clock cycles for RTC prescaler clock ck_spre. 0: Each ck_spre subtract HRCN[9:0] high-frequency clock cycles 1: Each ck_spre add HRCN[9:0] high-frequency clock cycles
25:16	HRCN[9:0]	Number of high-frequency clock cycles for high-resolution calibration adjustment
15	HRCP	The prescaler clock ck_spre reduces one RTC clock cycle per sub-second.
14:0	HRC_MASK[14:0]	Prescaler clock ck_spre increases HRC_MASK[14:0] RTC clock cycles. HRC_MASK[14:0] should not be writtern into value greater than FACTOR_S[14:0] bit in RTC_PSC register.

23. TIMER (TIMERx)

Table 23-1. Timers (TIMERx) are divided into four sorts

TIMER		TIMER0/7/19	TIMER1/2/3/4	TIMER9/10/12/13	TIMER15/40/41/42/43/44	TIMER5/6/50/51
TYPE		Advanced	General-L0	General-L2	General-L3	Basic
Prescaler		16-bit	16-bit	16-bit	16-bit	16-bit
Counter		16-bit	16-bit (TIMER2/3/4) 32-bit (TIMER1)	16-bit	16-bit	16-bit
Count mode		UP, DOWN, Center-aligned	UP, DOWN, Center-aligned	UP	UP, DOWN, Center-aligned	UP ONLY
Repetition		•	×	×	•	×
Channel Capture/Compare		4	4	1	3	0
Composite PWM mode		•	x	×	•	×
Asymmetric PWM mode		•	×	×	×	×
Output match pulse select		•	x	×	•	×
Complementary & Dead-time		•	×	×	•	×
Break function	**BREAK0**	•	×	×	•	×

TIMER		TIMER0/7/19	TIMER1/2/3/4	TIMER9/10/12/13	TIMER15/40/41/42/43/44	TIMER5/6/50/51
	Channel BREAK	●	×	×	×	×
Single Pulse		●	●	●	●	●
Quadrature decoder		●	●	×	×	×
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		
	TIMER19	TIMER0_TRGO	TIMER1_TRGO	TIMER4_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)	TIMER9	TRIGSEL				TRIGSEL
	TIMER10	TRIGSEL				
	TIMER12	TRIGSEL				
	TIMER13	TRIGSEL				

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI14
(4)	TIMER14	TIMER0_TRGO	TIMER2_TRGO	TIMER12_TRGO	TIMER40_CH0	
	TIMER40	TIMER0_TRGO	TIMER2_TRGO	TIMER13_TRGO	TIMER14_CH0	
	TIMER41	TIMER0_TRGO	TIMER2_TRGO	TIMER9_TRGO	TIMER13_TRGO	
	TIMER42	TIMER0_TRGO	TIMER2_TRGO	TIMER10_TRGO	TIMER13_TRGO	
	TIMER43	TIMER0_TRGO	TIMER2_TRGO	TIMER10_TRGO	TIMER13_TRGO	
	TIMER44	TIMER0_TRGO	TIMER2_TRGO	TIMER12_TRGO	TIMER13_TRGO	
(5)	Only update events will generate a DMA request. TIMER5/6 do not have DMAS bit (DMA request source selection).					

23.1. Advanced timer (TIMERx, x=0, 7, 19)

23.1.1. Overview

The advanced timer module (TIMER0/7/19) 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.

23.1.2. Characteristics

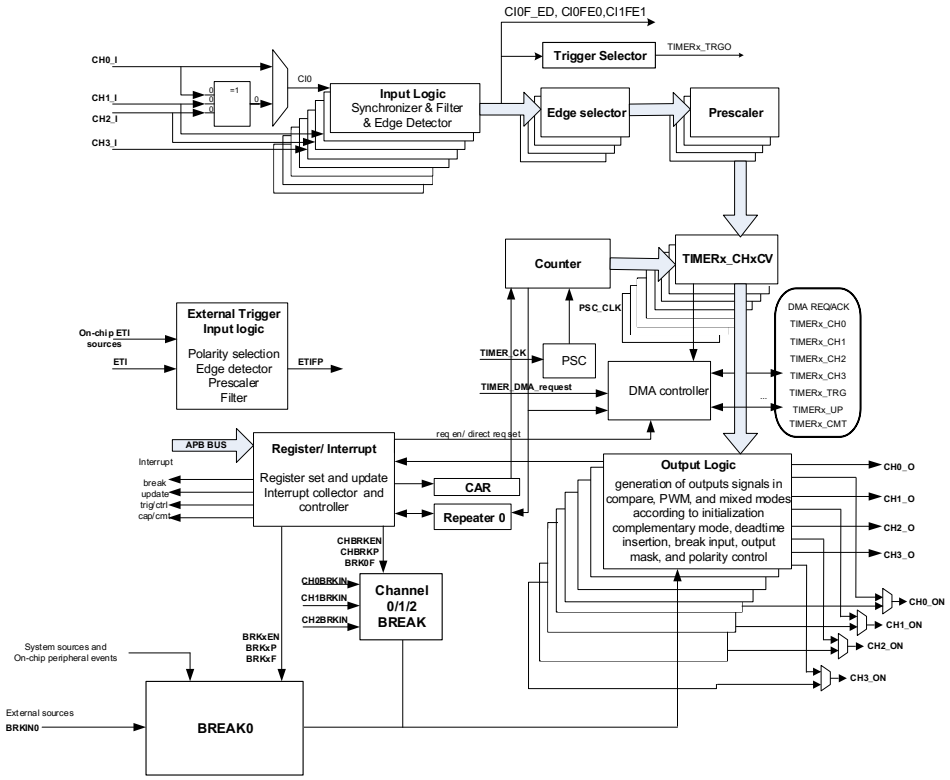
- Total channel num: 4.
- Counter width: 16 bits.
- Selectable clock source: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation 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: BREAK0 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.

23.1.3. Block diagram

Figure 23-1. Advanced timer block diagram provides details of the internal configuration of the advanced timer.

Figure 23-1. Advanced timer block diagram



23.1.4. 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=0,7,19) registers.

- TSCFGy[4:0] (y=0..6) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7,19) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

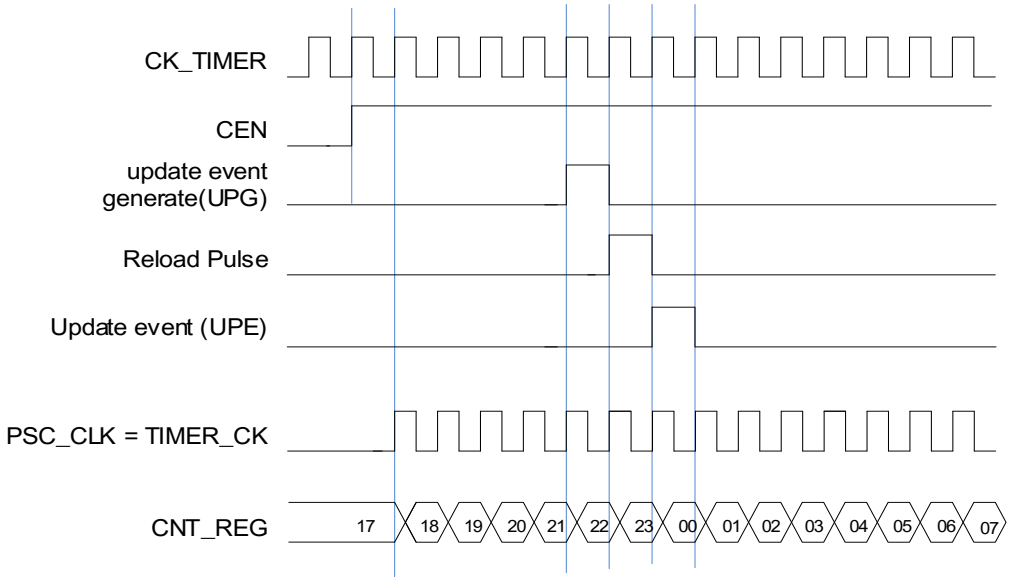
The default clock source is the CK_TIMER for driving the counter prescaler when

TSCFGy[4:0] (y=0..6) = 5'b00000 in SYSCFG_TIMERxCFG(x=0,7,19) 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=0,7,19) 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 23-2. 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.

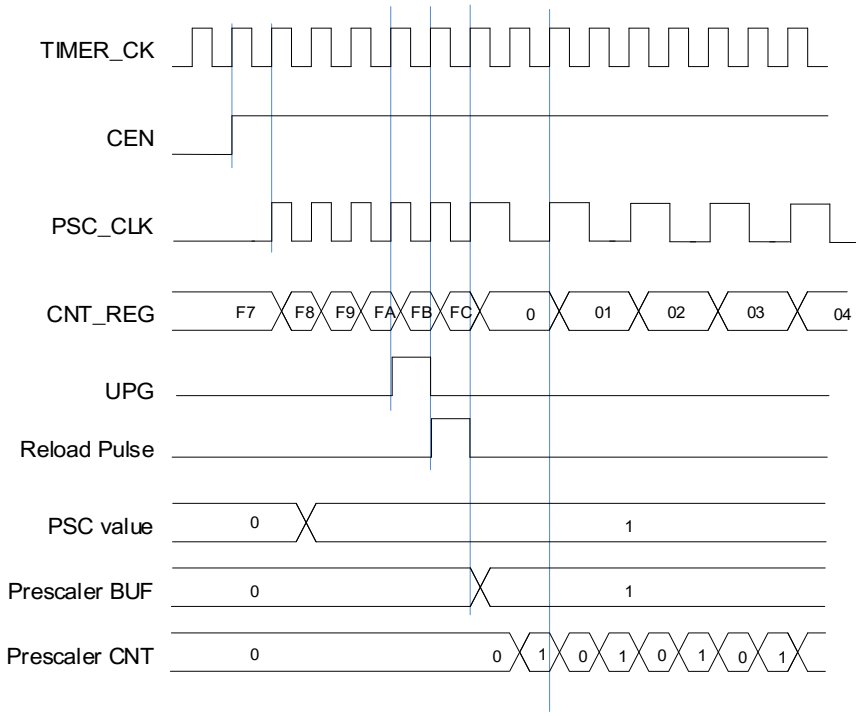
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

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 23-3. 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 23-4. Timing diagram of up counting mode, PSC=0/2 and *Figure 23-5. 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 23-4. Timing diagram of up counting mode, PSC=0/2

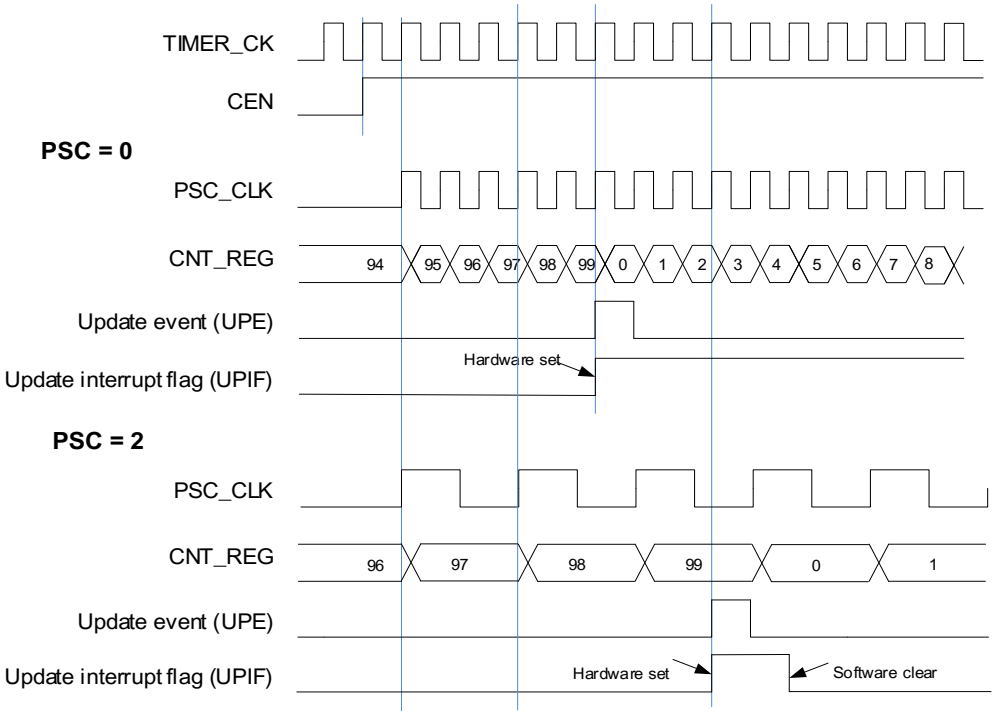
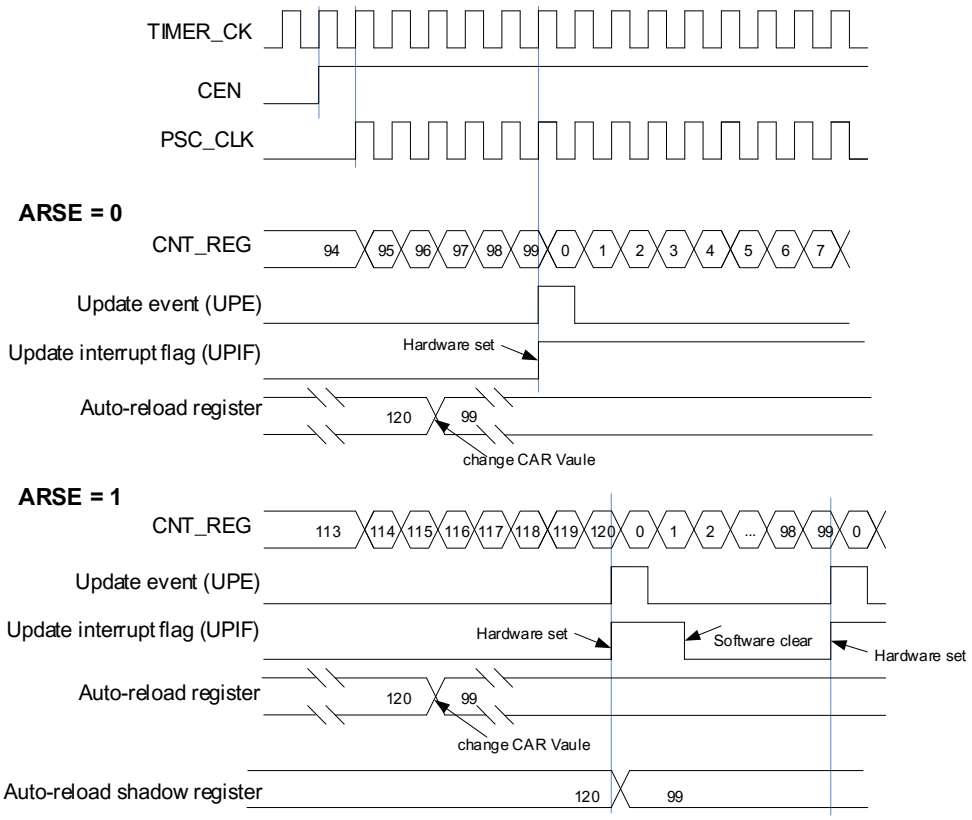


Figure 23-5. Timing diagram of up counting mode, change TIMERx_CAR ongoing



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 23-6. Timing diagram of down counting mode, PSC=0/2 and *Figure 23-7. 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 23-6. Timing diagram of down counting mode, PSC=0/2

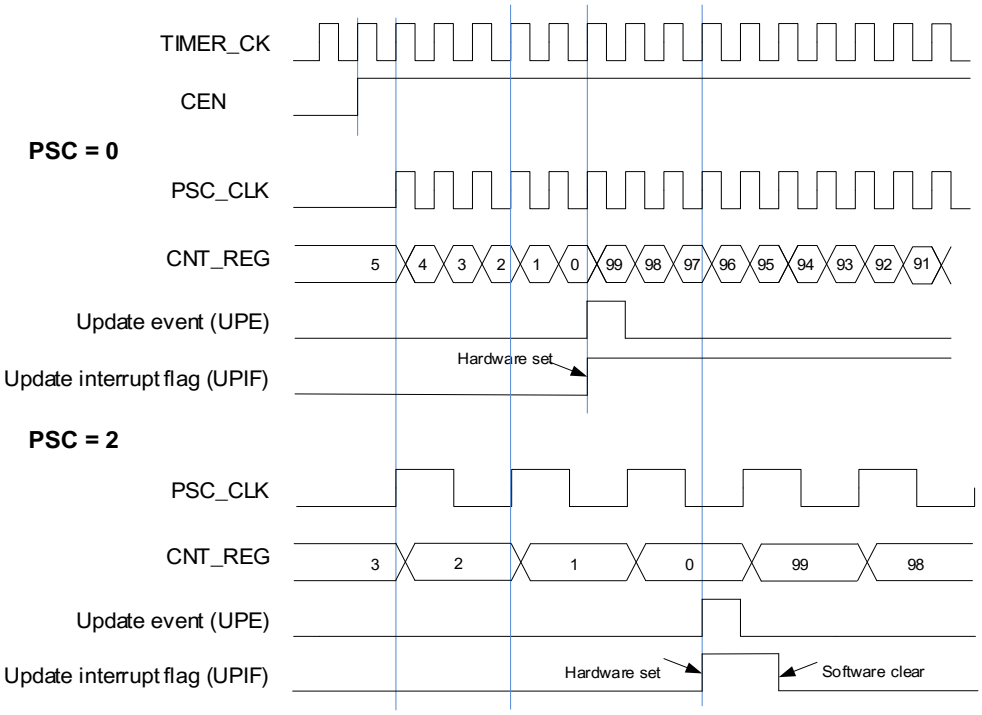
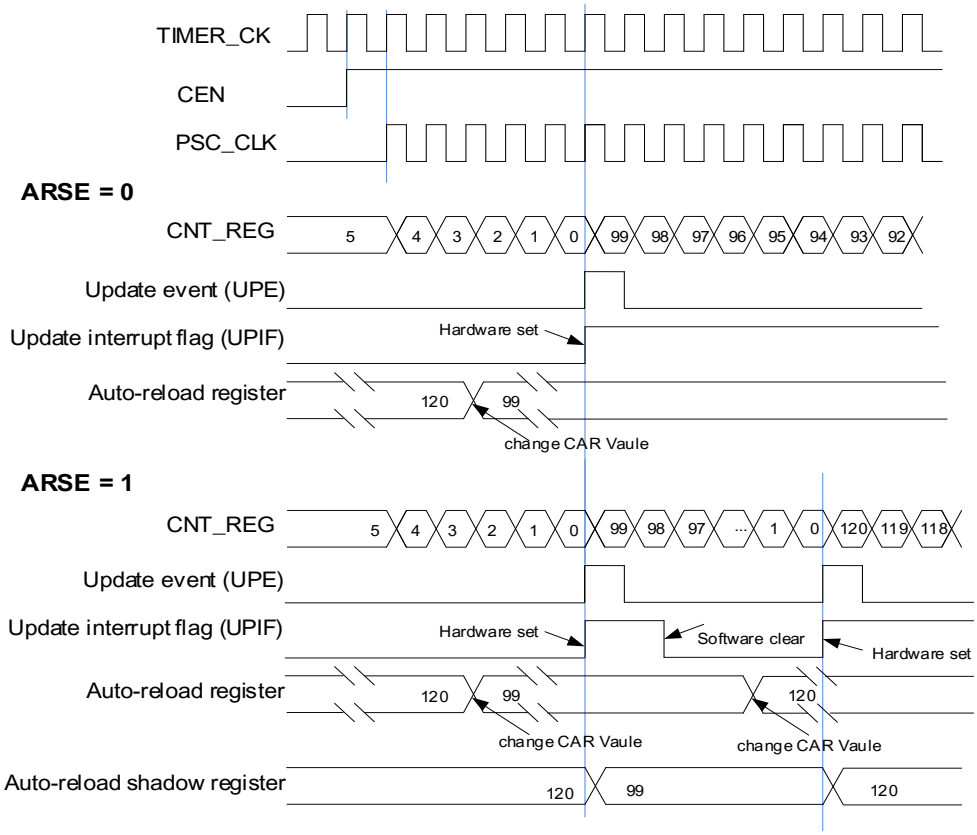


Figure 23-7. Timing diagram of down counting mode, change TIMERx_CAR ongoing



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 23-8. Timing diagram of center-aligned counting mode***.

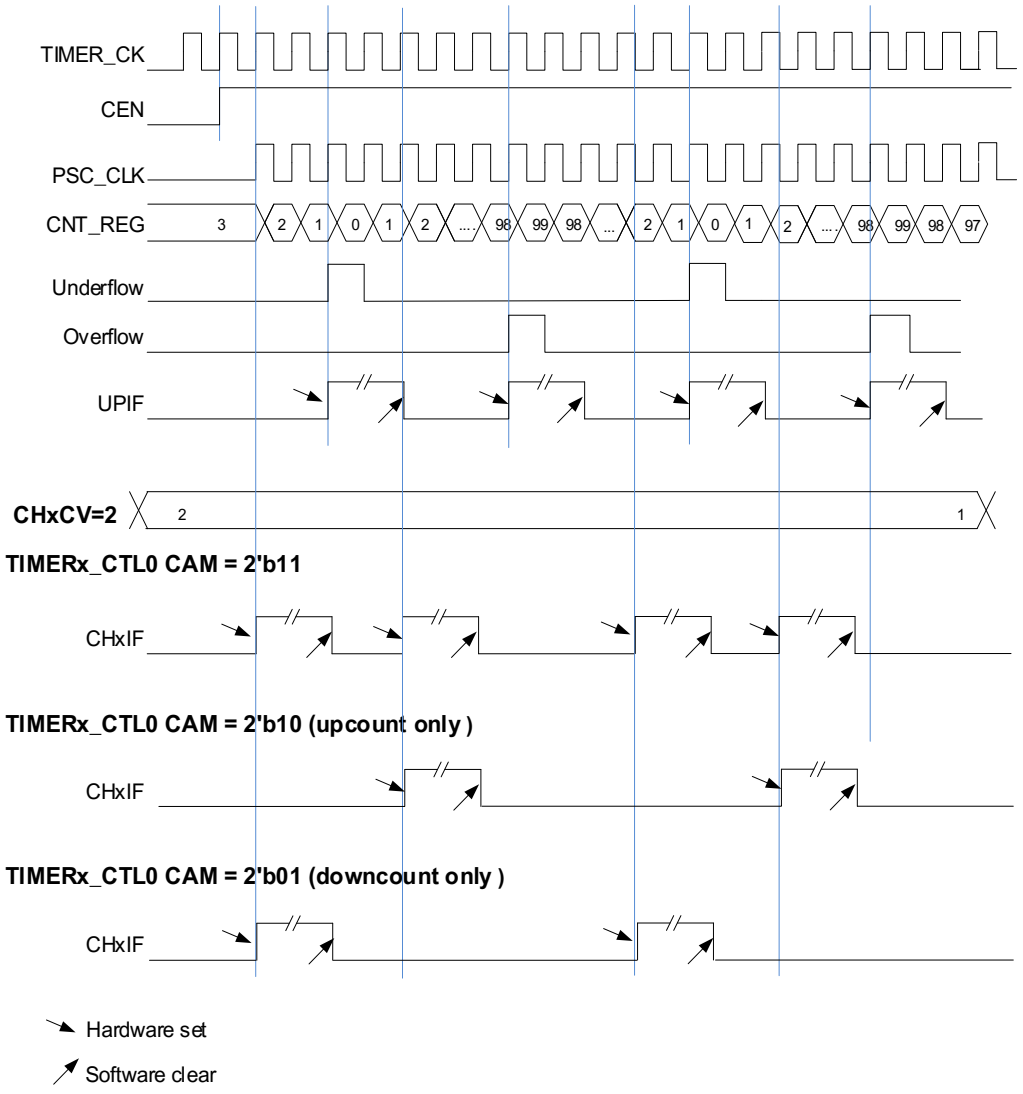
If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, auto-reload register,

prescaler register) are updated.

Figure 23-8. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 23-8. 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 23-9. Repetition counter timing diagram of center-aligned counting mode

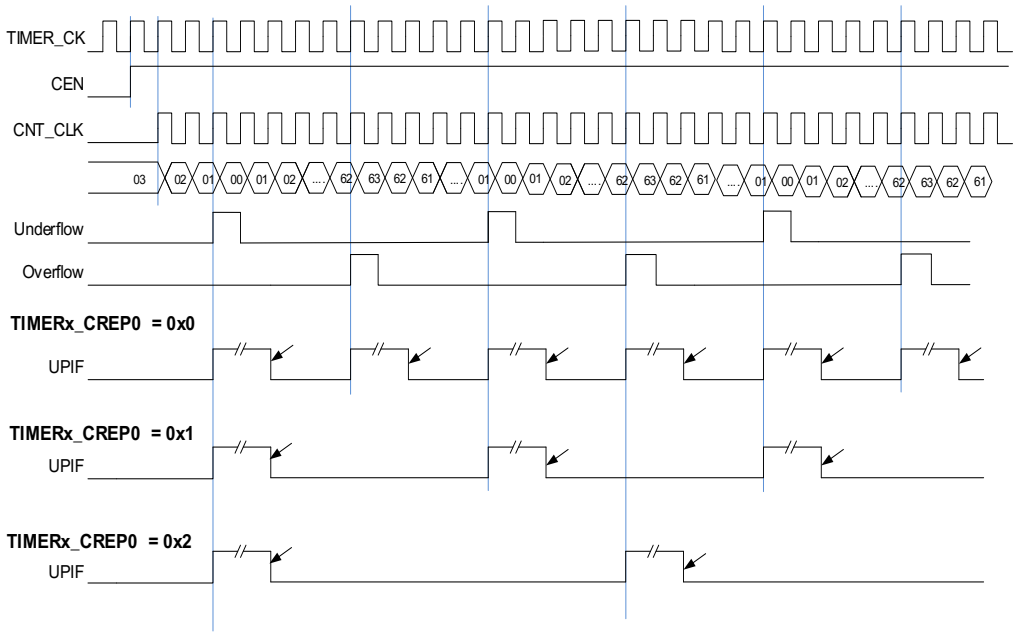


Figure 23-10. Repetition counter timing diagram of up counting mode

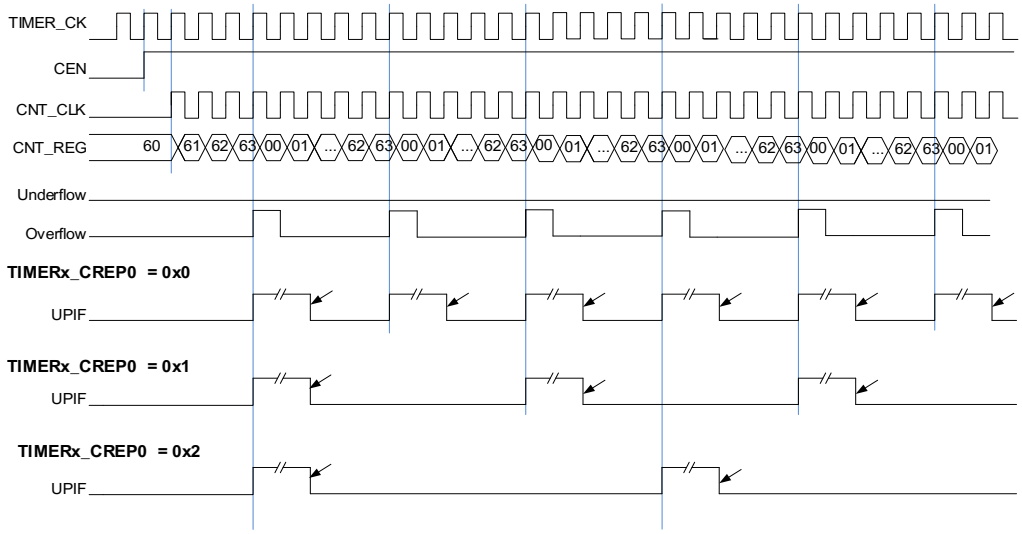
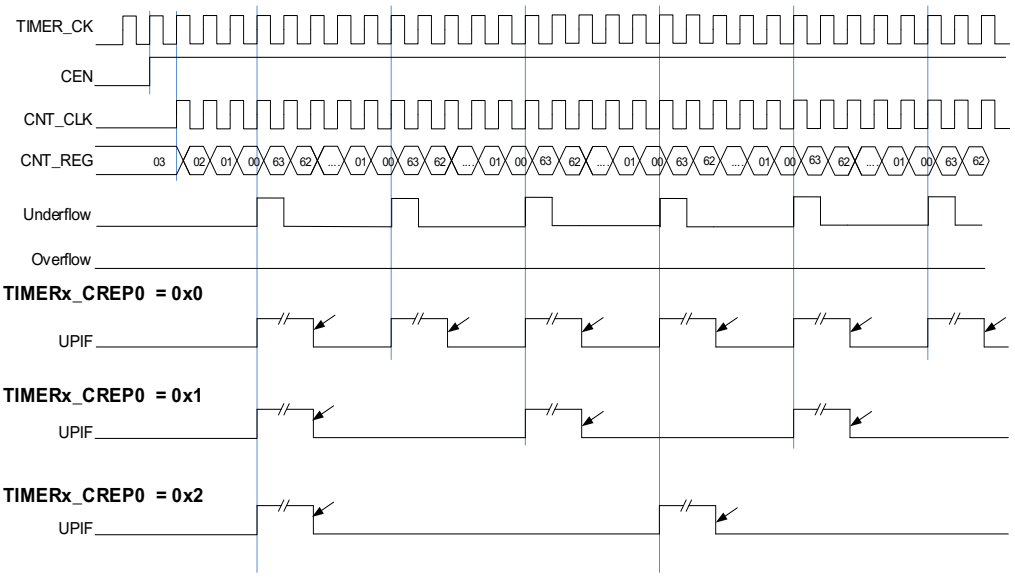


Figure 23-11. Repetition counter timing diagram of down counting mode



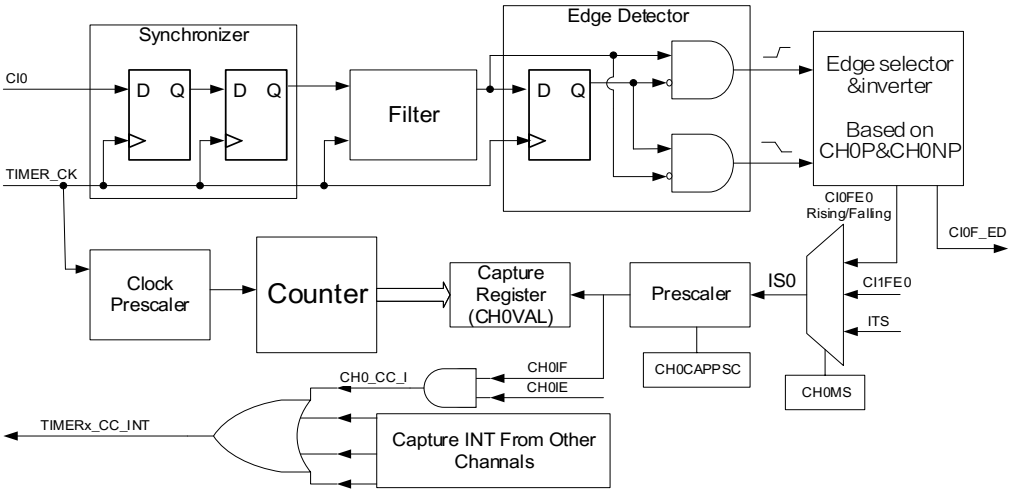
Capture/compare channels

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 23-12. 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.

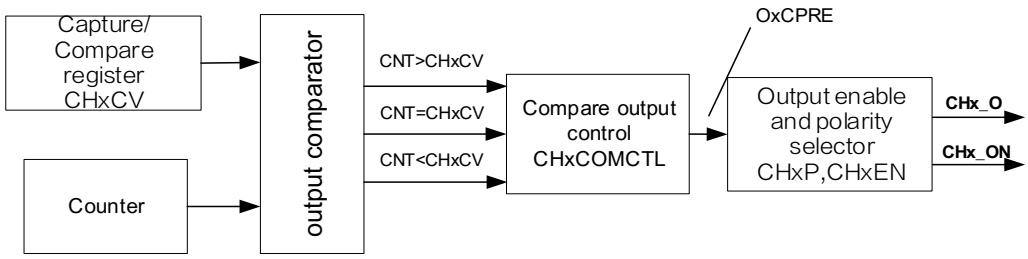
Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

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 23-13. Output compare logic showS the logic circuit of output compare mode.

Figure 23-13. 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.

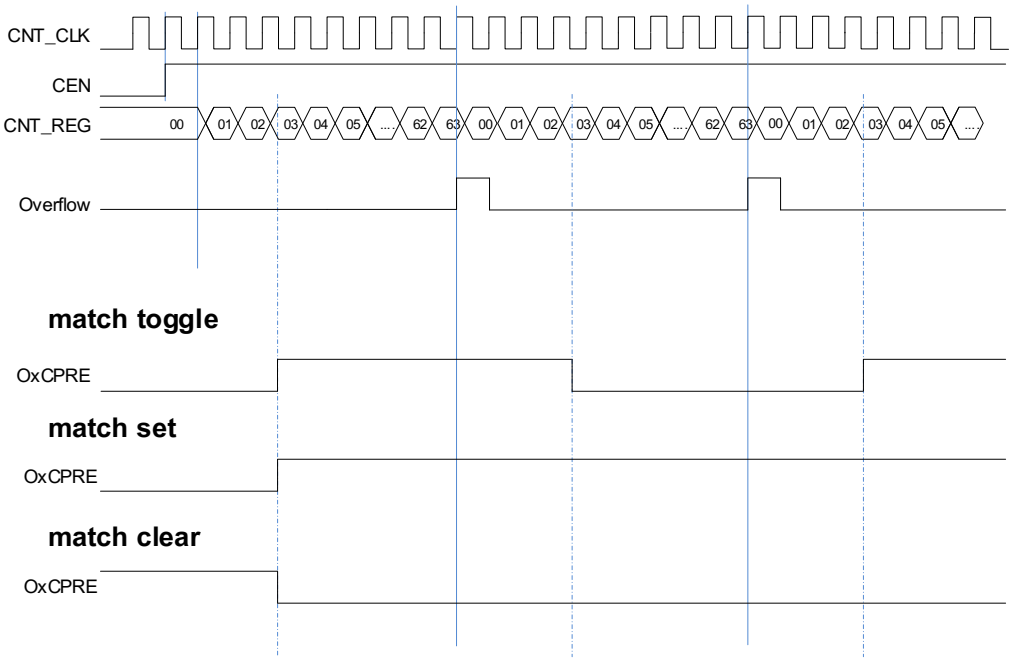
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.

The TIMERx_CHxCV can be changed ongoing to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 23-14. Output-compare in three modes shows the three compare modes: toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 23-14. 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 23-15. 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 23-16. 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 23-15. Timing diagram of EAPWM

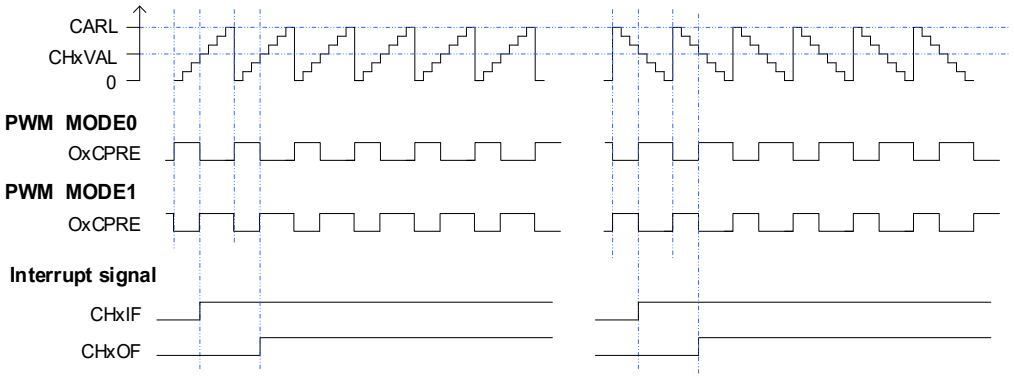
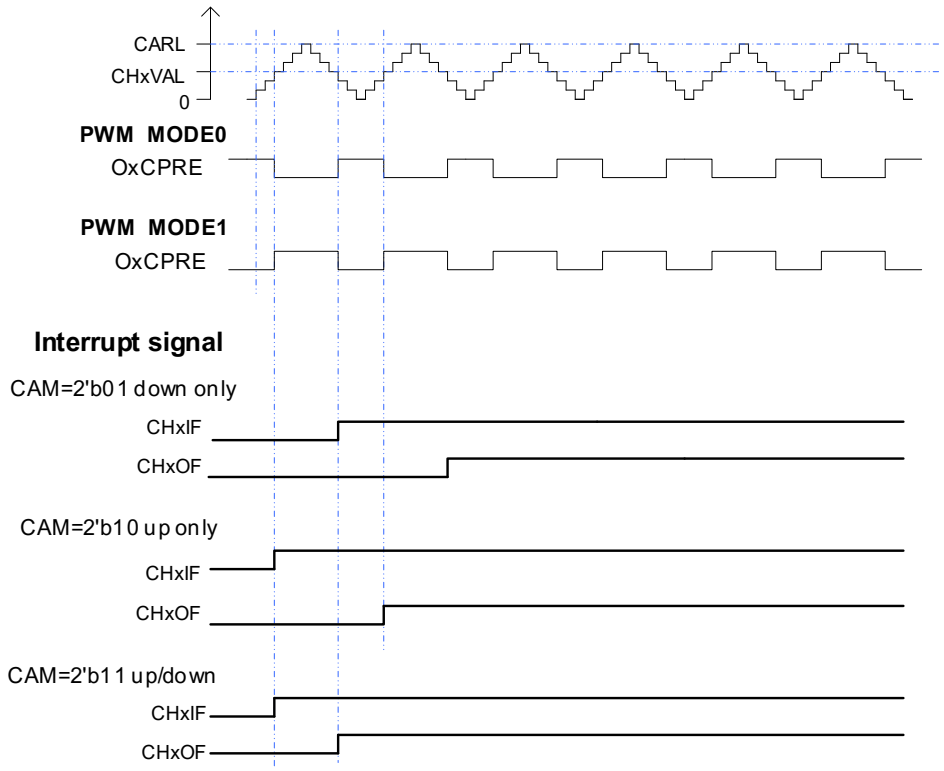


Figure 23-16. Timing diagram of CAPWM



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 23-2.The composite PWM pulse width***.

Table 23-2.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

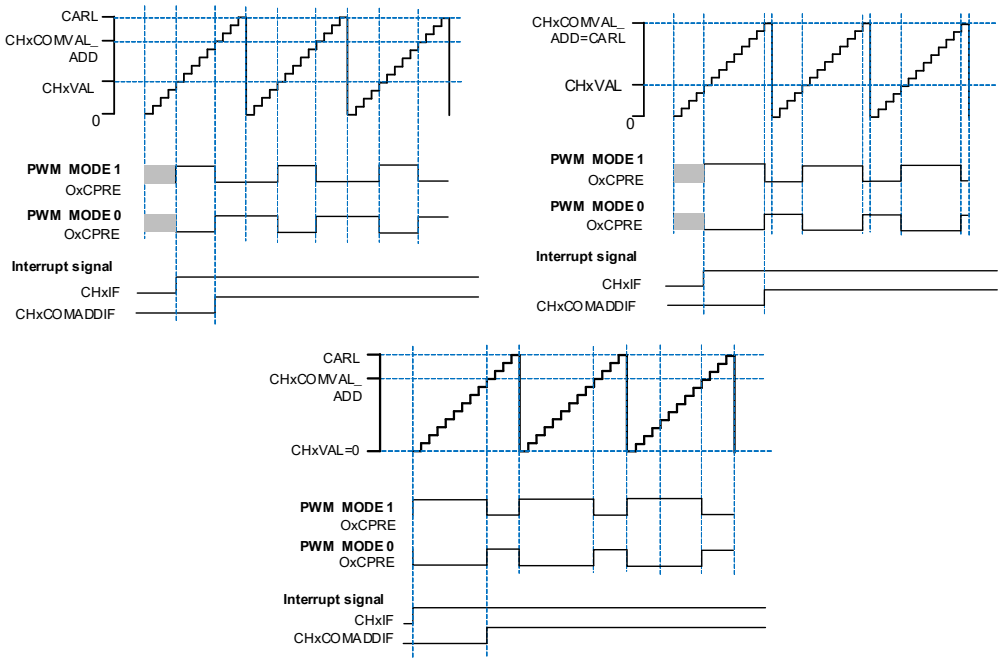
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when 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:

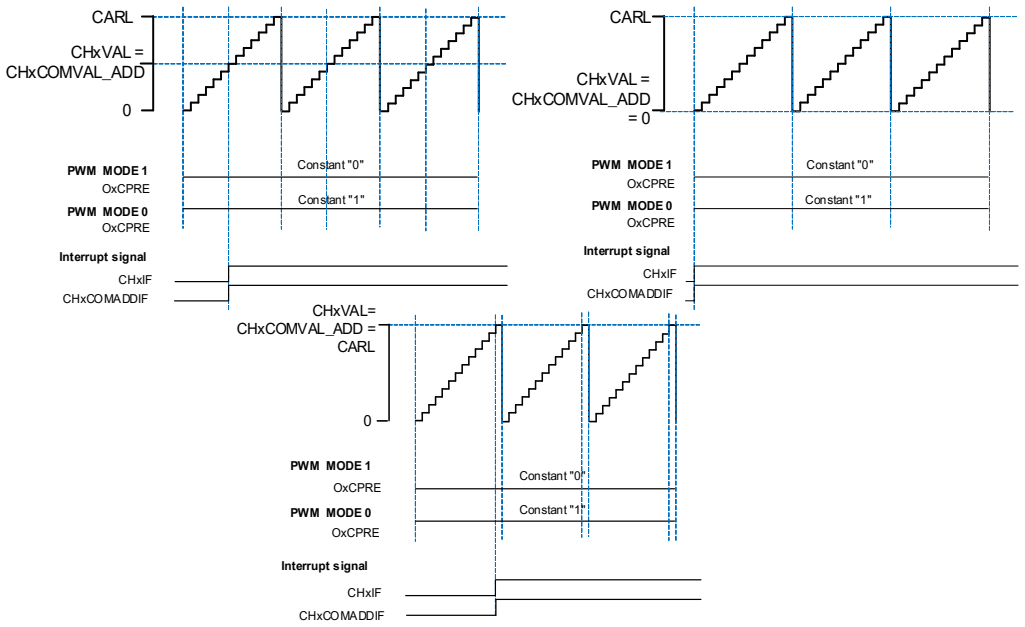
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-17. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



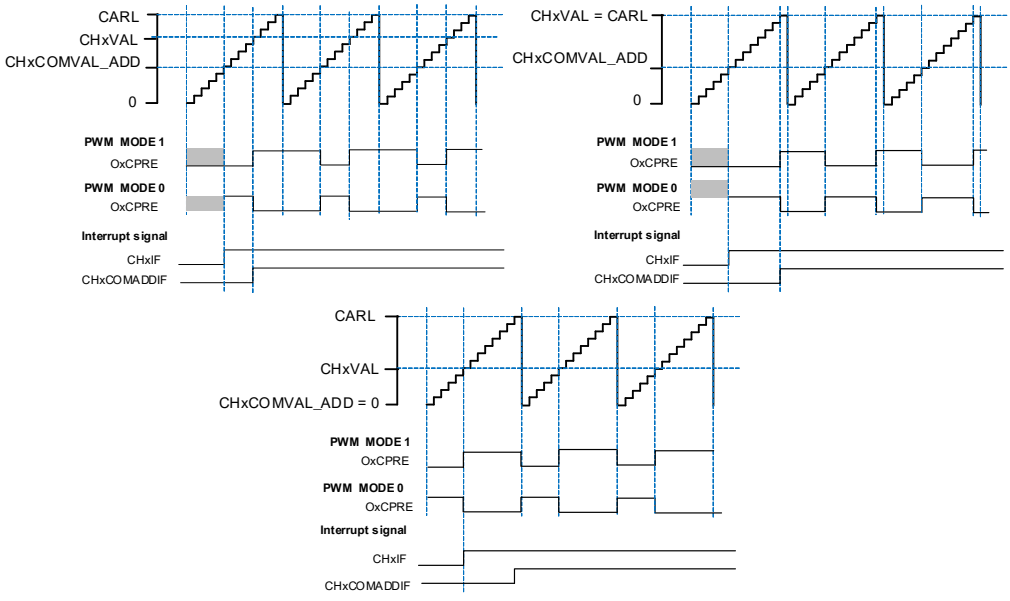
2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-18. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



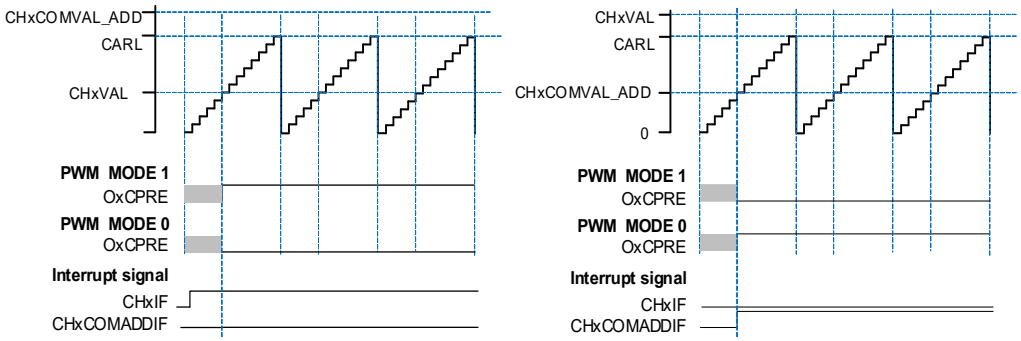
3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-19. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

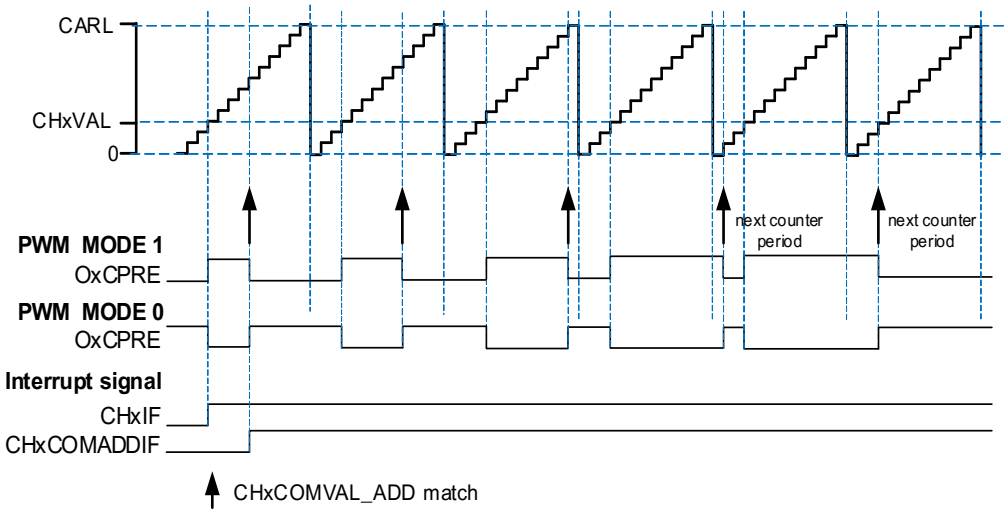
Figure 23-20. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 23-21. Channel x output PWM duty cycle changing with CHxCOMVAL_ADD* shows the PWM output and interrupts waveform.

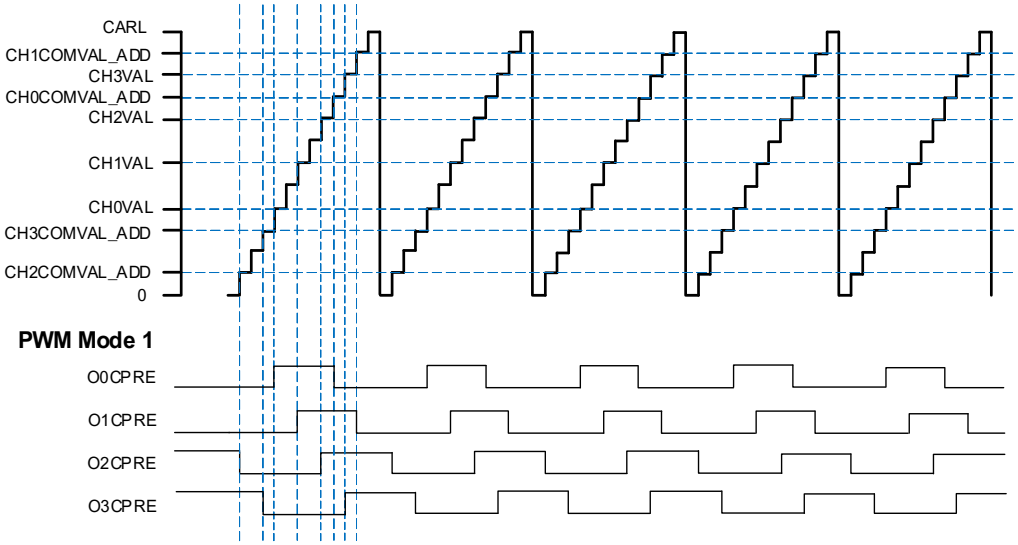
In some cases, the CHxCOMVAL_ADD match can happen on the next counter period (the value of CHxCOMVAL_ADD was written after the counter reaches the value of CHxVAL, and the value of CHxCOMVAL_ADD was less than or equal to the CHxVAL).

Figure 23-21. 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 23-22. 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 23-3.The asymmetric PWM pulse width*.

Table 23-3.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

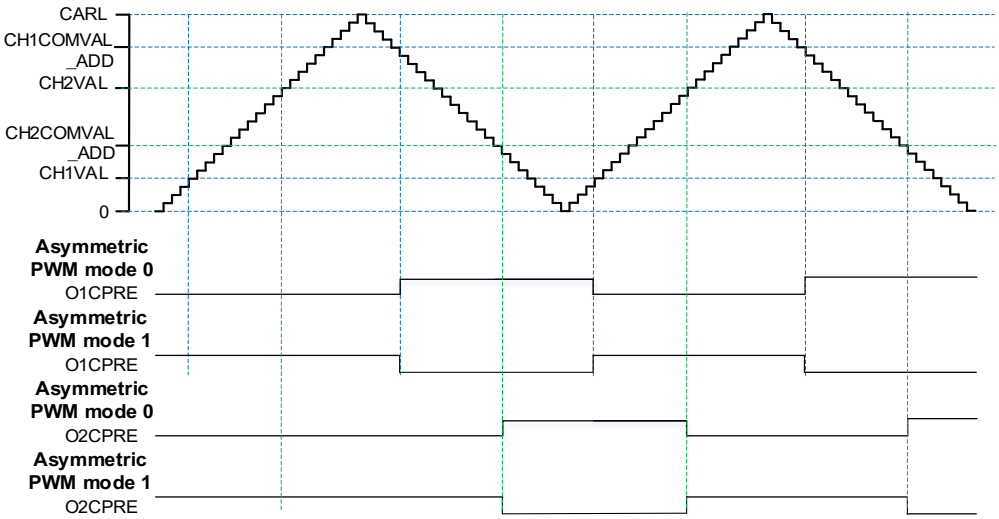
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in Composite PWM mode and Asymmetric 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:

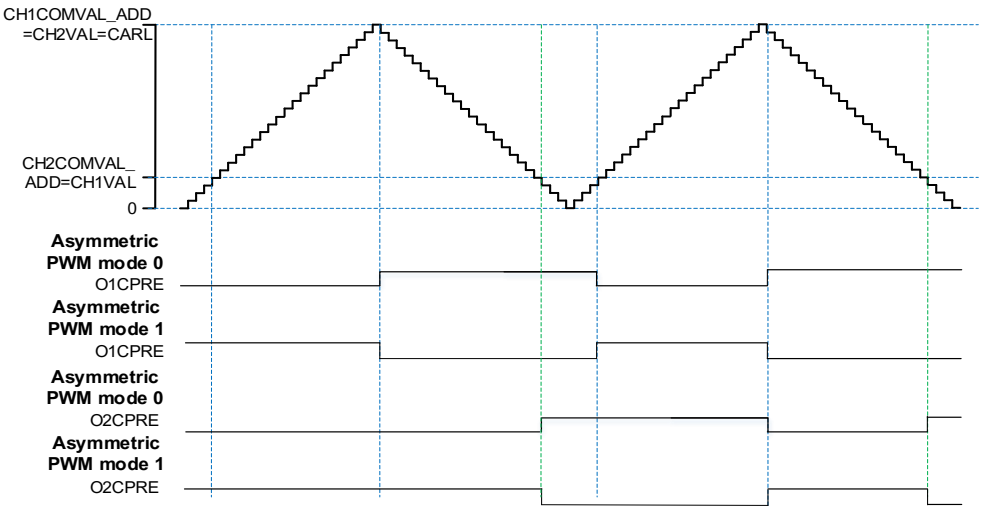
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-23. Channel x output PWM with(CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL)

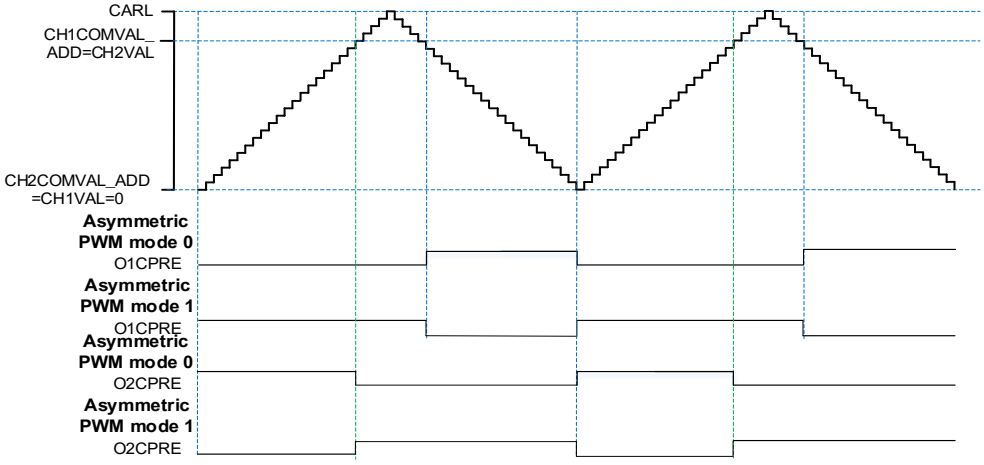
a. 0 < CH1VAL < CH1COMVAL_ADD < CARL / 0 < CH2COMVAL_ADD < CH2VAL < CARL



b. 0 < CH1VAL < CH1COMVAL_ADD = CARL / 0 < CH2COMVAL_ADD < CH2VAL = CARL, and CH1VAL = CH2COMVAL_ADD



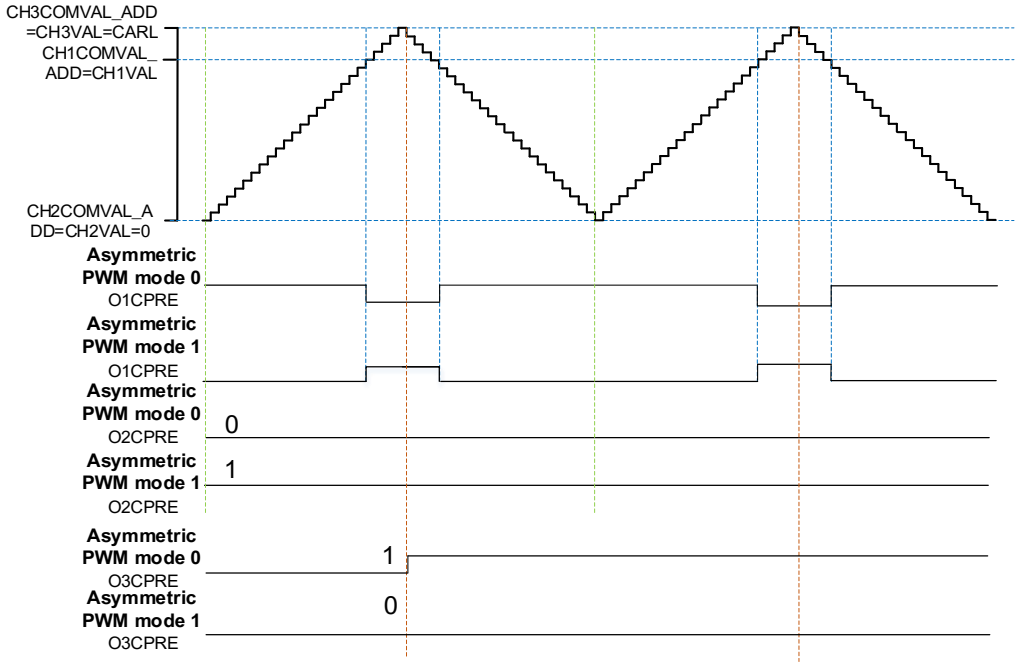
c. 0 = CH1VAL < CH1COMVAL_ADD < CARL / 0 = CH2COMVAL_ADD < CH2VAL < CARL, and CH1COMVAL_ADD = CH2VAL



2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-24. Channel x output PWM with(CHxVAL = CHxCOMVAL_ADD)

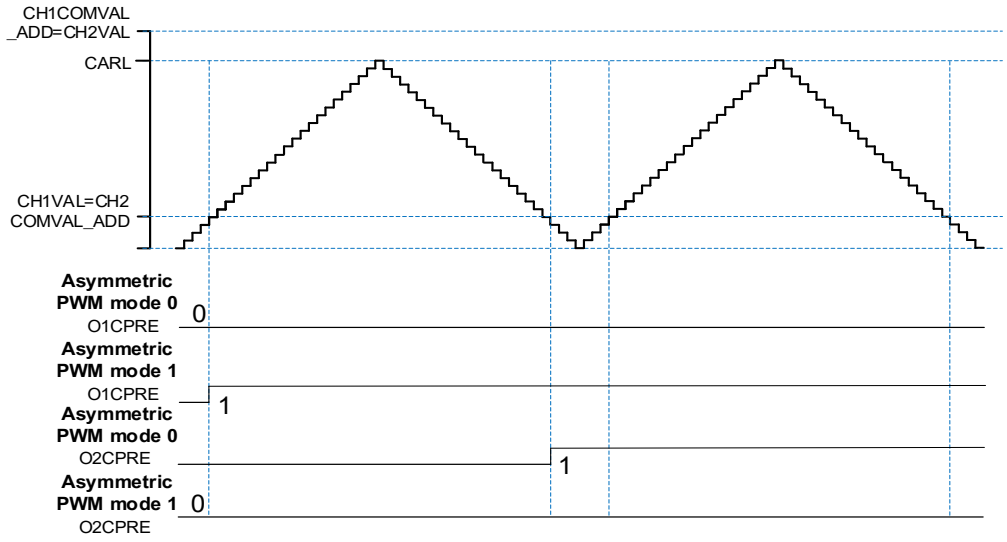
0 < CH1VAL = CH1COMVAL_ADD < CARL / CH2VAL = CH2COMVAL_ADD = 0 / CH3VAL = CH3COMVAL_ADD = CARL



3) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

Figure 23-25. 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 23-26. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00)

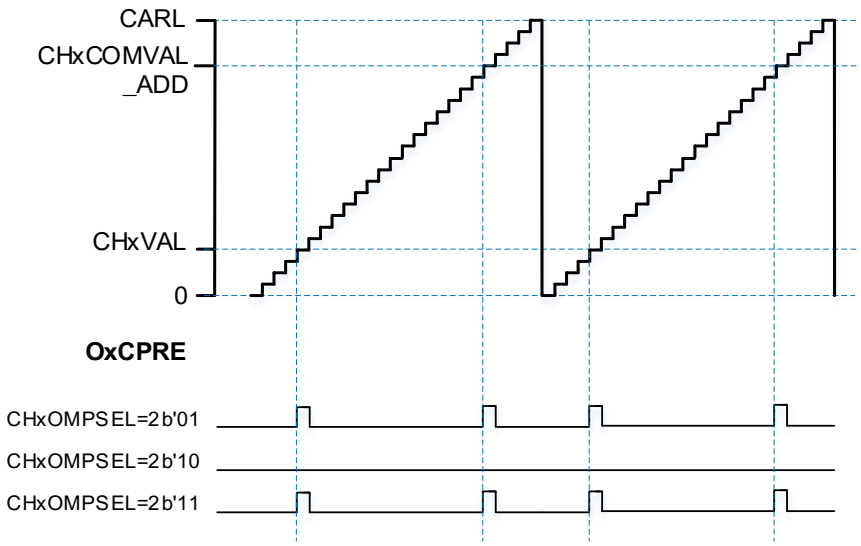
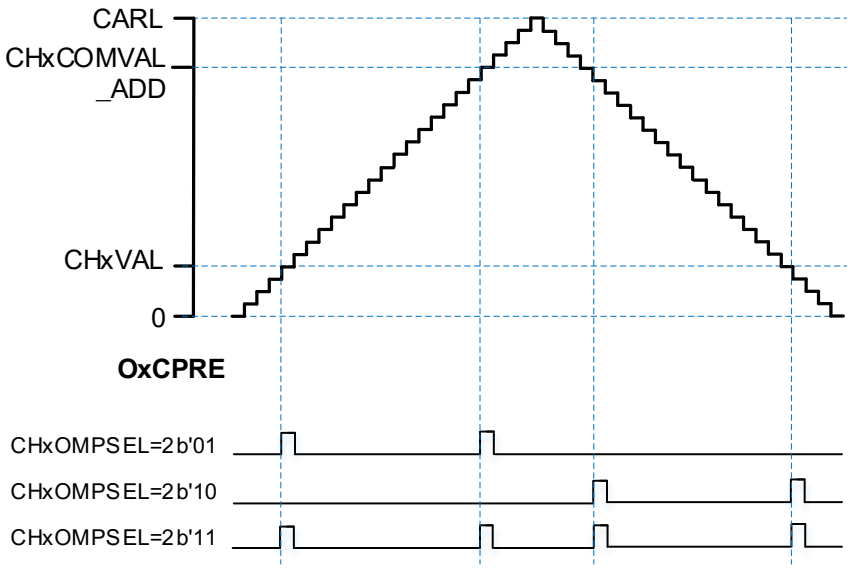


Figure 23-27. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00)



Channel output prepare signal

As is shown in *Figure 23-13. 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 23-4. Complementary outputs controlled by parameters***.

Table 23-4. 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

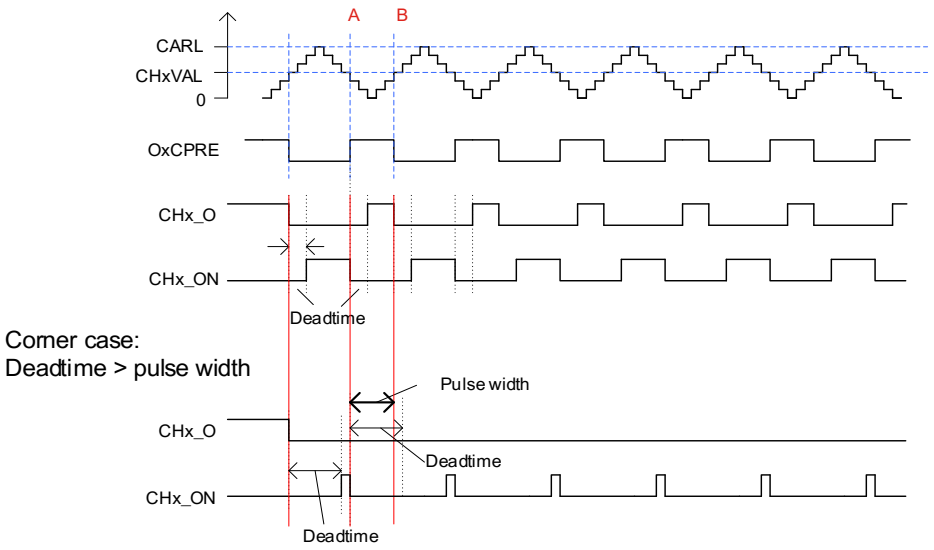
The dead time insertion is enabled when both CHxEN and CHxNEN are configured to 1'b1, it is also necessary to configure POEN&CHPOENx(x=0...2) to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the *Complementary channel* protection register 0 (TIMERx_CCHP0) 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 23-28. 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 23-28. Complementary output with dead time insertion***).

Figure 23-28. 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.

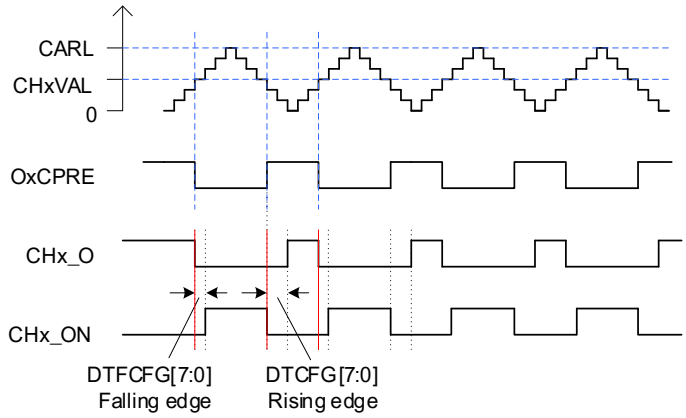
Different dead time insertion

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 23-29. 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 23-29. 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.

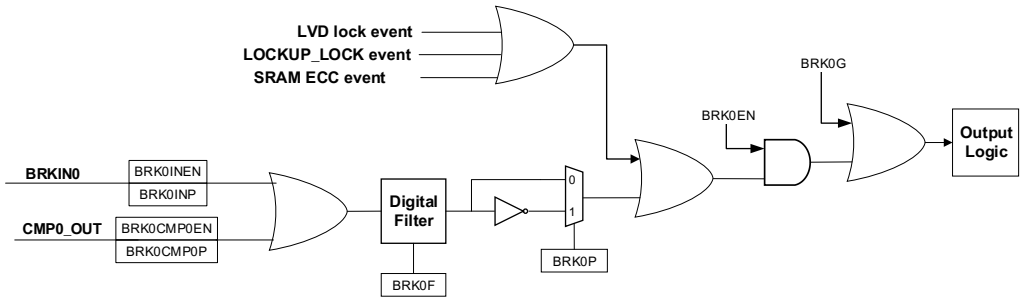
The break event is the result of logic ORed of all sources. The break functions can handle two types of event sources:

- 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 BRK0G bit in the TIMERx_SWEVG register.

Refer to ***Figure 23-30. BREAK0 function logic diagram***, BRKIN0 can select GPIO pins from the TRIGSEL module.

Figure 23-30. BREAK0 function logic diagram



BREAK0 can be used to handle the faults of system sources, on-chip peripheral events and external sources. When a BREAK0 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 23-31. Output behavior of the channel in response to BREAK0 (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 23-31. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1)

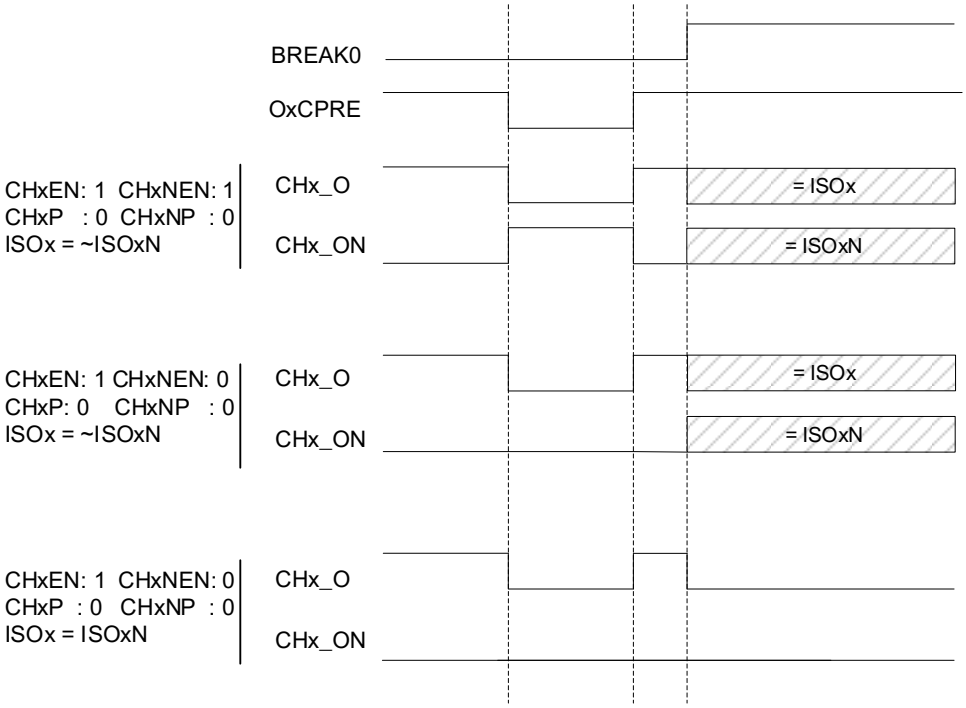


Table 23-5. Output behavior of the channel in response to a BREAK0 (the break input is high active)

BREAK0 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 BRK0IF 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

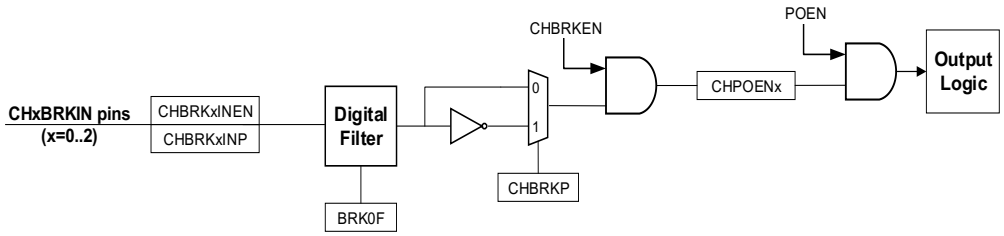
BEKENCHx(x=0...3) bit is "0" and a break event occurs, the corresponding channels CHx_O and CHx_ON will not be changed and the outputs is keeping.

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 BREAK0 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 BRK0F[3:0] with BREAK0. Please refer to ***Figure 23-32. 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 23-32. 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 23-6. 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 23-6. 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 23-33. Example of counter operation in quadrature decoder interface mode

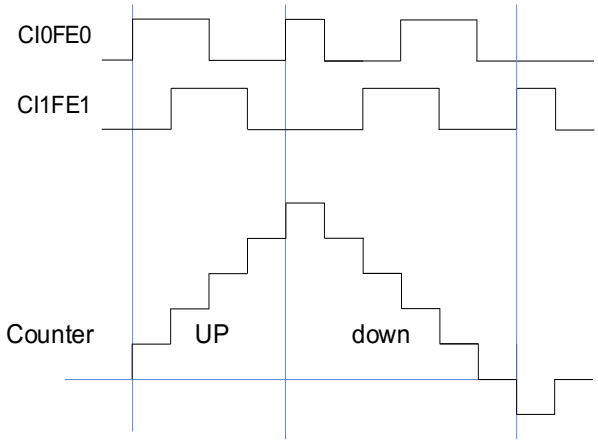
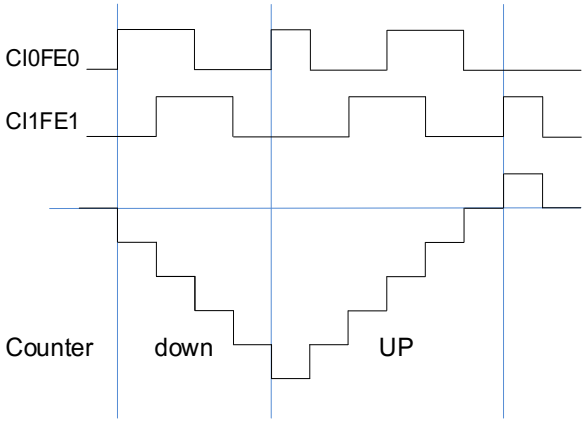


Figure 23-34. Example of quadrature decoder interface mode with CI0FE0 polarity inverted



Quadrature decoder and decoder clock output

TIMERs can output the decoder clock output signal as TRGO. This function can be enabled

by setting the MMC0[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 23-35. 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 23-35. Hall sensor is used for BLDC motor

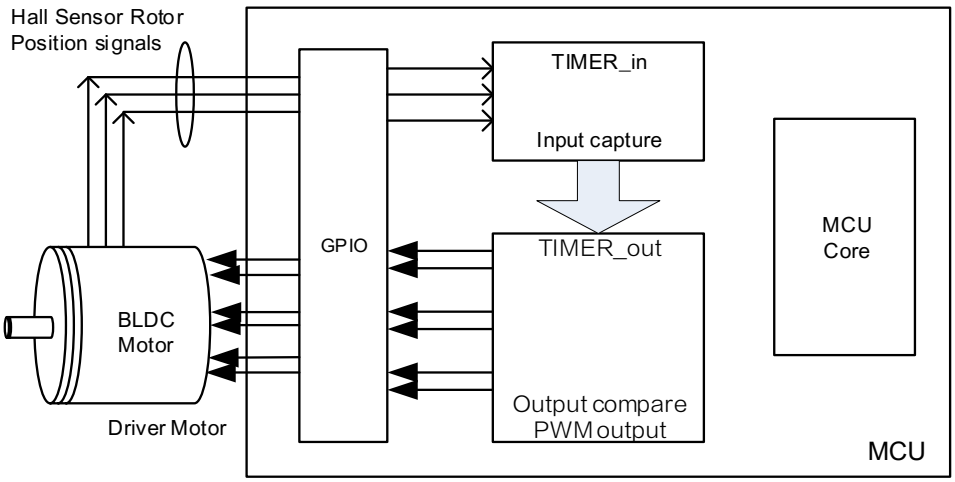
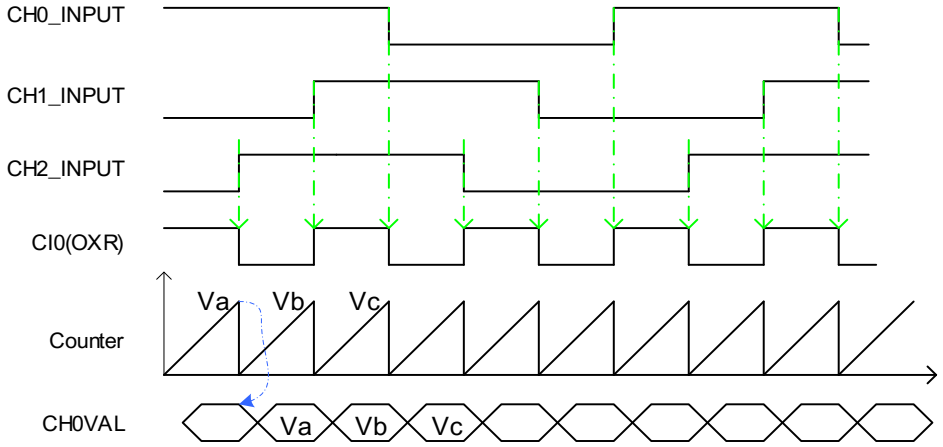
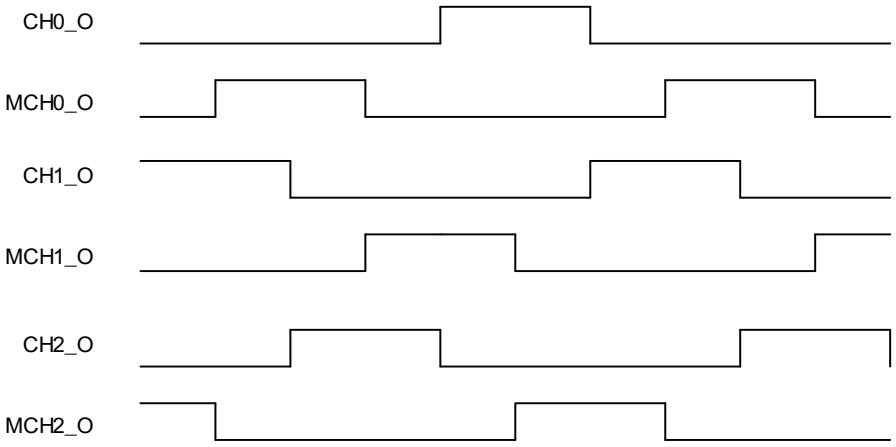


Figure 23-36. Hall sensor timing between two timers

Advanced/General L0 TIMER_in under input capture mode



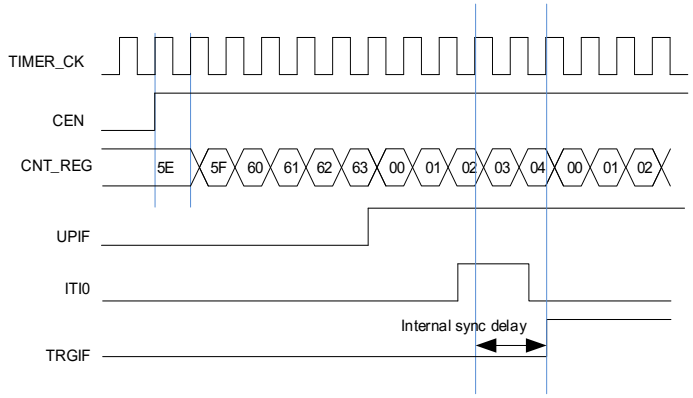
Advanced TIMER_out under output compare mode(PWM with Dead-time)



Master-slave management

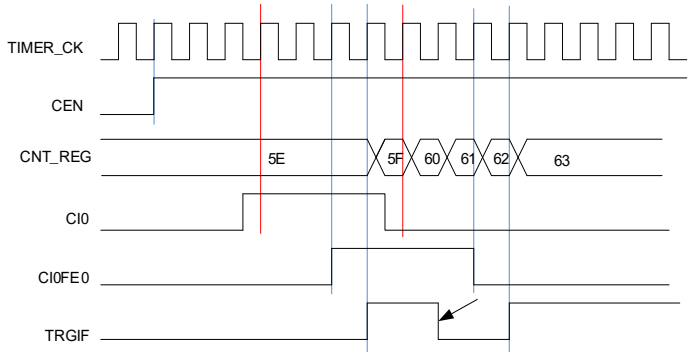
The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode and so on, which is selected by the TSCFGy[4:0] (y=3..6) in SYSCFG_TIMERxCFG(x=0,7).

Table 23-7. 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 00110: CI0FE0 00111: CI1FE1 01000: ETIFP(1) 01001: CI2FE2 01010: CI3FE3	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 output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	For the ITIx, no filter and prescaler can be used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of trigger input comes.	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 23-37. Restart mode  			
Exam2	**Pause mode** The counter will be paused when the trigger	TSCFG4[4:0] =5'b00110, CI0FE0 is selected.	TI0S=0 (Non-xor) [CH0NP=0, CH0P=0] CI0FE0 does not	Filter is bypassed in this example.

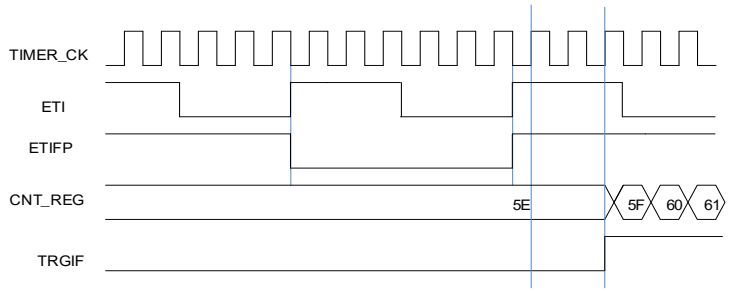
	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	input is low, and it will start when the trigger input is high.		invert. The capture event will occur on the rising edge only.	

Figure 23-38. 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 23-39. 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.

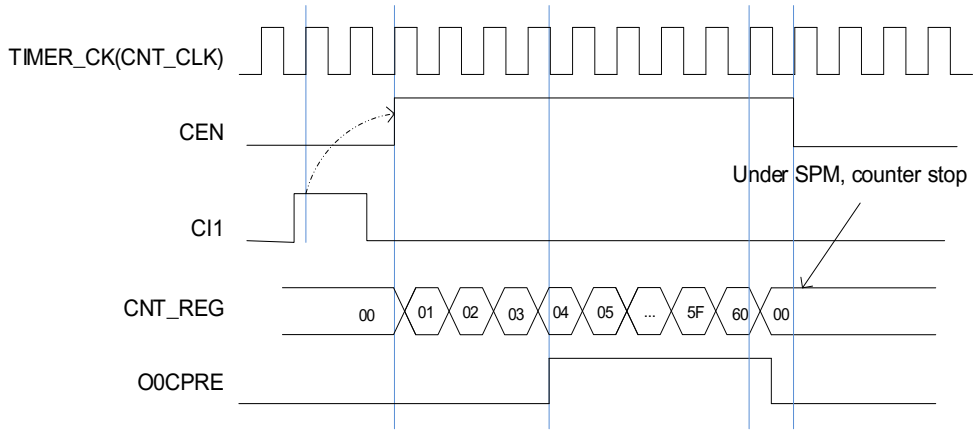
Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs 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 23-40. 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 (MMC0=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

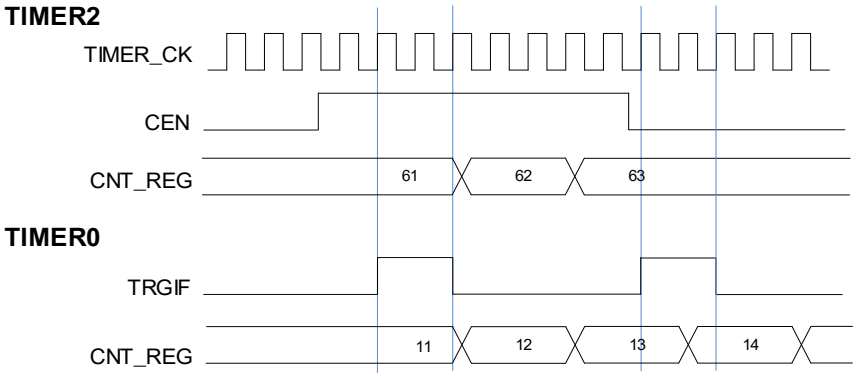
First, enable TIMER0 with the enable signal of TIMER2. Refer to ***Figure 23-41. Trigger mode of TIMER0 controlled by enable signal of TIMER2***. TIMER0 starts counting from

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 (MMC0=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).

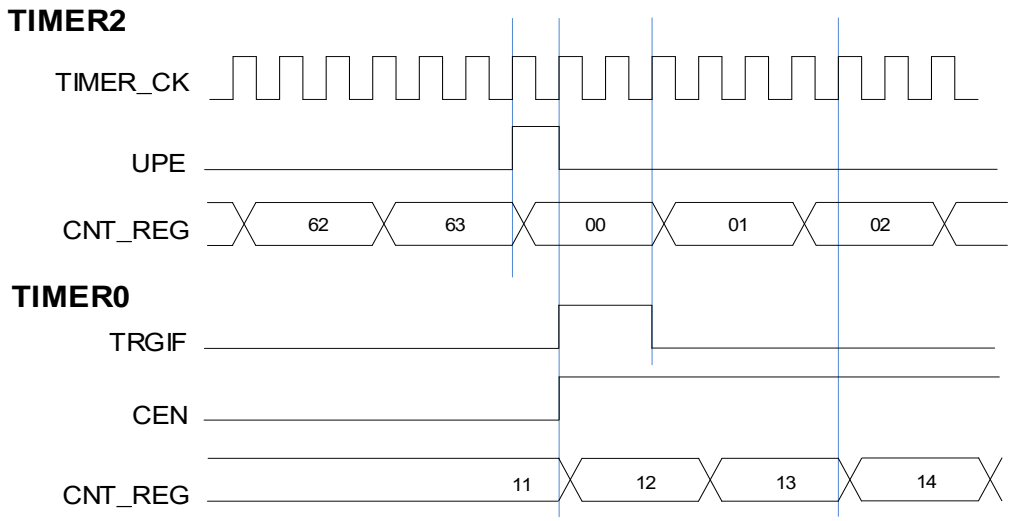
Figure 23-41. Trigger mode of TIMER0 controlled by enable signal of TIMER2



In this example, the update event can also be used as trigger source instead of enable signal. Refer to ***Figure 23-42. 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 (MMC0=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 23-42. Trigger mode of TIMER0 controlled by update signal of TIMER2

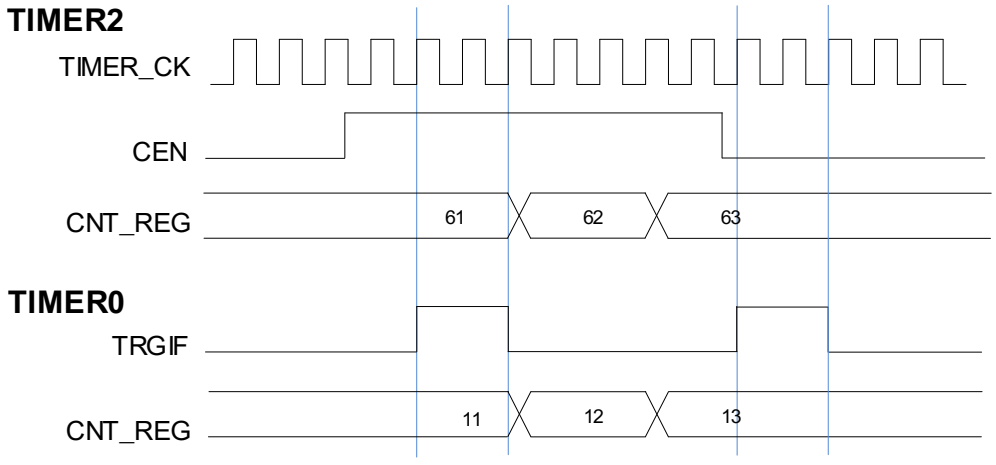


■ Enable TIMER0 to count with the enable/O0CPRE signal of TIMER2.

In this example, TIMER0 is enabled with the enable signal of TIMER2. Refer to ***Figure 23-43. 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 (MMC0=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 23-43. Pause mode of TIMER0 controlled by enable signal of TIMER2

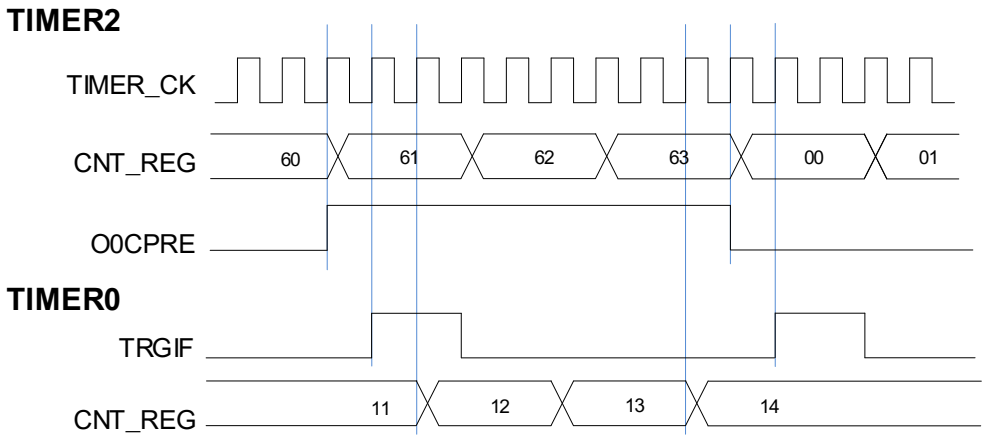


In this example, O0CPRE can also be used as trigger source instead of enable signal output.

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 23-44. Pause mode of TIMER0 controlled by O0CPRE signal of TIMER2



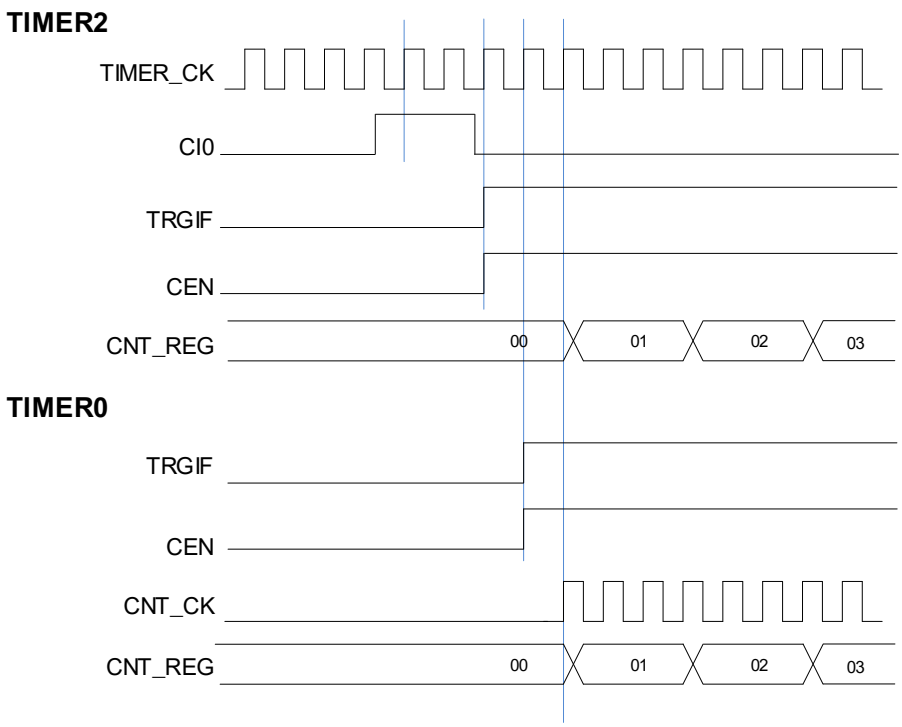
■ Using an external trigger to start two timers synchronously.

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 23-45. Trigger TIMER0 and TIMER2 by the CI0 signal of TIMER2



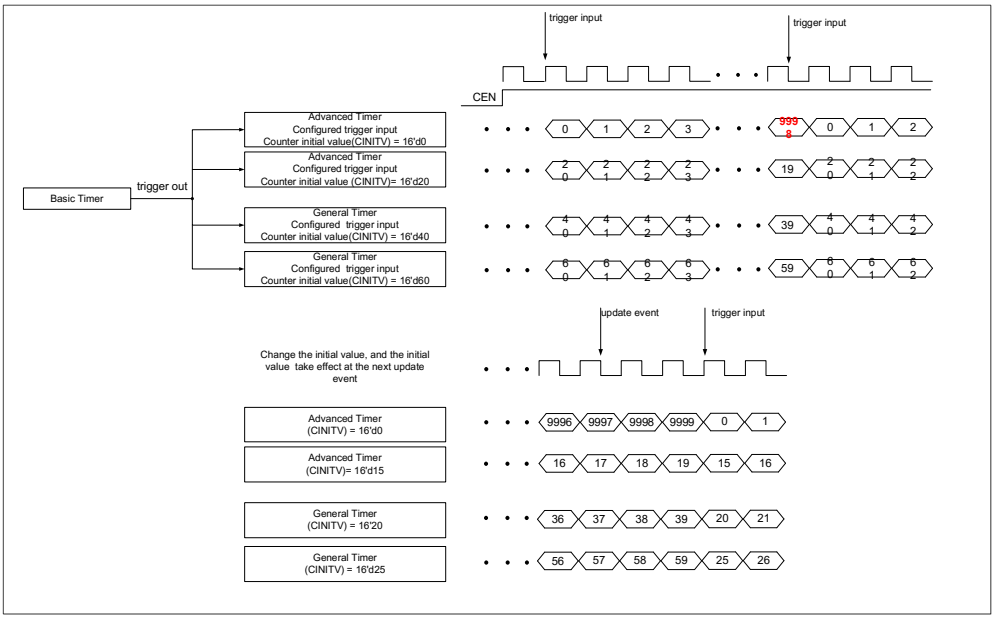
Counter synchronization and counter initial direction and value refresh

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.

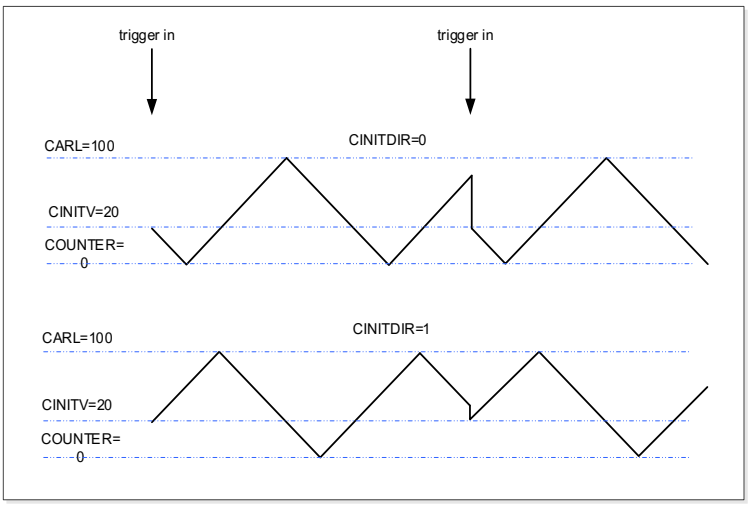
Figure 23-46. Triggering four timers by trigger out of basic timer



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 down after resetting the counter. When CINITDIR is 1, the count direction is up after resetting the counter.

Figure 23-47. 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 MMC0[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.

23.1.5. Registers definition (TIMERx, x=0, 7, 19)

TIMER0 base address: 0x4001 2C00

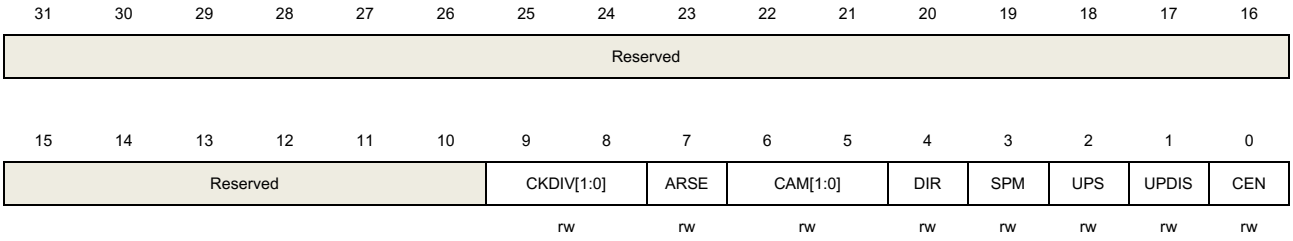
TIMER7 base address: 0x4001 3400

TIMER19 base address: 0x4001 4800

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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



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				MMC0[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	MMC0[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	MMC0[3]	Master mode control 0 Refer to MMC0[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	MMC0[2:0]	Master mode control 0 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	BRK0IF	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	BRK0IF	BREAK0 interrupt flag This flag is set by hardware when the BREAK0 input is active, and cleared by software if the BREAK0 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								BRK0G	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	BRK0G	BREAK0 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 BRK0IF flag will be set. 0: No generate a BREAK0 event 1: Generate a BREAK0 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

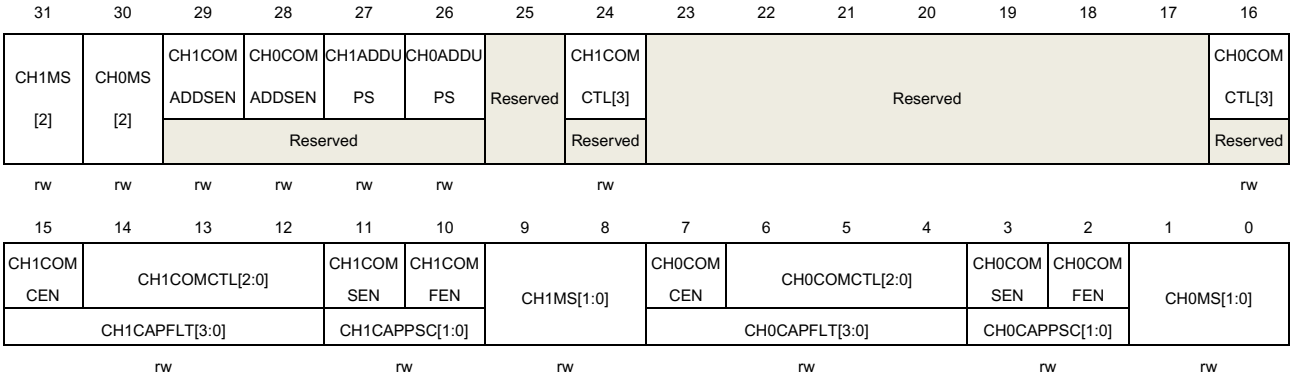
1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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



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 When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. 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

CH3COMCEN	CH3COMCTL[2:0]	CH3COMSEN	CH3COMFEN	CH3MS[1:0]	CH2COMCEN	CH2COMCTL[2:0]	CH2COMSEN	CH2COMFEN	CH2MS[1:0]
CH3CAPFLT[3:0]		CH3CAPPSC[1:0]			CH2CAPFLT[3:0]		CH2CAPPSC[1:0]		
rw		rw		rw	rw		rw		rw

Output compare mode:

Bits	Fields	Descriptions
31	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
When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM1 or PWM2 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the

		result of the comparison. 0: Channel 2 output quickly compare disable. 1: Channel 2 output quickly compare enable.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. 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															

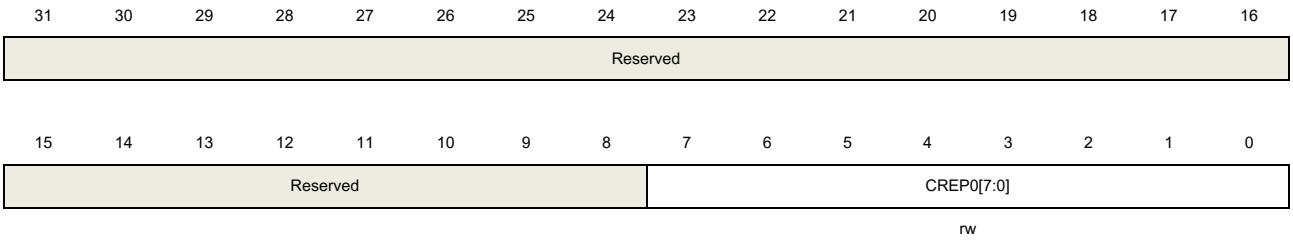
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-field specifies the auto reload value of the counter.

Counter repetition register 0 (TIMERx_CREP0)

Address offset: 0x30

Reset value: 0x0000 0000

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



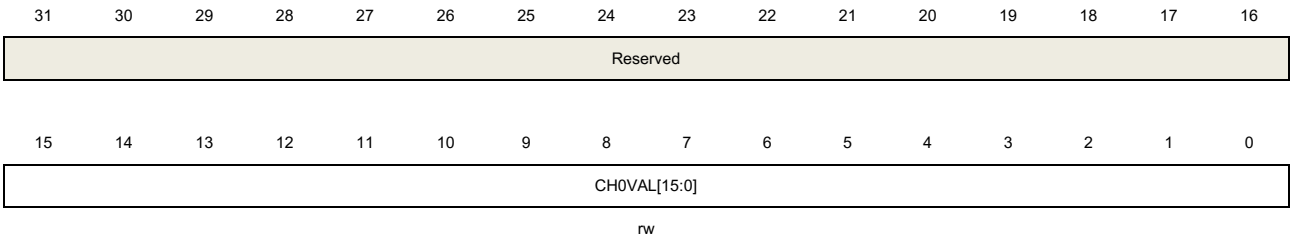
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP0[7:0]	Counter repetition value 0 This bit-field specifies the update event generation rate. Each time the repetition counter counts down to zero, an update event will be generated. The update rate of the shadow registers is 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.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C
Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

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

15:0	CH2VAL[15:0]	Capture/compare value of channel 2 When channel 2 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3VAL[15:0]	Capture/compare value of channel 3 When channel 3 is configured in input mode, this bit-field indicates the counter value at the last capture event. And this bit-field is read-only. When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event.

Complementary channel protection register 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										BRK0F[3:0]			
rw	rw											rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRK0P	BRK0EN	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 BRK0F[3:0]
BREAK0 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 BREAK0 input signal and the length of the digital filter applied to BREAK0.
0000: Filter disabled. BREAK0 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	BRK0P	BREAK0 input signal polarity This bit specifies the polarity of the BREAK0 input signal. 0: BREAK0 input active low 1: BREAK0 input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
12	BRK0EN	BREAK0 input signal enable This bit can be set to enable the BREAK0 input signal 0: BREAK0 input disabled 1: BREAK0 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 23-4. 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 23-4. 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 BRK0EN /BRK0P/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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode (when CH1CPWMEN=1).

Channel 2 additional compare value register (TIMERx_CH2COMV_ADD)

Address offset: 0x6C

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH2COMVAL_ADD [15:0]	Additional compare value of channel 2 When channel 2 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the

shadow register updates by every update event.

Note: This register just used in composite PWM mode (when CH2CPWMEN=1).

Channel 3 additional compare value register (TIMERx_CH3COMV_ADD)

Address offset: 0x70

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMVAL_ADD[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH3COMVAL_ADD [15:0]	Additional compare value of channel 3 When channel 3 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode (when CH3CPWMEN=1).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 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 PWMEN	CH2C PWMEN	CH1C PWMEN	CH0C PWMEN	Reserved											
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					BRK0CMP1P	BRK0CMP0P	Reserved								BRK0IN0P
					rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK0CMP1EN	BRK0CMP0EN	Reserved								BRK0IN0EN
					rw	rw									rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	BRK0CMP1P	BREAK0 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =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.
25	BRK0CMP0P	BREAK0 CMP0 input polarity This bit is used to configure the CMP0 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP0 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP0 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register

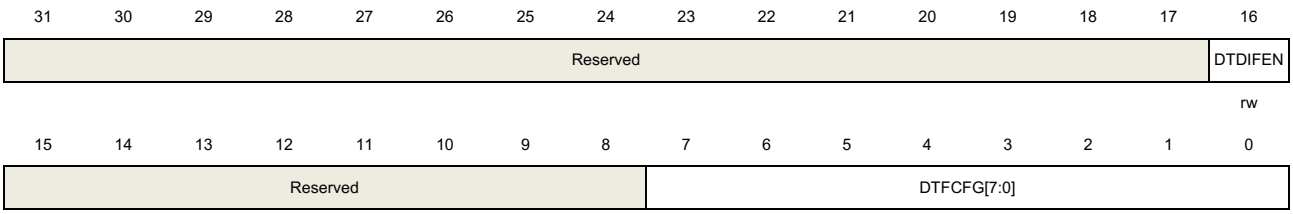
		is 00.
24:17	Reserved	Must be kept at reset value.
16	BRK0IN0P	BREAK0 BRKIN0 alternate function input polarity This bit is used to configure the BRKIN0 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: BRKIN0 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =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.
15:11	Reserved	Must be kept at reset value.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled 1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
9	BRK0CMP0EN	BREAK0 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	BRK0IN0EN	BREAK0 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).



rw

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

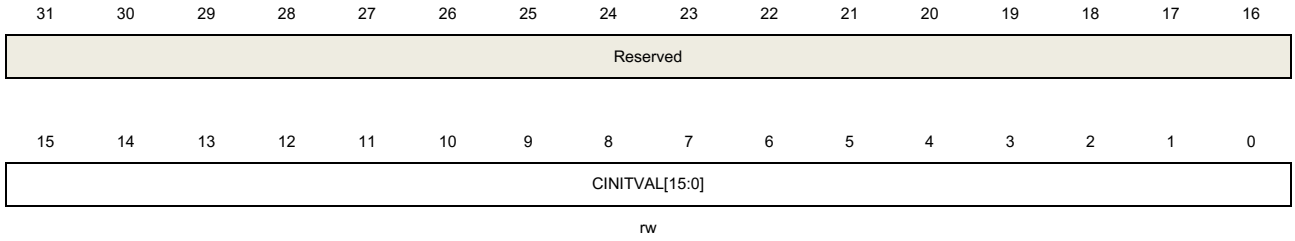
		always return a 0. 0: No affect 1: Generate soft synchronization event
1	CINITDIR	Counter initial count direction 0: When the synchronization event occurs, the counter count down. 1: When the synchronization event occurs, the counter count up. This bit indicates the initial direction of the counter after 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).



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).

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

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.

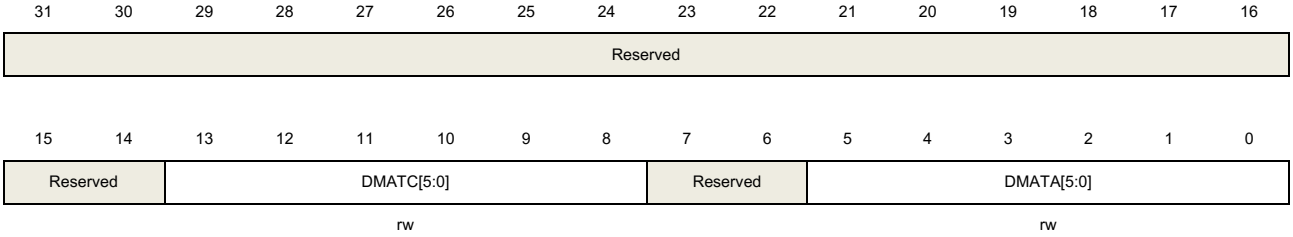
Note: When CH0BRKMIE =1 and BRKIE =1, if this flag is set, an interrupt is generated.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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



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 … 6'b111111: TIMERx_DMACFG 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.

23.2. General level0 timer (TIMERx, x=1,2,3,4)

23.2.1. 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.

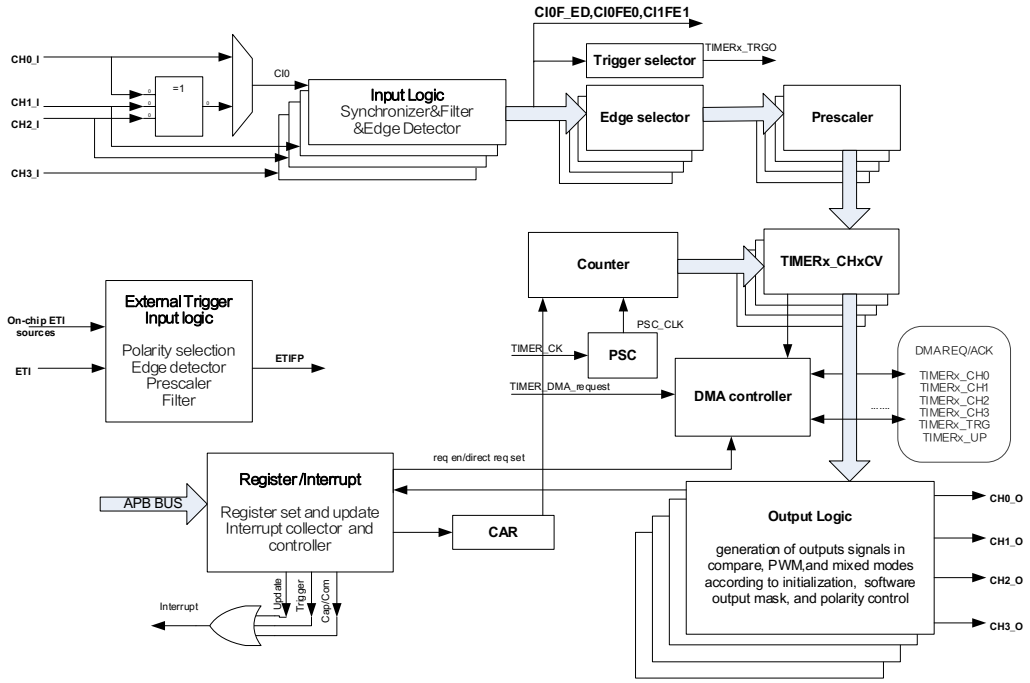
23.2.2. Characteristics

- 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.

23.2.3. Block diagram

*Figure **23-48. General Level 0 timer block diagram*** provides details on the internal configuration of the general level 0 timer.

Figure 23-48. General Level 0 timer block diagram



23.2.4. 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.

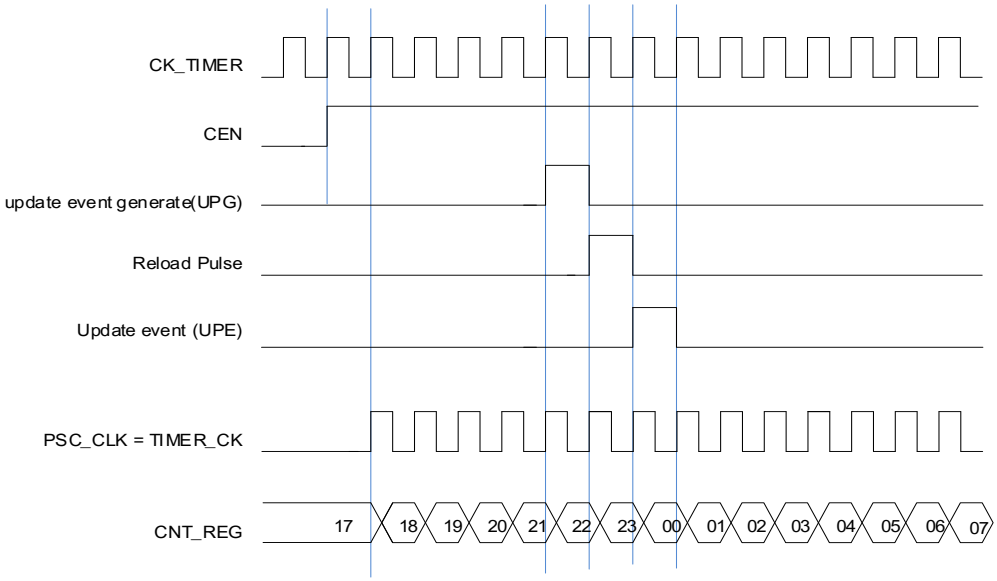
- TSCFGy[4:0] (y=0..6,9..15) in SYSCFG_TIMERxCFG(x=1..4) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

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 23-49. 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.

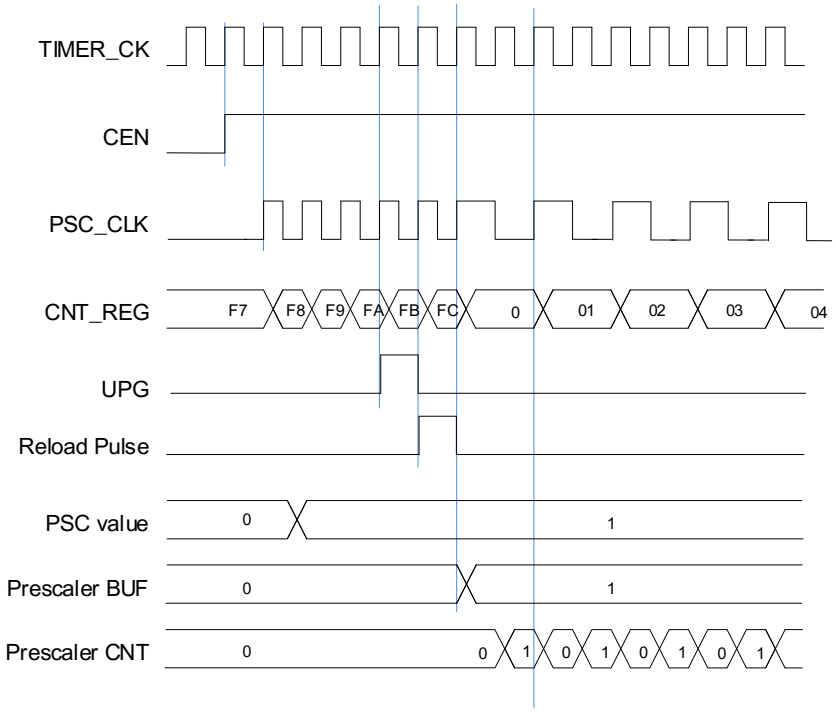
■ SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

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 23-50. 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 23-51. Timing chart of up counting mode, PSC=0/2 and *Figure 23-52. 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 23-51. Timing chart of up counting mode, PSC=0/2

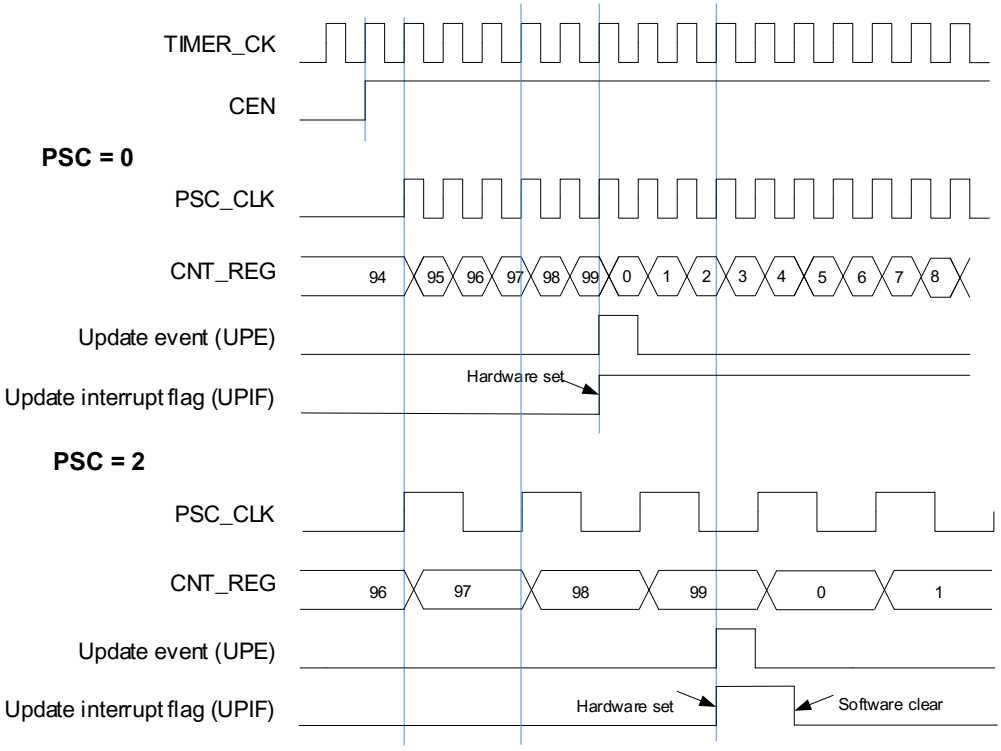
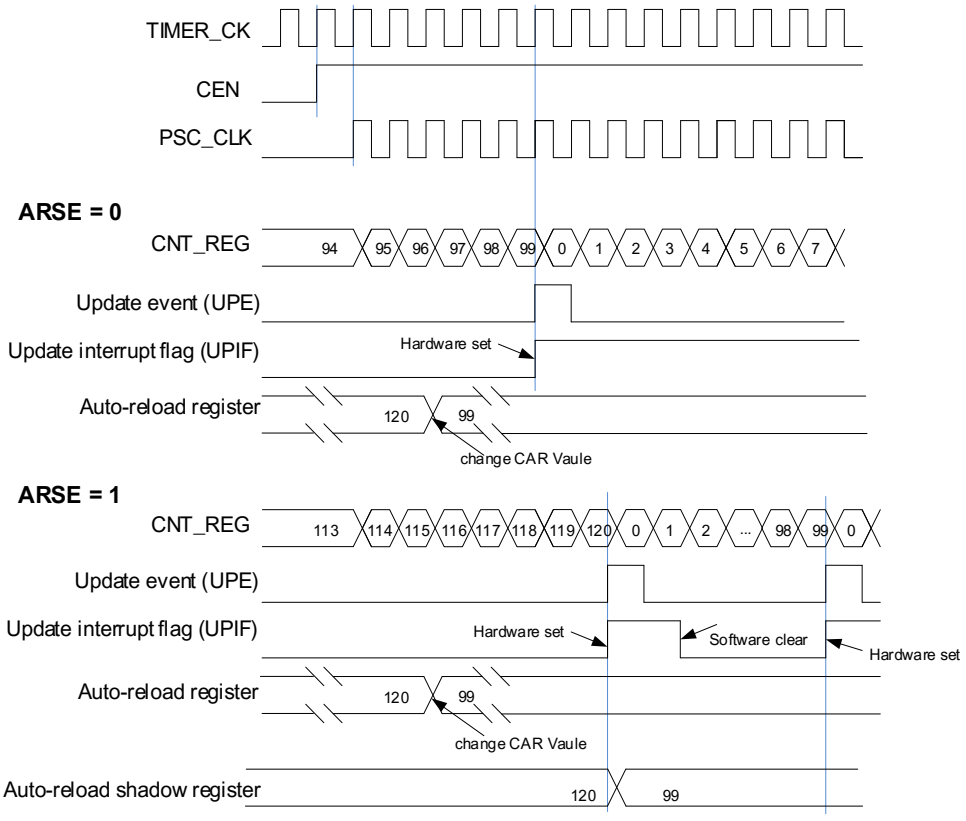


Figure 23-52. Timing chart of up counting, change TIMERx_CAR ongoing



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 23-53. Timing chart of down counting mode, PSC=0/2 and *Figure 23-54. 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 23-53. Timing chart of down counting mode, PSC=0/2

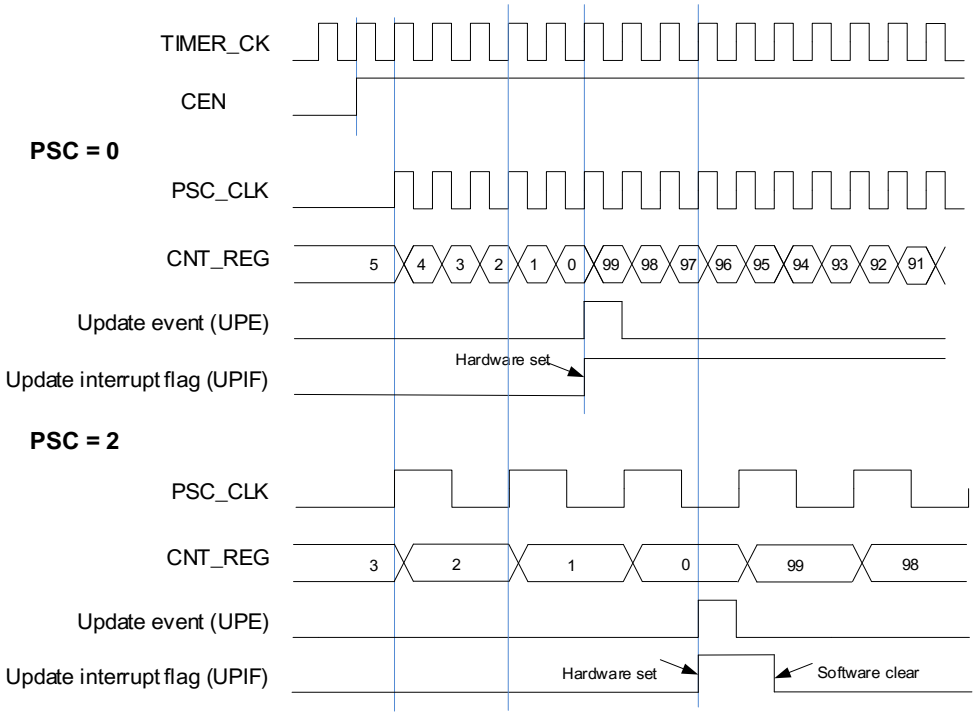
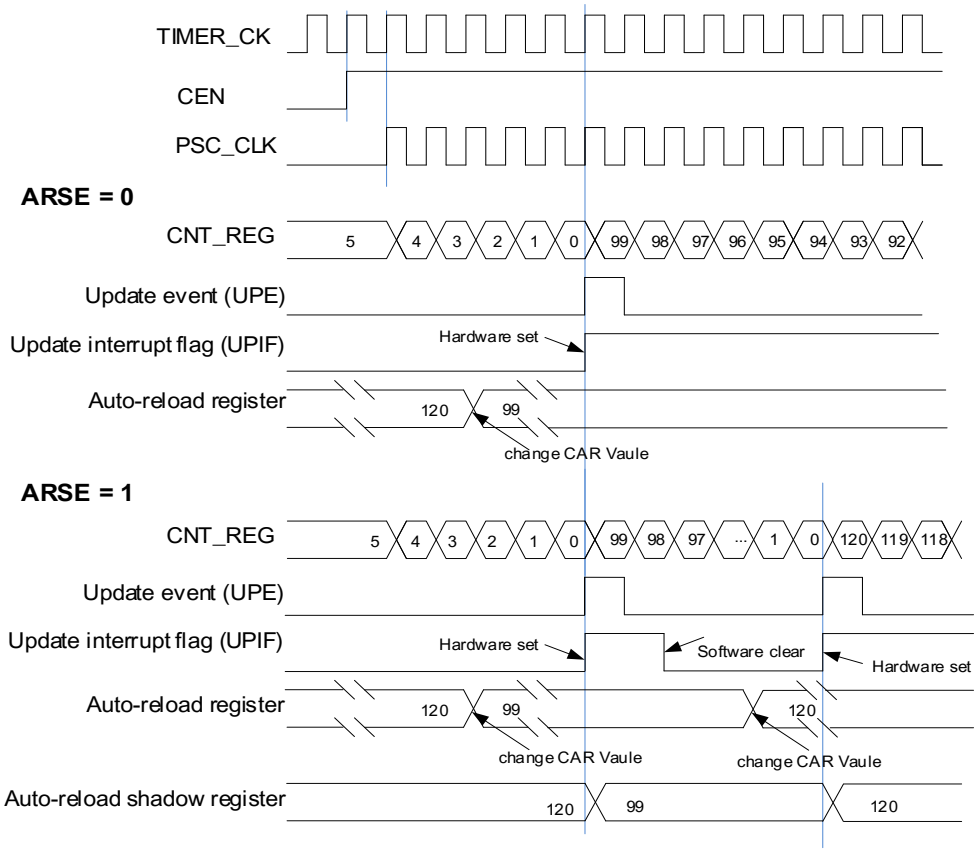


Figure 23-54. Timing chart of down counting mode, change TIMERx_CAR ongoing



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.

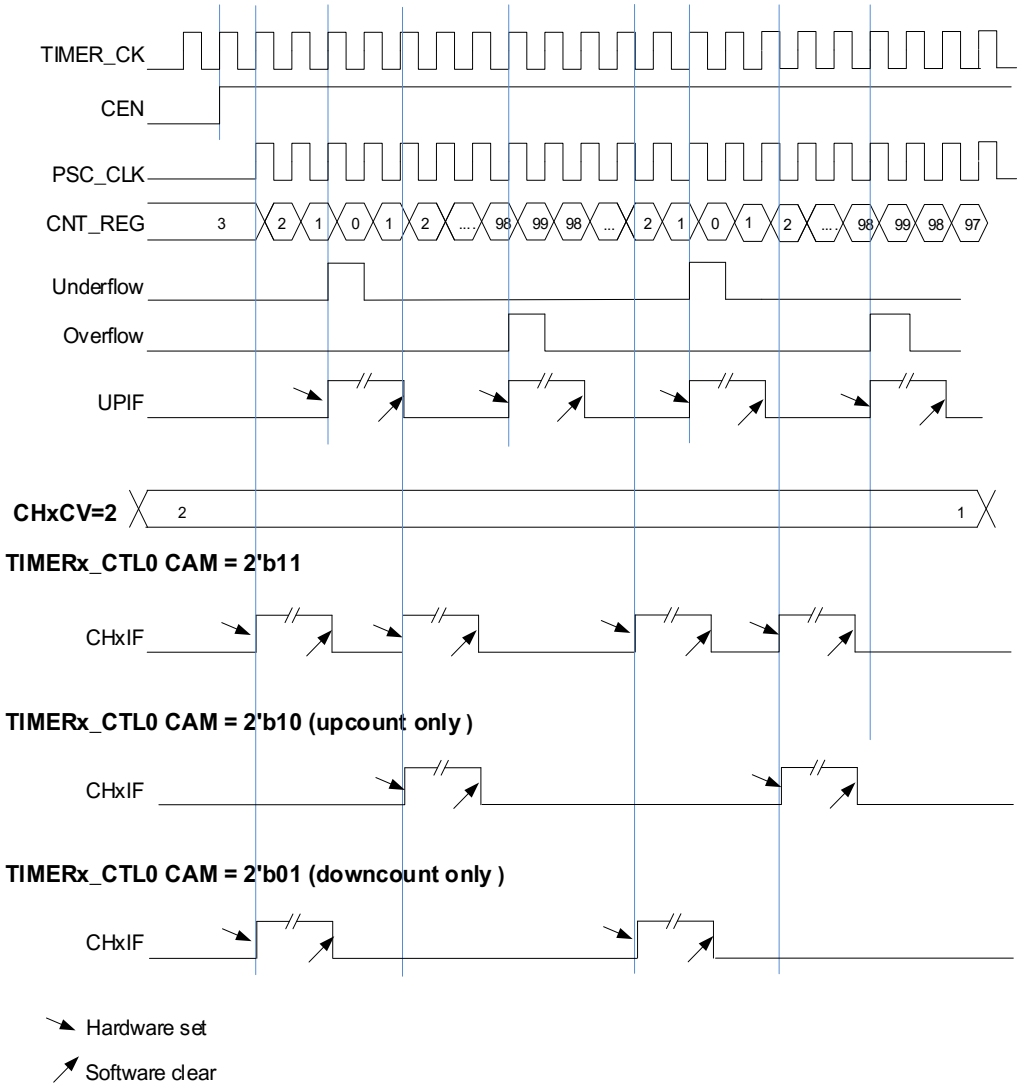
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 23-55. Timing chart of center-aligned counting mode.***

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 23-55. Timing chart of center-aligned counting mode* shows the example of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0.

Figure 23-55. Timing chart of center-aligned counting mode



Capture/compare channels

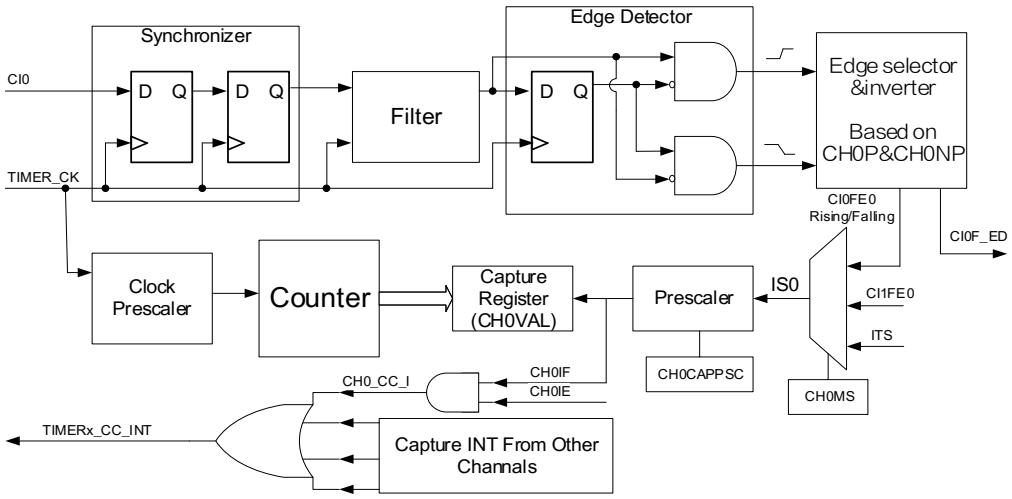
The general level0 timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

■ 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 23-56. 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 23-57. Output compare logic (x=0,1,2,3)

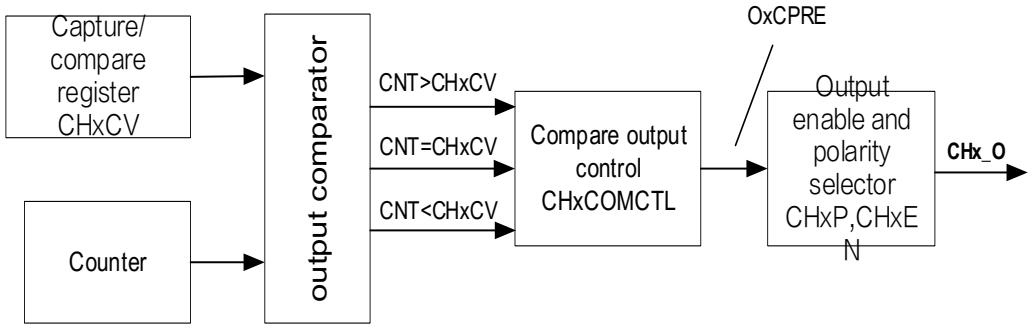


Figure 23-57. Output compare logic (x=0,1,2,3) shows the logic circuit of output compare mode. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output* prepare signal) is described as blew: The active level of O0CPRE is high, the output level 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.

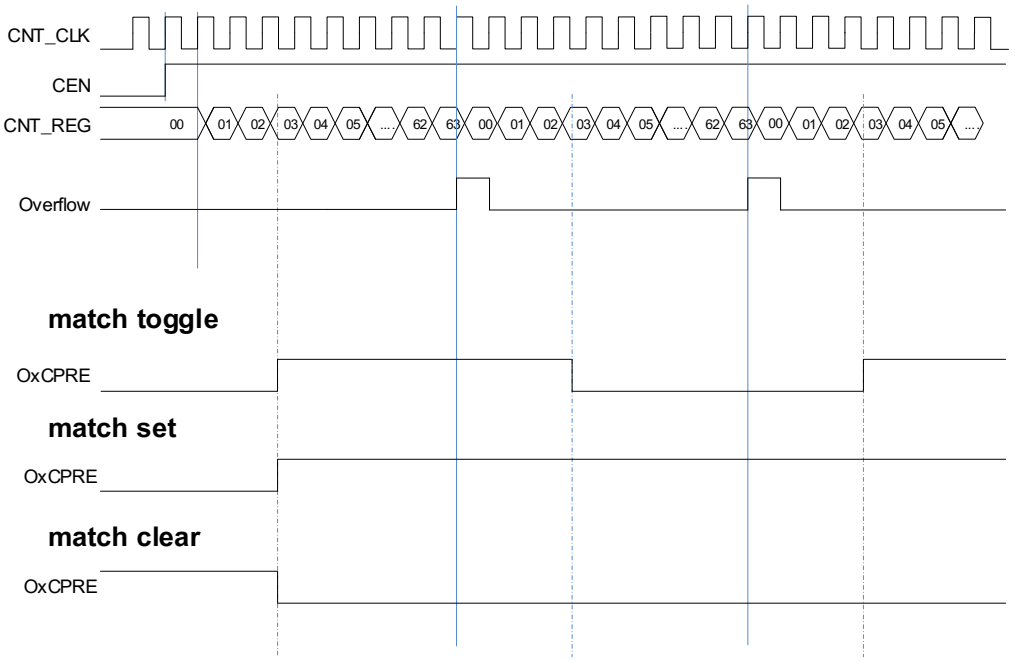
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.

The TIMERx_CHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 23-58. Output-compare under three modes shows the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 23-58. 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 23-59. 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 23-60. 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 23-59. Timing chart of EAPWM

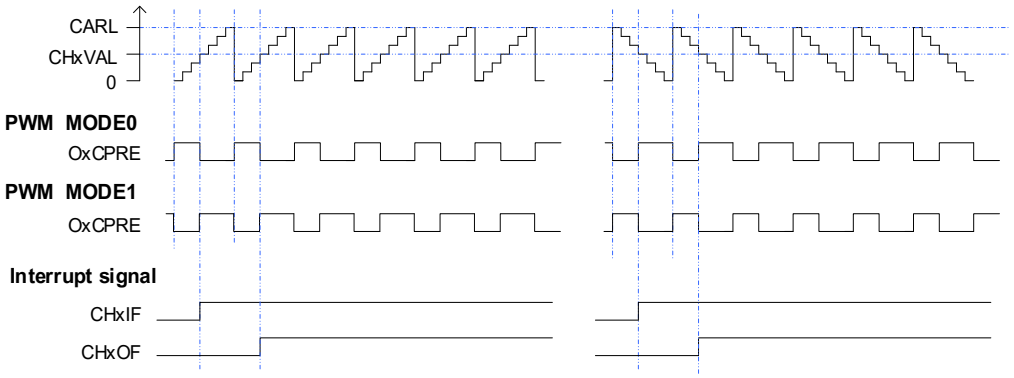
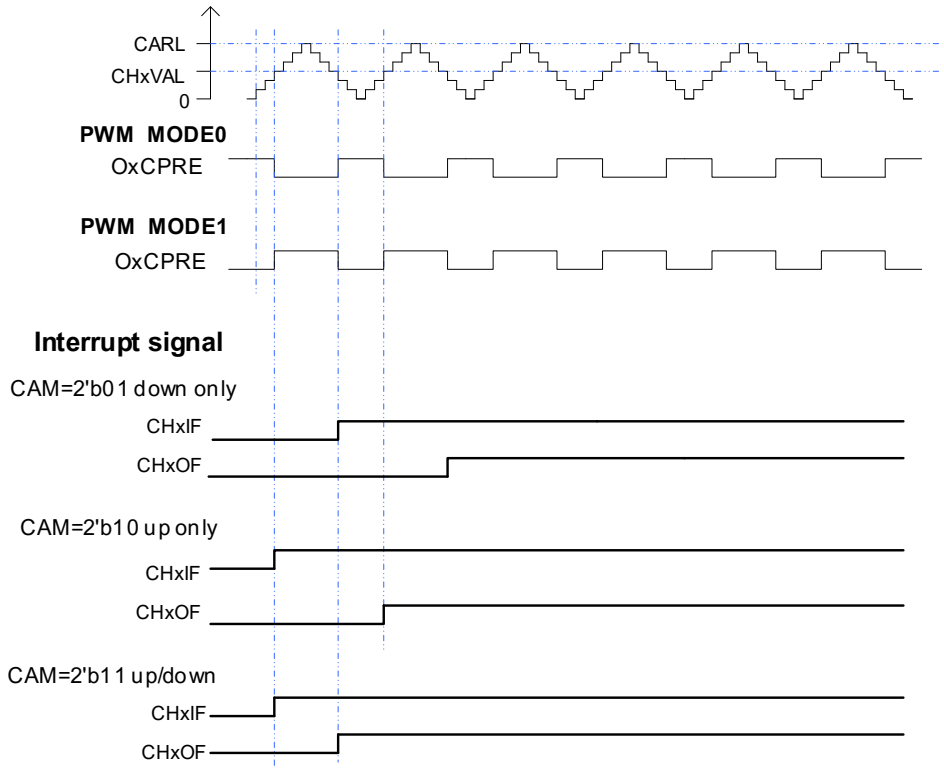


Figure 23-60. Timing chart of CAPWM



Channel output prepare signal

As is shown in *Figure 23-57. 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 23-8. 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 23-8. Counting direction in different quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0	CI1FE1=1	Down	Up	-	-
TSCFG0[4:0]!= 5'b00000	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1	CI0FE0=1	-	-	Up	Down
TSCFG1[4:0]!= 5'b00000	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
TSCFG2[4:0]!= 5'b00000	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up
Quadrature decoder mode 3	CI1FE0=1	Down	Up	-	-
TSCFG13[4:0]!= 5'b00000	CI1FE0=0	-	-	-	-
Quadrature decoder mode 4	CI0FE1=1	-	-	Up	Down
TSCFG14[4:0]!= 5'b00000	CI0FE1=0	-	-	-	-

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 23-61. Example of counter operation in decoder interface mode

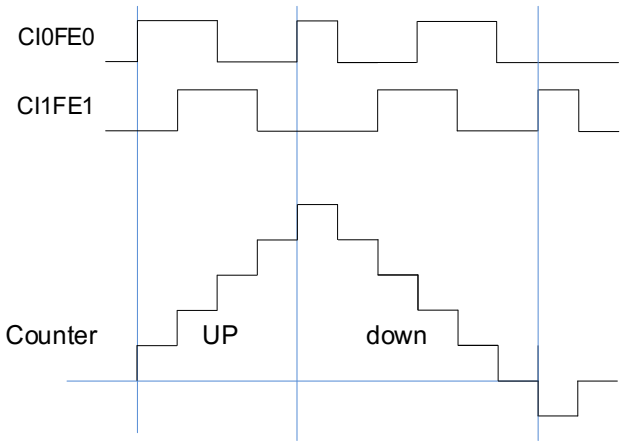
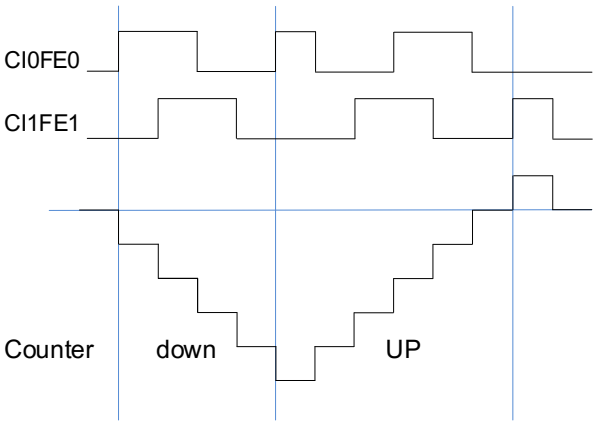


Figure 23-62. Example of decoder interface mode with CI0FE0 polarity inverted



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.

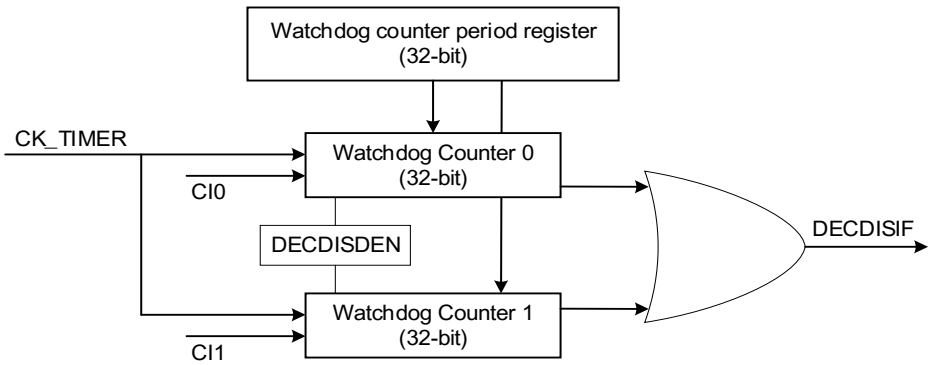
When DECJDEN=1, if the level transitions of the two quadrature signals CI0 and CI1 occur simultaneously, the interrupt flag DECJIF is set, if DECJIE=1, the corresponding interrupt is generated.

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 23-63. 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 23-63. Quadrature decoder signal disconnection detection block diagram



Decoder

The decoder function has four modes: 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 decoder mode 0 and decoder mode 1 are enabled, the CI0FE0 is used as the count

direction signal and the CI1FE1 signal is used as the count pulse.

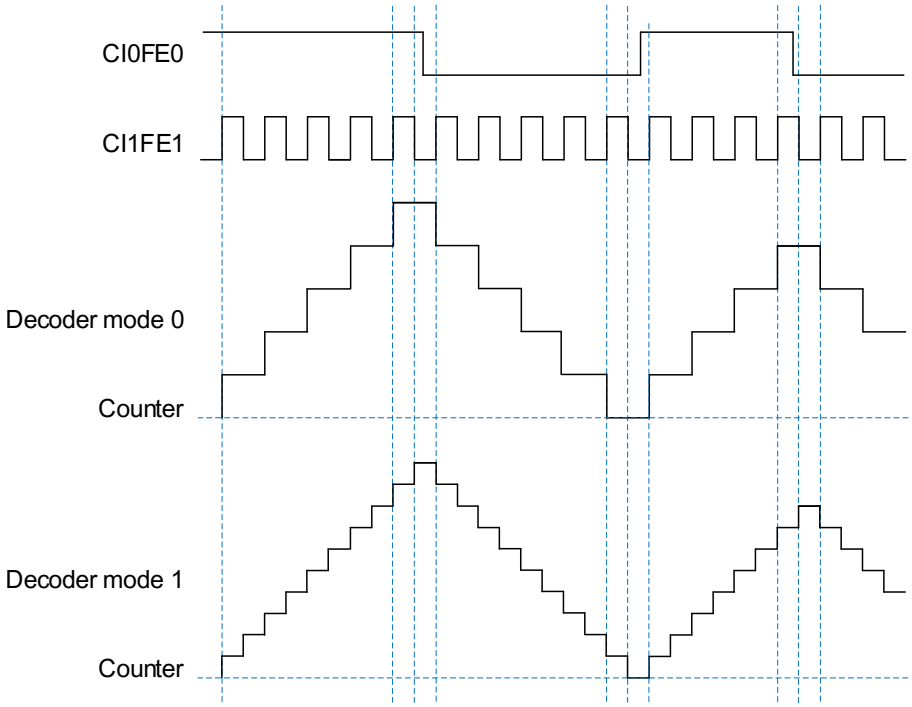
The CH0P is used to configure the count direction selection: When CH0P =0, the counter will count up when the CI0FE0 signal is high and will count down when the CI0FE0 signal is low; When CH0P =1, the counter will count down when the CI0FE0 signal is high and will count up when the CI0FE0 signal is low.

The CH1P is used to configure the count pulse edge selection: In decoder mode 0, the counter will count on both rising and falling edges of the CI1FE1 signal. In 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 decoder mode 1 is shown in ***Table 23-9. the counter operation in decoder mode 1*** and ***Figure 23-64. Example of counter operation in decoder mode 0 / 1 with CH1P=0***.

Table 23-9. the counter operation in decoder 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 23-64. Example of counter operation in decoder mode 0 / 1 with CH1P=0



The 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 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 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 decoder mode 2/3 is shown in ***Figure 23-65. Example of counter operation in decoder mode 2 / 3 (CH0P / CH1P=0)*** and ***Table 23-10. the counter operation in decoder mode 2 / 3.***

Figure 23-65. Example of counter operation in decoder mode 2 / 3 (CH0P / CH1P=0)

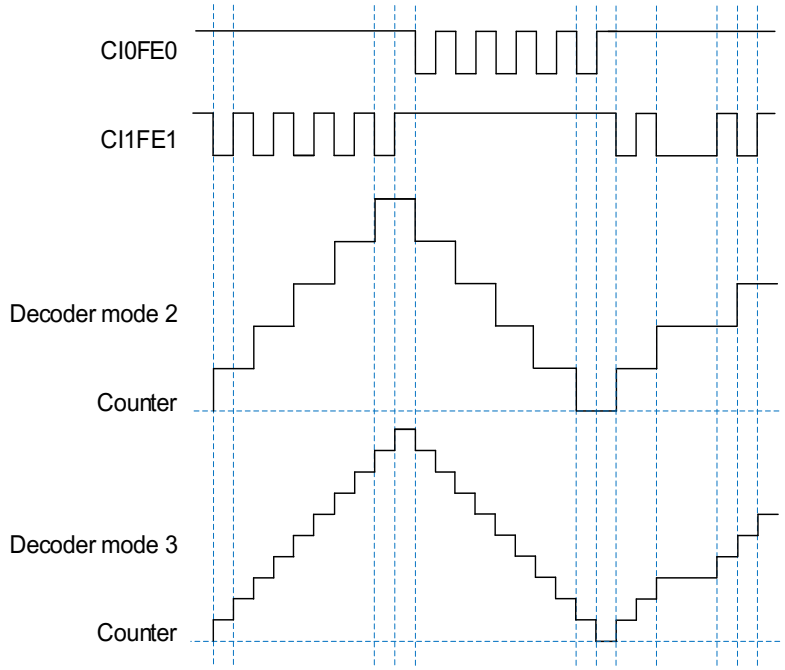


Table 23-10. the counter operation in decoder mode 2 / 3

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
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
decoder mode 3 TSCFG12[4:0] != 5'b00000	CHxP=0 (x=0 or 1)	CI1FE1=1	-	Down	x	x
		CI1FE1=0	-	-	x	x
		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	-	-

Counting mode	Polarity	Level	CI0FE0		CI1FE1	
			Rising	Falling	Rising	Falling
		CI0FE0=0	x	x	Up	-

When the counter direction changes in the 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 decoder clock output

TIMERs can output the decoder clock output signal as TRGO. This function can be enabled by setting the MMC0[3:0] bit-field to 4'b1000 in the TIMERx_CTL1 register, and just used in quadrature decoder mode 0~4 and decoder mode 0~3.

Hall sensor function

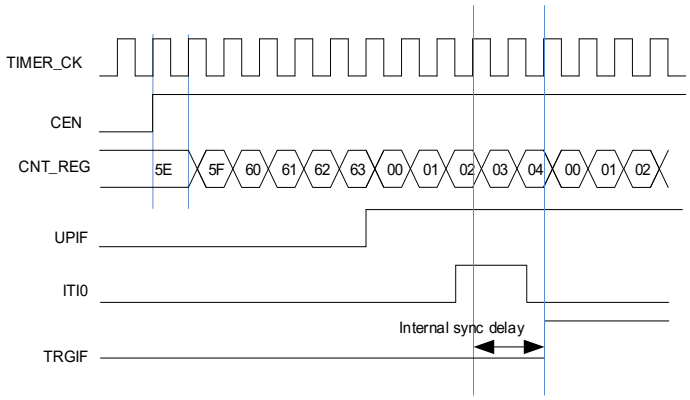
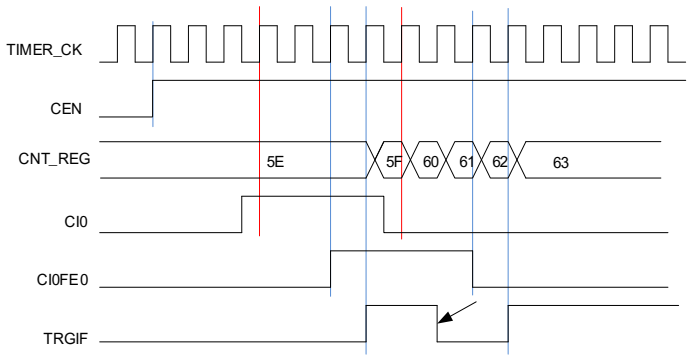
Refer to ***Advanced timer (TIMERx, x=0, 7, 19)*** **Hall sensor** function.

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).

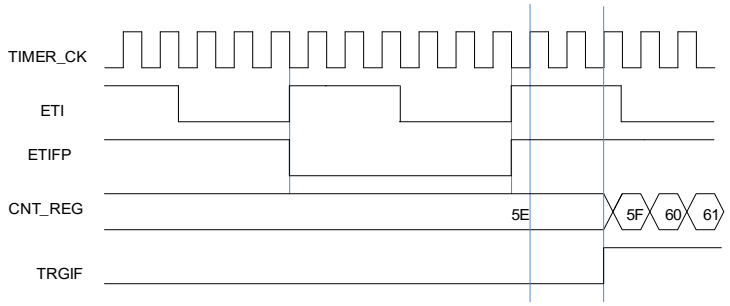
Table 23-11. Examples of slave mode

	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.
Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of	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.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	trigger input comes.			
	Figure 23-66. 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 23-67. 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 23-68. Event mode



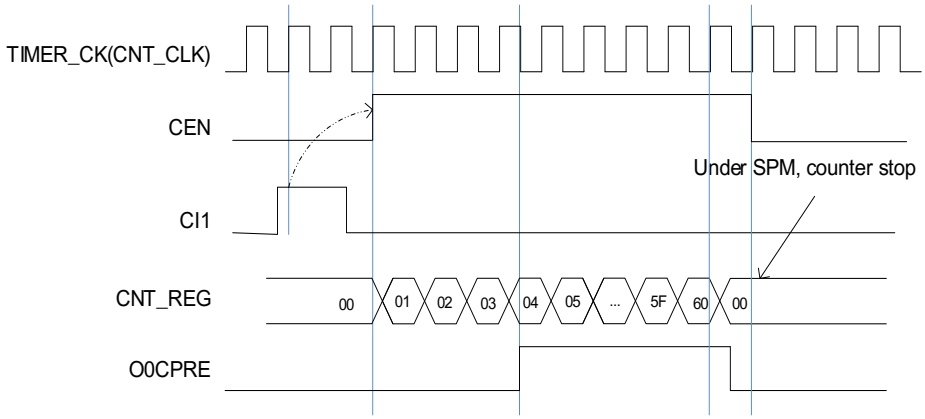
Single pulse mode

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 23-69. Single pulse mode TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60



Timers interconnection

Please refer to ***Advanced timer (TIMERx, x=0, 7, 19)*****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.

23.2.5. 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.

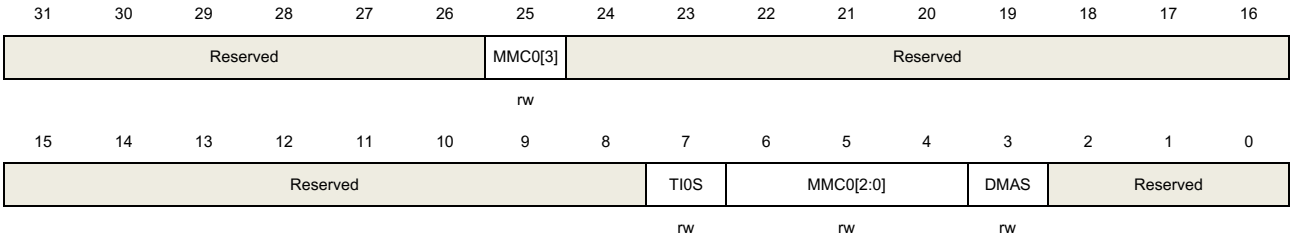
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode or decoder mode.
3	SPM	Single pulse mode 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The CEN bit is cleared by hardware and 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	MMC0[3]	Master mode control 0 Refer to MMC0[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	MMC0[2:0]	Master mode control 0 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 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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	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

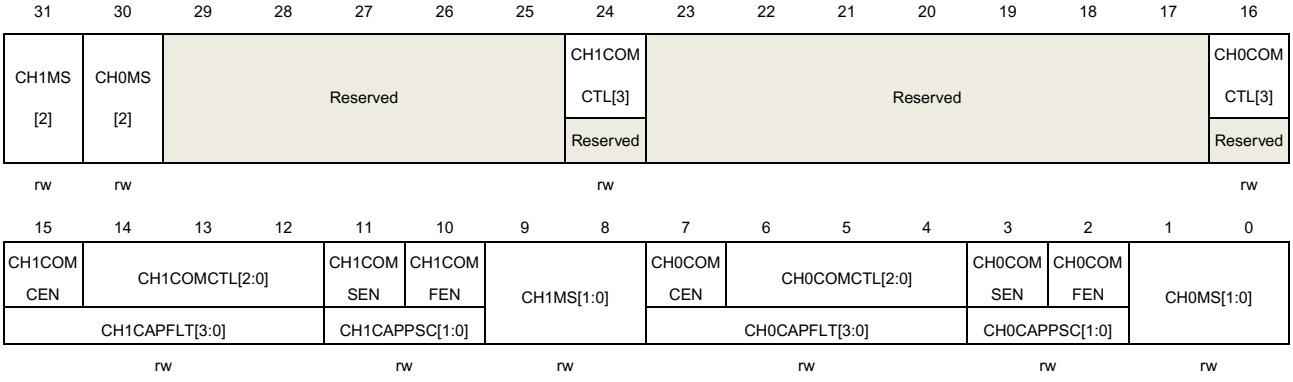
2	CH1G	Channel 1 capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0 capture or compare event generation This bit is set by software to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag will be set, 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).



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 When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. 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.

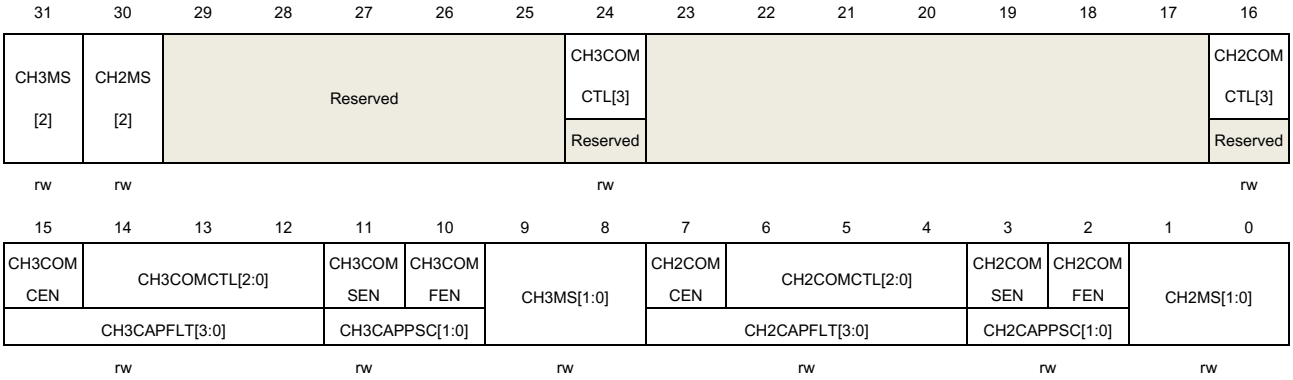
10: Capture is done every 4 channel input edges.
11: Capture is done every 8 channel input edges.

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 When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM1 or PWM2 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the result of the comparison. 0: Channel 2 output quickly compare disable. 1: Channel 2 output quickly compare enable.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. 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).

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

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

Bits	Fields	Descriptions
31:16	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).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[31:16]															
rw															

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

CH0VAL[15:0]
rw

Bits	Fields	Descriptions
31:16	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															

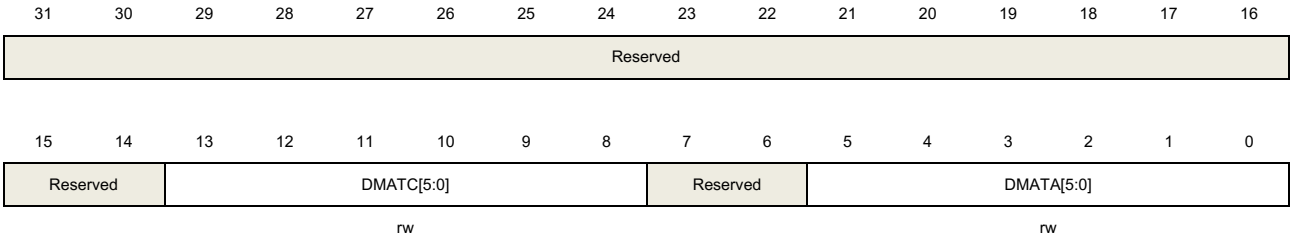
Bits	Fields	Descriptions
31:0	WDGPER[31:0]	Watchdog counter period value This register contains the period of the two watchdog counter. When the counters continus to count to this value, the counter will timeout and the interrupt flag DECDISIF is set. If DECDISIE=1, the corresponding interrupt is generated. **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).



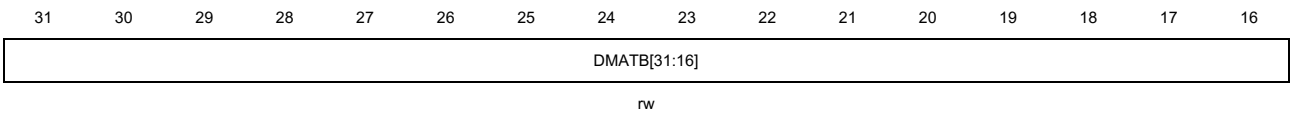
Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:8	DMATC[5:0]	DMA transfer count This field defines the times of accessing(R/W) the TIMERx_DMATB register by DMA. 6'b000000: transfer 1 time 6'b000001: transfer 2 times … 6'b111111: transfer 63 times
7:6	Reserved	Must be kept at reset 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 … 6'b111111: TIMERx_DMACFG 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).



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														CHVSEL	Reserved
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection bit This bit-field is set and reset by software. 1: If the value to be written to the CHxVAL register is the same as the value of CHxVAL register, the write access is ignored. 0: No effect.
0	Reserved	Must be kept at reset value.

23.3. General level2 timer (TIMERx, x=9, 10, 12, 13)

23.3.1. Overview

The general level2 timer module (Timer9, 10, 12, 13) is a one-channel timer that supports input capture, output compare. They can generate PWM signals to control motor or be used for power management applications. The general level2 time reference is a 16-bit counter that can be used as an unsigned counter.

In addition, the general level2 timers can be programmed and be used to count or time external events that drive other Timers.

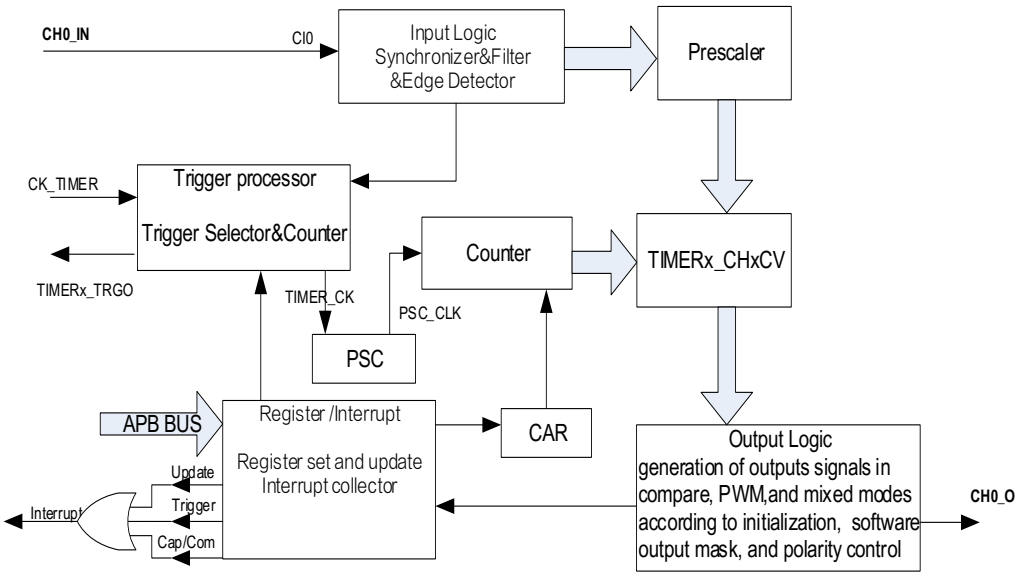
23.3.2. Characteristics

- Total channel num: 1.
- Counter width: 16bit.
- Source of count clock: internal clock.
- Counter mode: count up only.
- Programmable prescaler: 16 bits. Factor can be changed on the go.
- Each channel is user-configurable:
  Input capture mode, output compare mode, programmable and PWM mode.
- Auto-reload function.
- Interrupt output on: update event and compare/capture event.

23.3.3. Block diagram

Figure 23-70. General level2 timer block diagram provides details on the internal configuration of the general level2 timer.

Figure 23-70. General level2 timer block diagram



23.3.4. Function overview

Clock source configuration

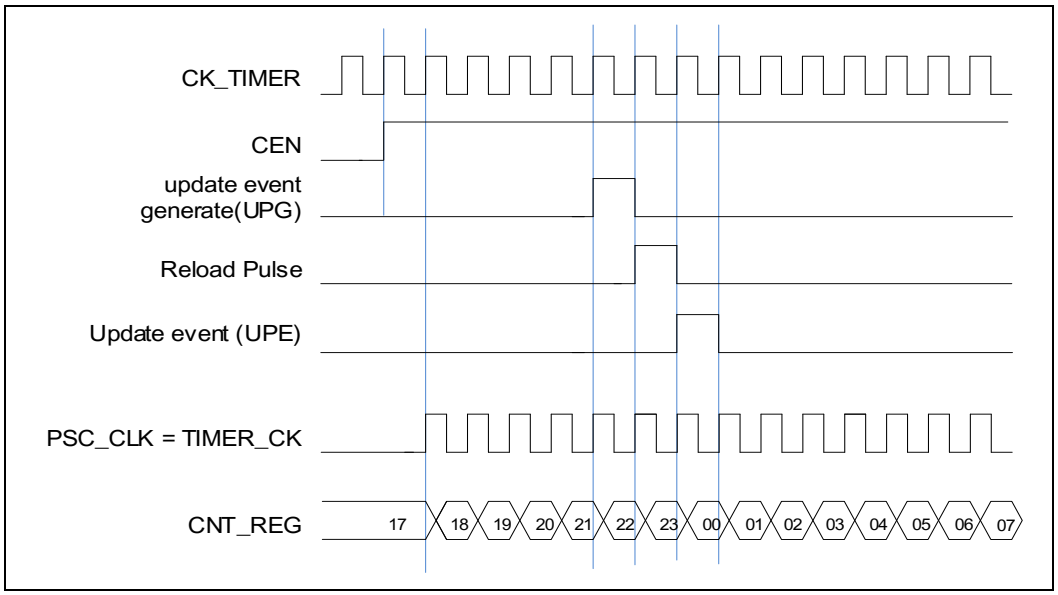
The general level2 TIMER can only being clocked by the CK_TIMER.

■ Internal timer clock CK_TIMER which is from module RCU

The general level2 TIMER has only one clock source which is the internal CK_TIMER, used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

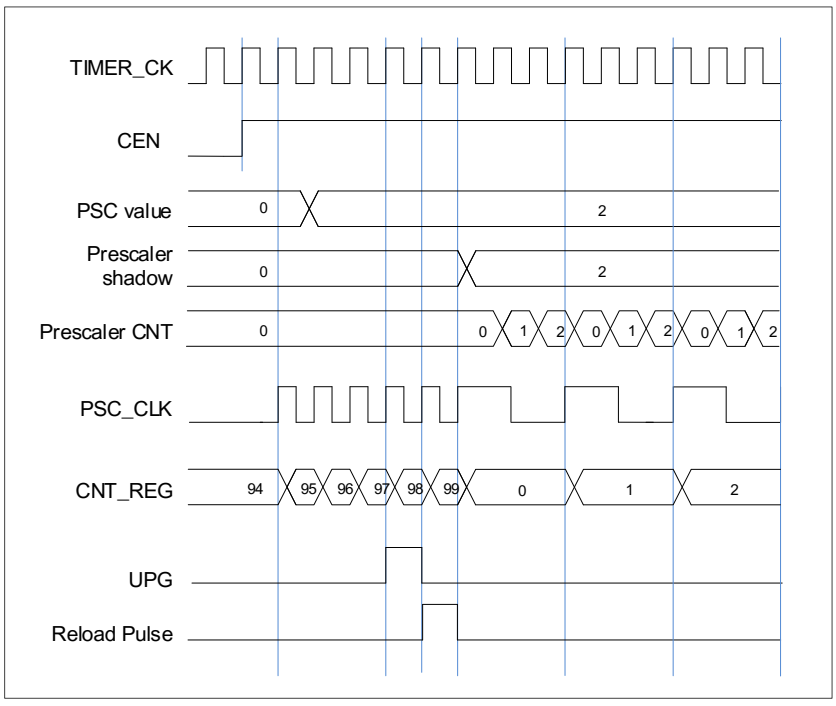
Figure 23-71. Timing chart of internal clock divided by 1



Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 23-72. Timing chart of PSC value change from 0 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again. The update event is generated at each counter overflow.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and generates an update event.

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

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

Figure 23-73. Timing chart of up counting mode, PSC=0/2 and *Figure 23-74. Timing chart of up counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 23-73. Timing chart of up counting mode, PSC=0/2

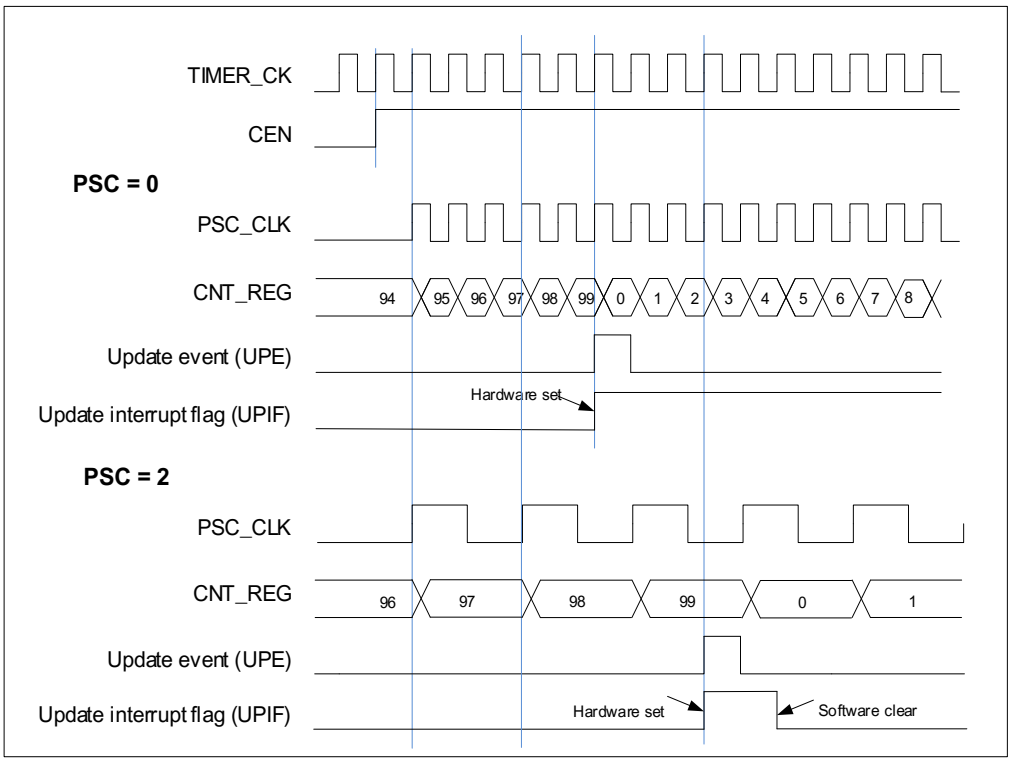
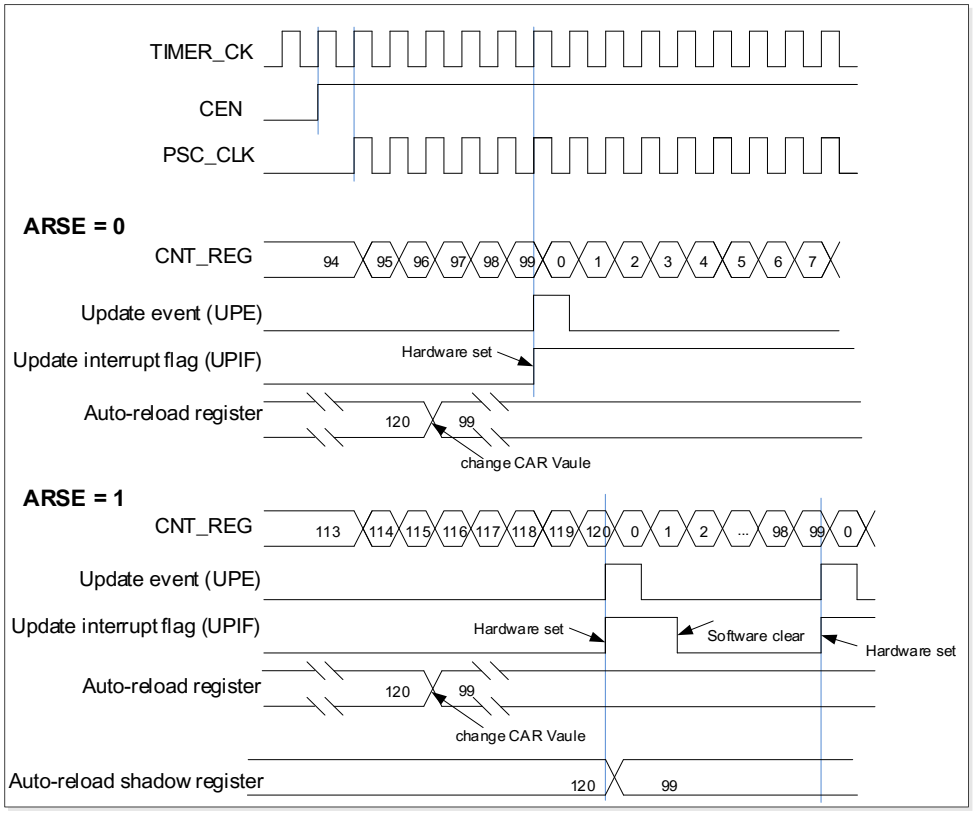


Figure 23-74. Timing chart of up counting mode, change TIMERx_CAR ongoing



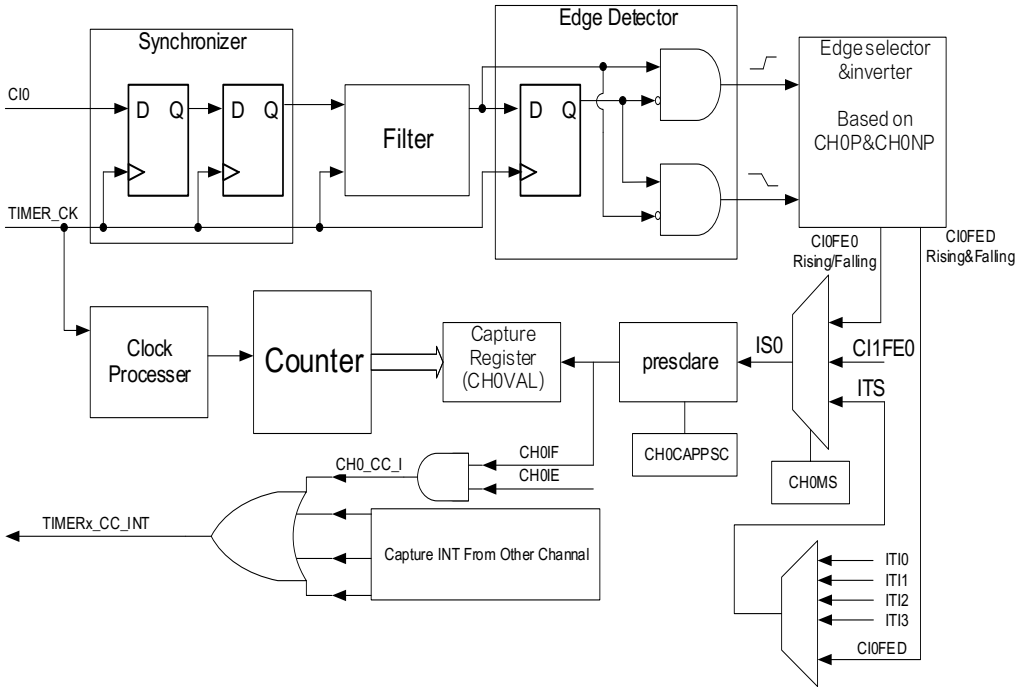
Input capture and output compare channels

The general level2 timer has one independent channel which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

■ Channel input capture function

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 enabled by CHxIE = 1.

Figure 23-75. Channel input capture principle



First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and fall edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, TIMERx_CHxCV will restore the value of counter.

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

Step1: Filter configuration. (CHxCAPFLT in TIMERx_CHCTL0)

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

Step2: Edge selection. (CHxP/CHxNP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel to input mode (CHxMS!=0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable. (CHxIE in TIMERx_DMAINTEN)

Enable the related interrupt enable; you can got the interrupt.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2)

Result: When you wanted input signal is got, TIMERx_CHxCV will be set by Counter's value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt will be asserted based on the your configuration of CHxIE in TIMERx_DMAINTEN

Direct generation: If you want to generate a DMA request or Interrupt, you can set CHxG by software directly.

The channel input capture function can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

■ **Channel output compare function**

Figure 23-76. Channel output compare principle

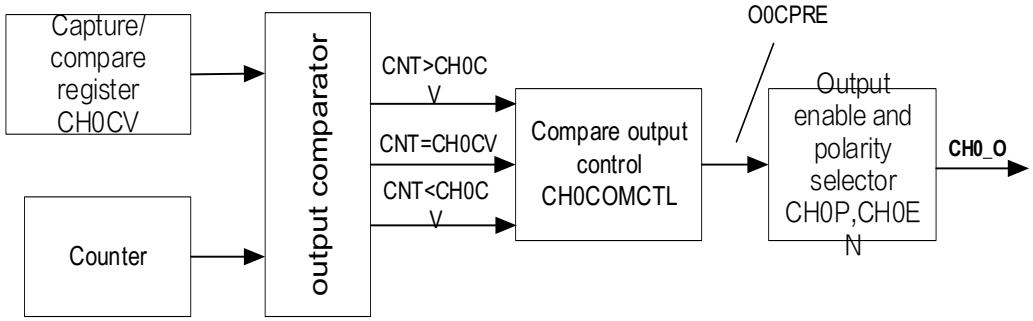


Figure 23-76. Channel output compare shows the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output prepare signal*) is described as blew: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE signal, CHxP bit and CHxEN 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 CHxVAL register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. when the counter reaches the value in the CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1.

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

Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

* Set the shadow enable mode by CHxCOMSEN

* Set the output mode (Set/Clear/Toggle) by CHxCOMCTL.
* Select the active high polarity by CHxP/CHxNP
* Enable the output by CHxEN

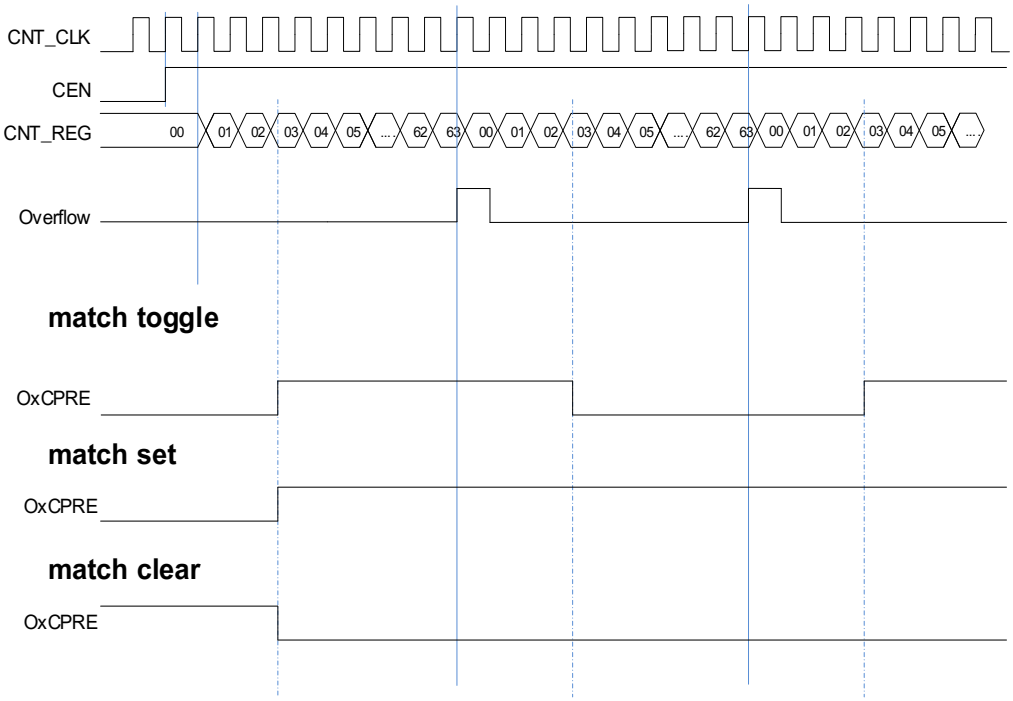
Step3: Interrupt/DMA-request enables configuration by CHxIE

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV. About the CHxVAL, you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

Figure 23-77. Output-compare under three modes show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 23-77. Output-compare under three modes



Channel output prepare signal

As is shown in *Figure 23-76. Channel output compare* , when the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL filed. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal

level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. With regard to a more detail description refer to the relative bit definition.

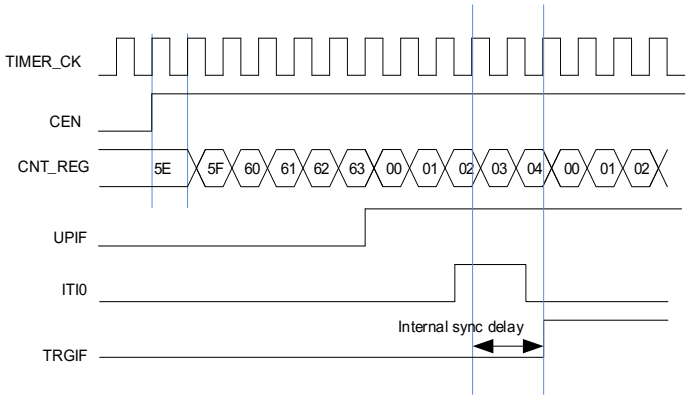
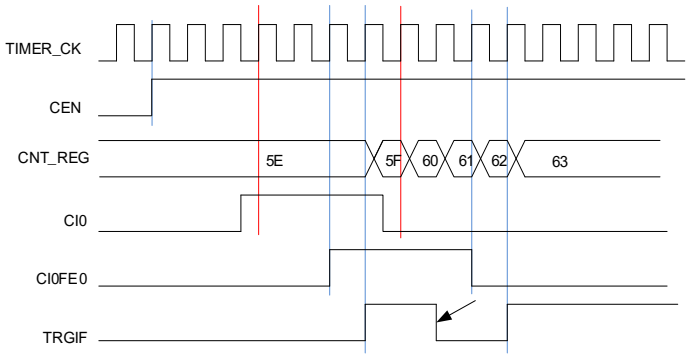
Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

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).

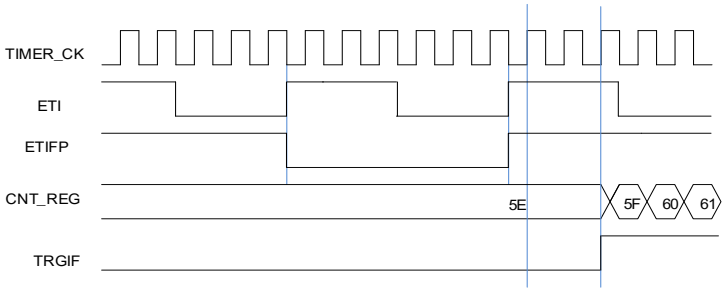
Table 23-12. Examples of slave mode

	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 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 output of external trigger input ETI) is selected as the trigger source, configure the ETP for polarity selection and inversion.	For the ITIx, no filter and prescaler can be used. For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used. For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.
Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of trigger input comes.	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.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 23-78. 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 23-79. 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 23-80. Event mode



Timers interconnection

Please refer to ***Advanced timer (TIMERx, x=0, 7, 19)*****Timers interconnection**.

Timer debug mode

When the Cortex®-M33 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

23.3.5. TIMERx registers(x=9, 10, 12, 13)

TIMER9 base address: 0x4001 5000

TIMER10 base address: 0x4001 5400

TIMER12 base address: 0x4000 1C00

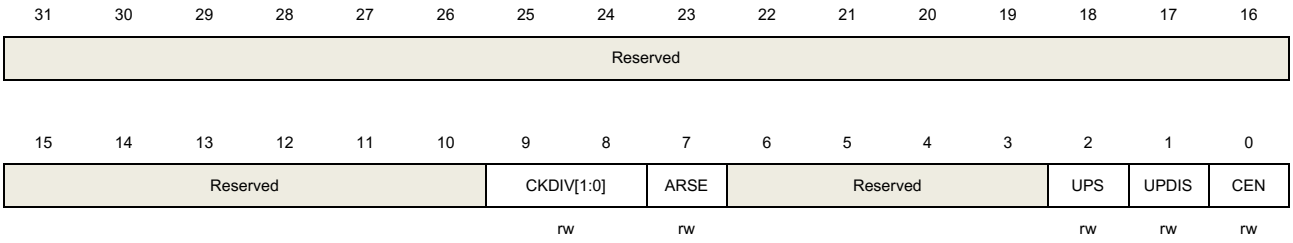
TIMER13 base address: 0x4000 2000

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: $f_{DTS}=f_{CK_TIMER}$ 01: $f_{DTS}= f_{CK_TIMER}$ /2 10: $f_{DTS}= f_{CK_TIMER}$ /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:3	Reserved	Must be kept at reset value.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate update interrupts or DMA requests: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: This event generates update interrupts or DMA requests:

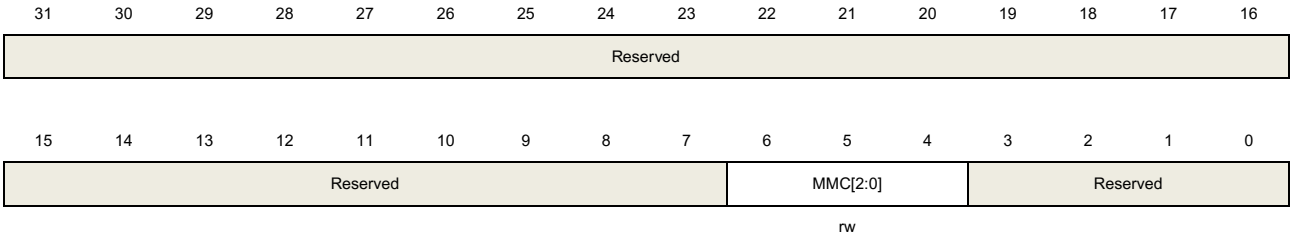
		The counter generates an overflow or underflow event
1	UPDIS	Update disable. This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: Update event disable. **Note:** When this bit is set to 1, setting UPG bit or the restart mode does not generate an update event, but the counter and prescaler are initialized.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock, pause mode and quadrature decode mode.

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 in master mode to slave timers for synchronization function. 000: When a counter reset event occurs, a TRGO trigger signal is output. The counter resert source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a conter start event occurs, a TRGO trigger signal is output. The counter start source :

CEN control bit is set

The trigger input in pause mode is high

010: When an update event occurs, a TRGO trigger signal is output. The update source depends on UPDIS bit and UPS bit.

011: When a capture or compare pulse event occurs in channel0, a TRGO trigger signal is output.

100: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O0CPRE.

101: Reserved

110: Reserved

111: Reserved

3:0	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
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.

Interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

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

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

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TRGIE	Reserved					CH0IE	UPIE
								rw						rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRGIE	Trigger interrupt enable 0: Disabled 1: Enabled
5:2	Reserved	Must be kept at reset value.
1	CH0IE	Channel 0 capture/compare interrupt enable 0: disabled 1: enabled
0	UPIE	Update interrupt enable 0: disabled 1: enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CH0OF	Reserved		TRGIF	Reserved				CH0IF	UPIF
						rc_w0			rc_w0					rc_w0	rc_w0

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred

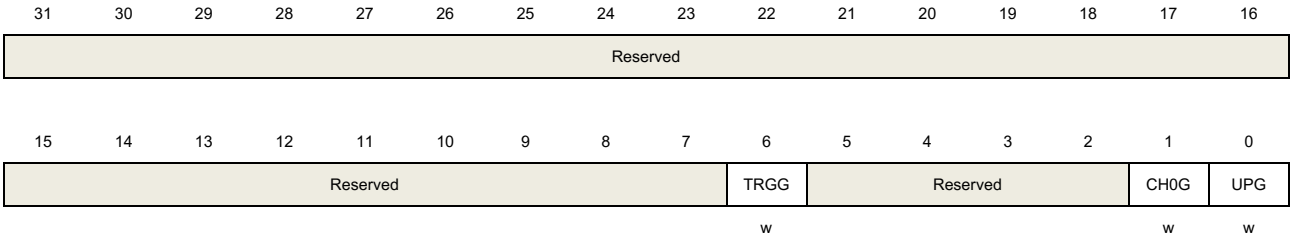
8:7	Reserved	Must be kept at reset value.
6	TRGIF	Trigger interrupt flag This flag is set by hardware on trigger event and cleared by software. When the slave mode controller is enabled in all modes but pause mode, an active edge of trigger input generates a trigger event. When the slave mode controller 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
5:2	Reserved	Must be kept at reset value.
1	CH0IF	Channel 0 's capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. If Channel0 is set to input mode, this bit will be reset by reading TIMERx_CH0CV. 0: No Channel 1 interrupt occurred 1: Channel 1 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:7	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

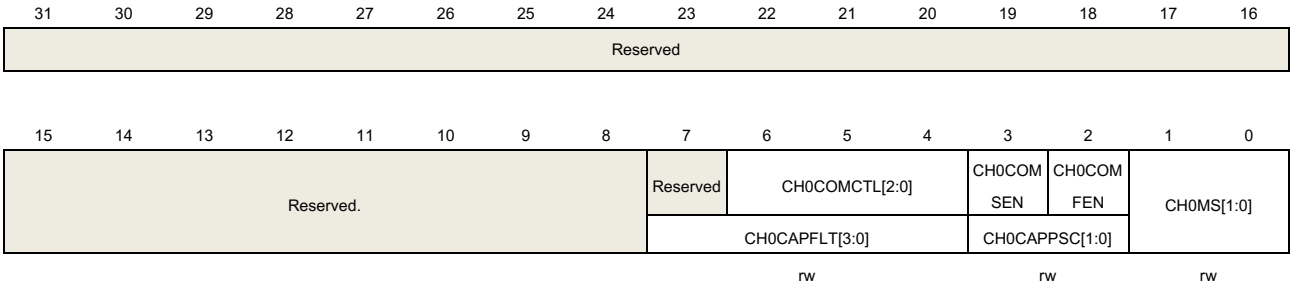
		1: Generate a trigger event
5:2	Reserved	Must be kept at reset value.
1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

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



Output compare mode:

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP bits. 000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 001: Set the channel output. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter is

		equals to the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter is equals to the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced to low level. 101: Force high. O0CPRE is forced to high level. 110: PWM mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. If configured in PWM mode, the O0CPRE level changes only when the output compare mode is adjusted from "Timing" mode to "PWM" mode or the comparison result changes.
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled. 0: Channel 0 output compare shadow disable 1: Channel 0 output compare shadow enable The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1)
2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).). 00: Channel 0 is configured as output 01: Channel 0 is configured as input, IS0 is connected to CI0FE0 10: Reserved 11: Reserved

Input capture mode:

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.

7:4 CH0CAPFLT[3:0] Channel 0 input capture filter control

The CI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability.

Basic principle of digital filter: continuously sample the CI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level.

The filtering capability configuration is as follows:

CH0CAPFLT [3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] Channel 0 input capture prescaler

This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is clear.

00: Prescaler disable, input capture occurs on every channel input edge

01: The input capture occurs on every 2 channel input edges

10: The input capture occurs on every 4 channel input edges

11: The input capture occurs on every 8 channel input edges

1:0 CH0MS[1:0] Channel 0 mode selection

Same as output compare mode

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved..												CH0NP	Reserved	CH0P	CH0EN
												rw		rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 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.
2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in input mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in output mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-filed indicates the current counter value. Writing to this bit-filed can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-filed will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C
Reset value: 0x0000 0000

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

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

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

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

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

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

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

rw

Bits	Fields	Descriptions

31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	Reserved	Must be kept at reset value.

23.4. General level3 timer (TIMERx, x=14, 40, 41, 42, 43, 44)

23.4.1. Overview

The general level3 timer module (TIMER14/40/41/42/43/44) 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.

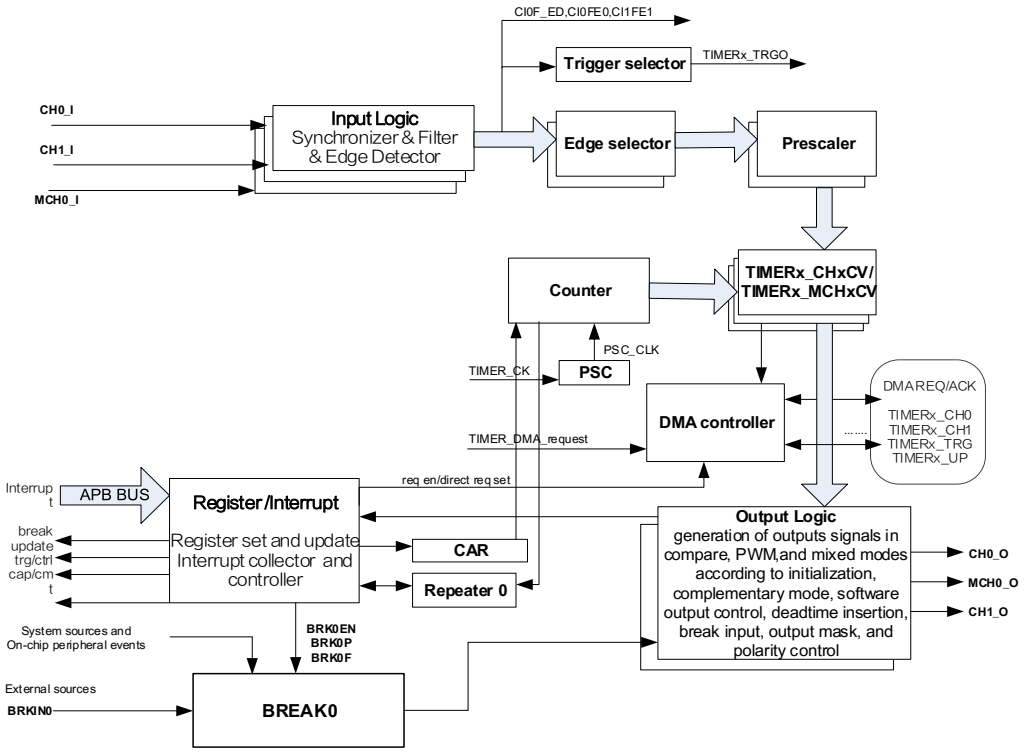
23.4.2. Characteristics

- 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: BREAK0.
- 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.

23.4.3. Block diagram

Figure 23-81. General level3 timer block diagram provides details of the internal configuration of the general level3 timer.

Figure 23-81. General level3 timer block diagram



23.4.4. 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=14,40,41,42,43,44) registers. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

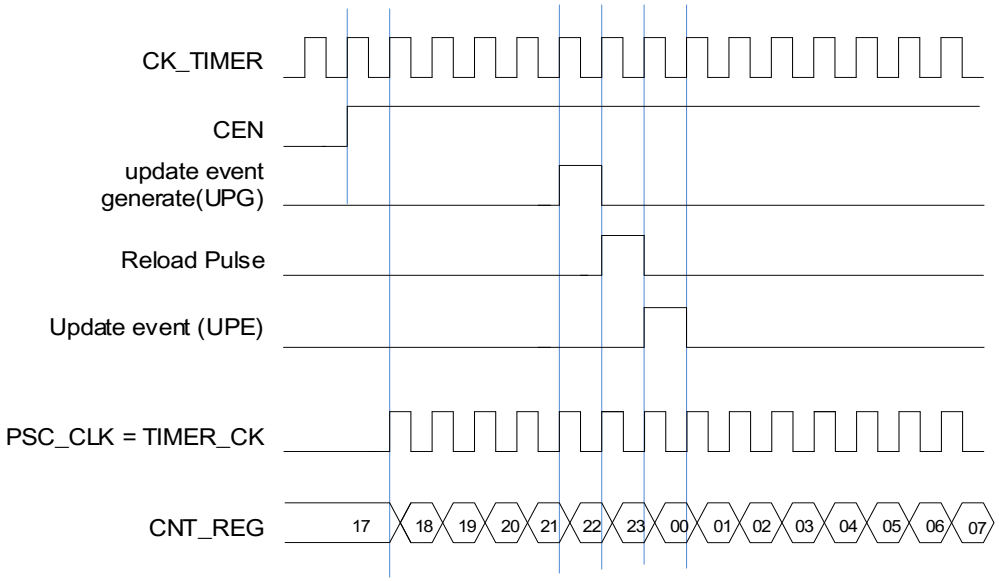
The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[4:0] (y=3..6,15) = 5'b00000 in SYSCFG_TIMERxCFG(x=14,40,41,42,43,44) registers. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

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 23-82. 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

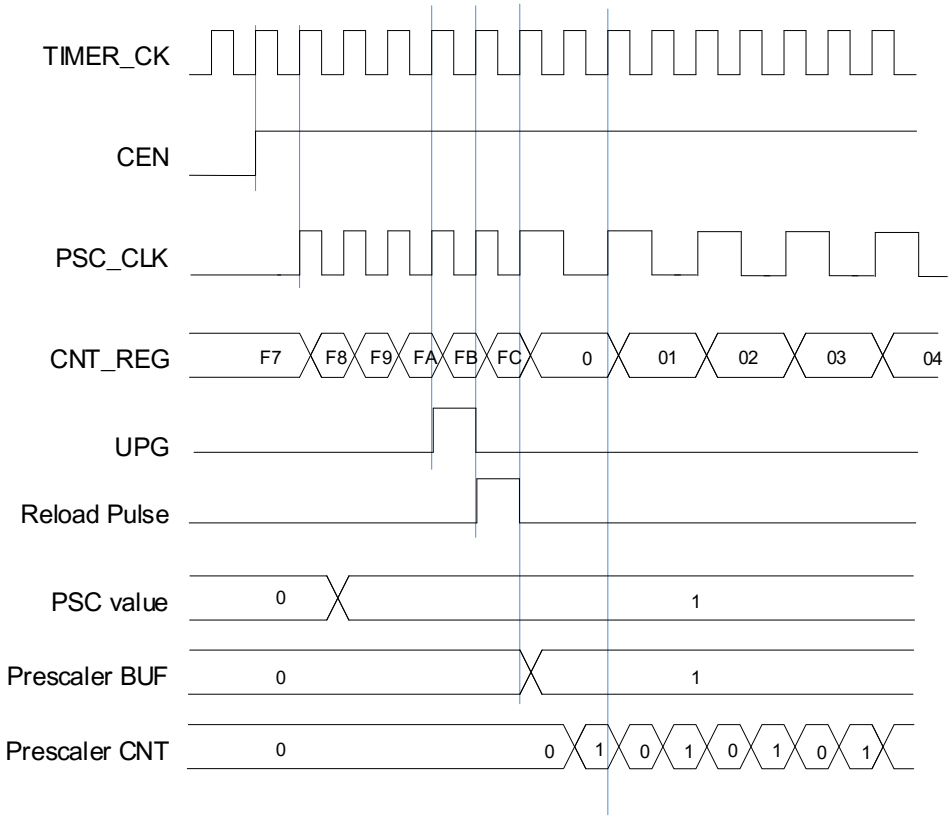
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[4:0] to 0x5~0x7 and 0xB.

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 23-83. 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 23-84. Timing diagram of up counting mode, PSC=0/2 and *Figure 23-85. 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 23-84. Timing diagram of up counting mode, PSC=0/2

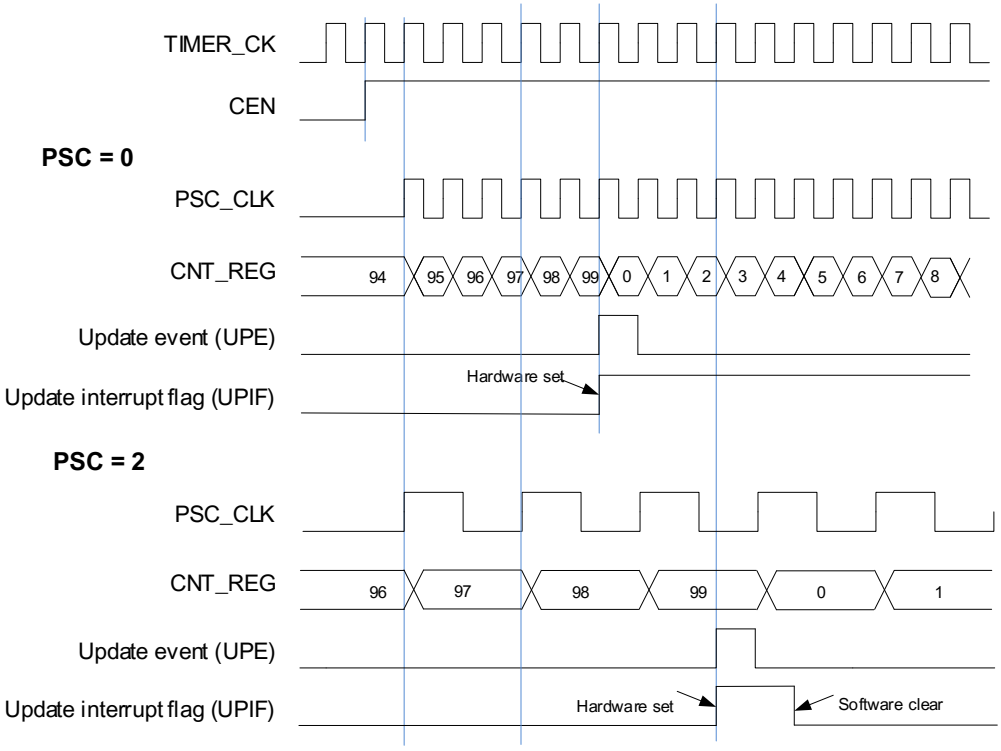
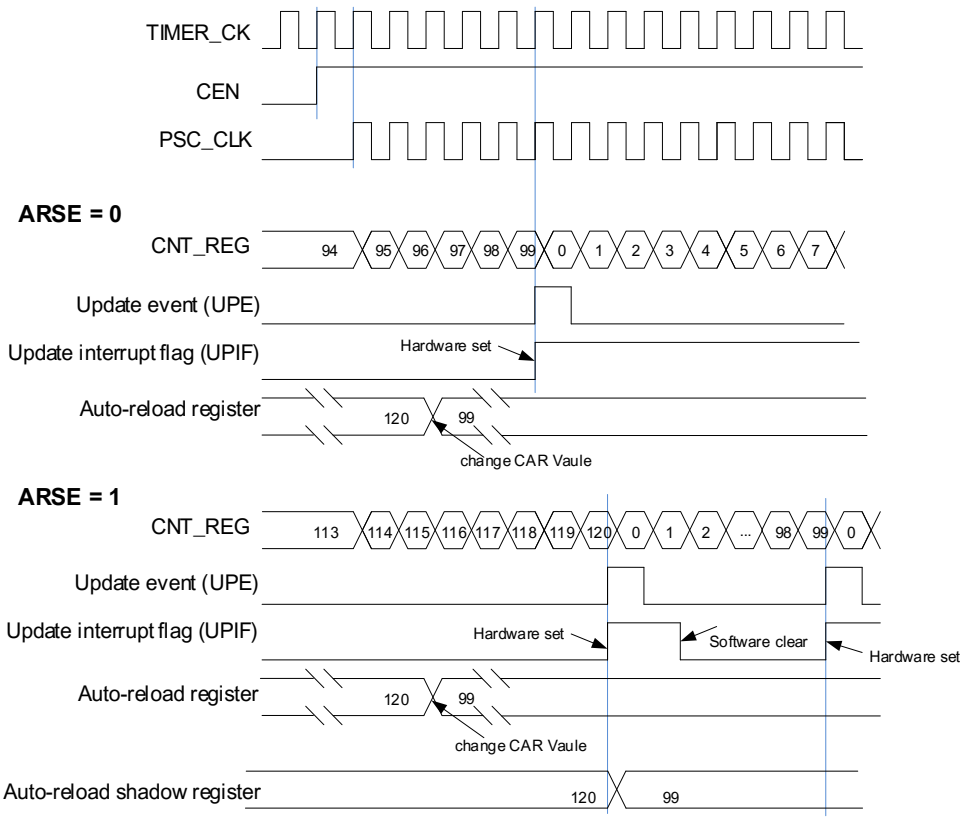


Figure 23-85. Timing diagram of up counting mode, change TIMERx_CAR on the go



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 23-86. Timing diagram of down counting mode, PSC=0/2 and *Figure 23-87. 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 23-86. Timing diagram of down counting mode, PSC=0/2

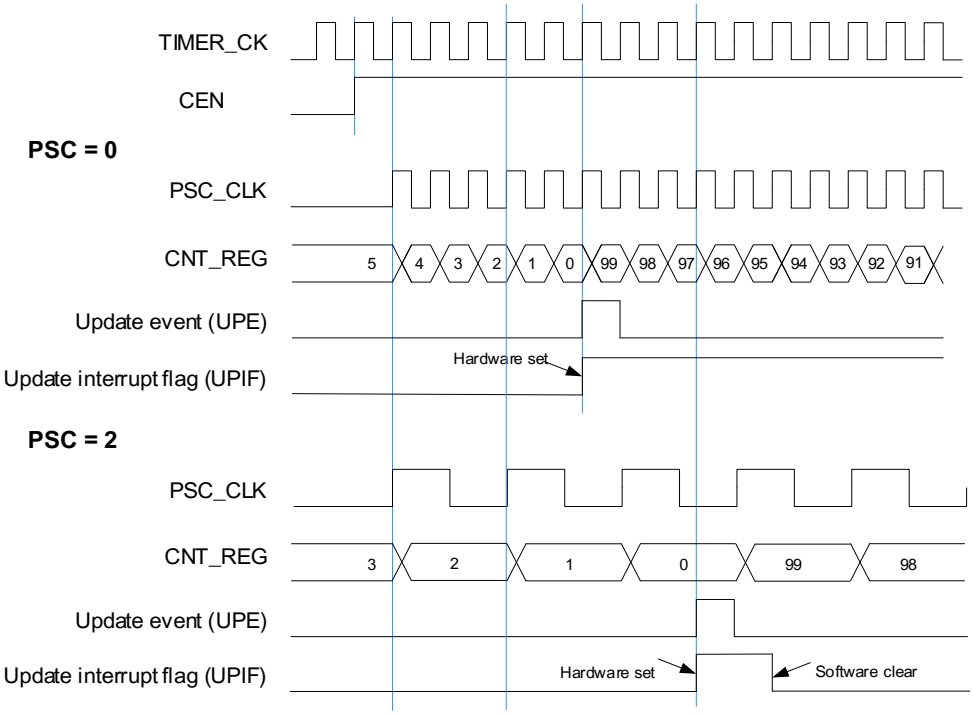
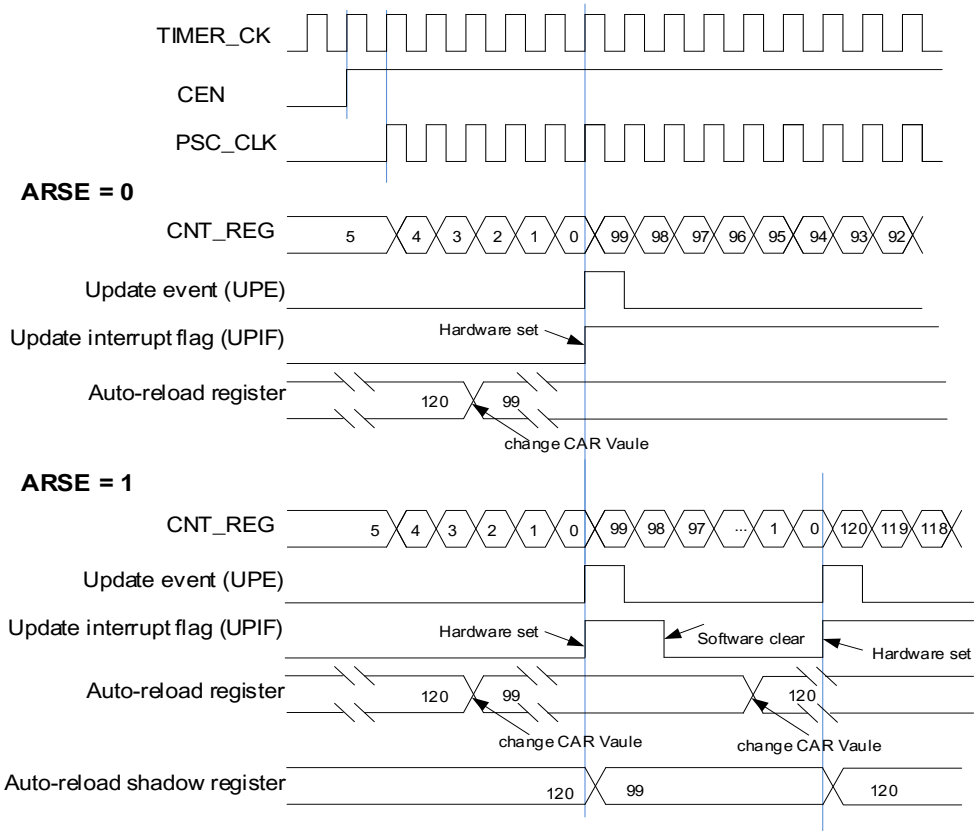


Figure 23-87. Timing diagram of down counting mode, change TIMERx_CAR ongoing



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 23-88. Timing diagram of center-aligned counting mode***.

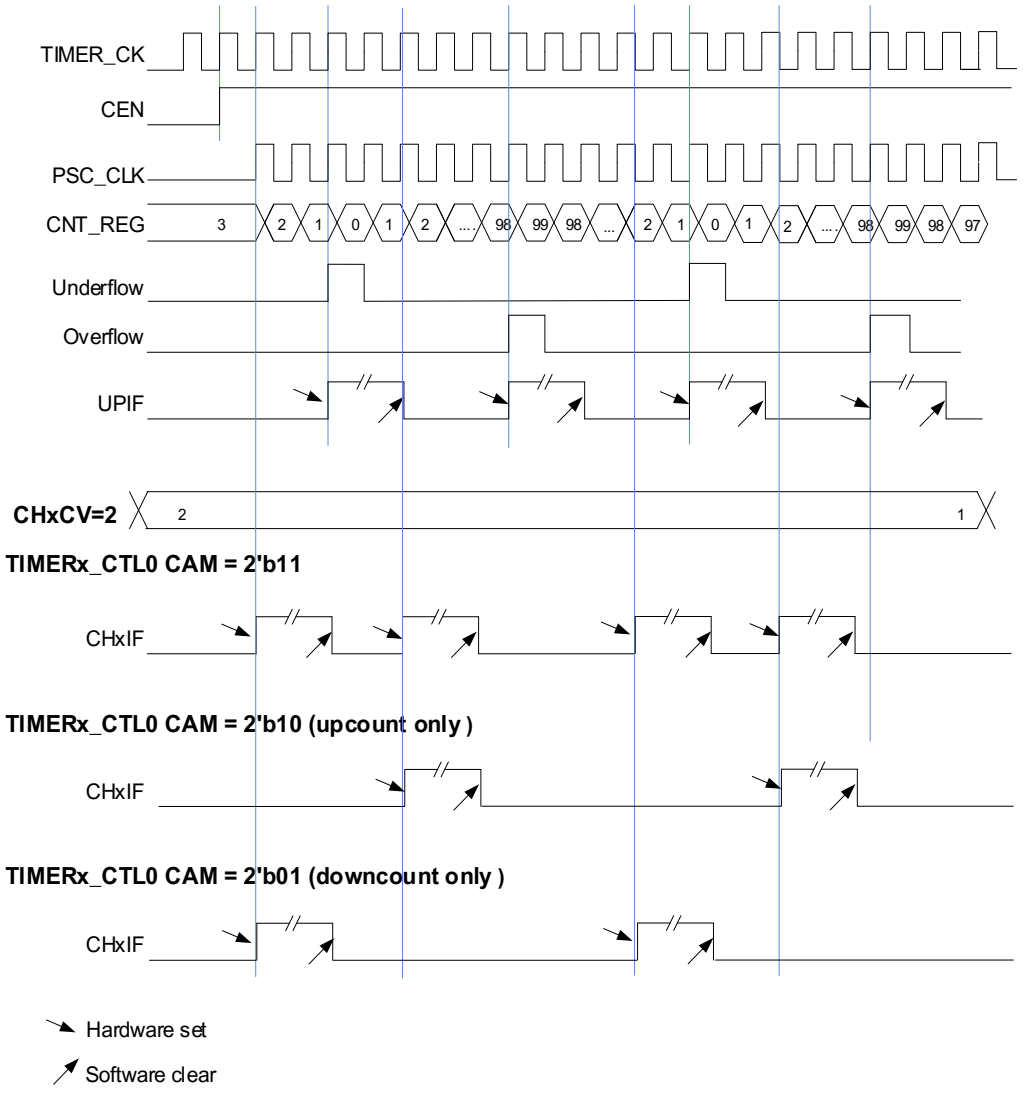
If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, auto-reload register,

prescaler register) are updated.

Figure 23-88. Timing diagram of center-aligned counting mode shows some examples of the counter behavior when TIMERx_CAR=0x99. TIMERx_PSC=0x0.

Figure 23-88. 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 23-89. Repetition counter timing diagram of center-aligned counting mode

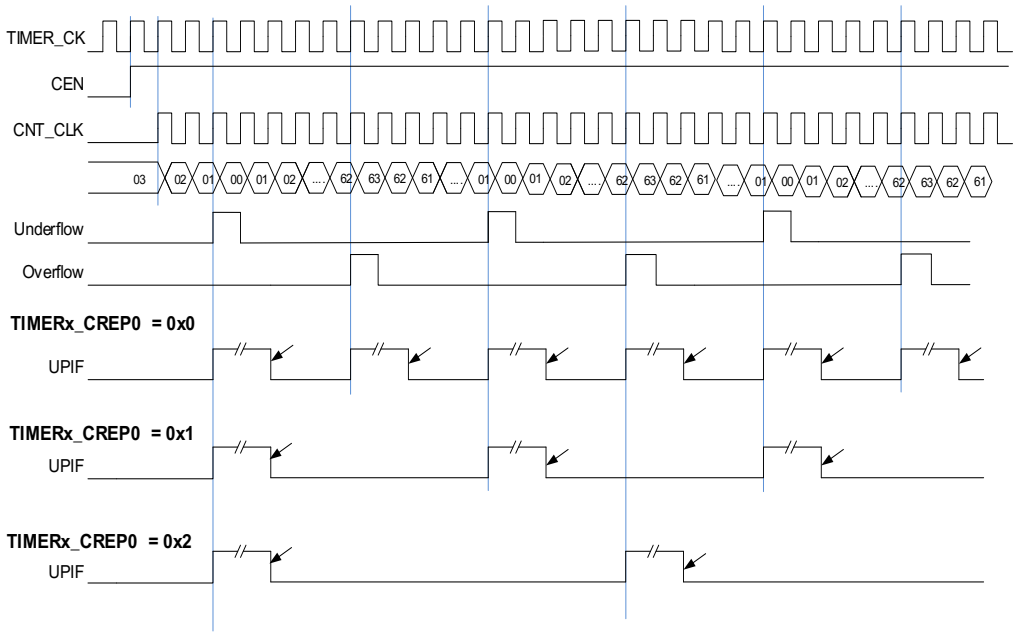


Figure 23-90. Repetition counter timing diagram of up counting mode

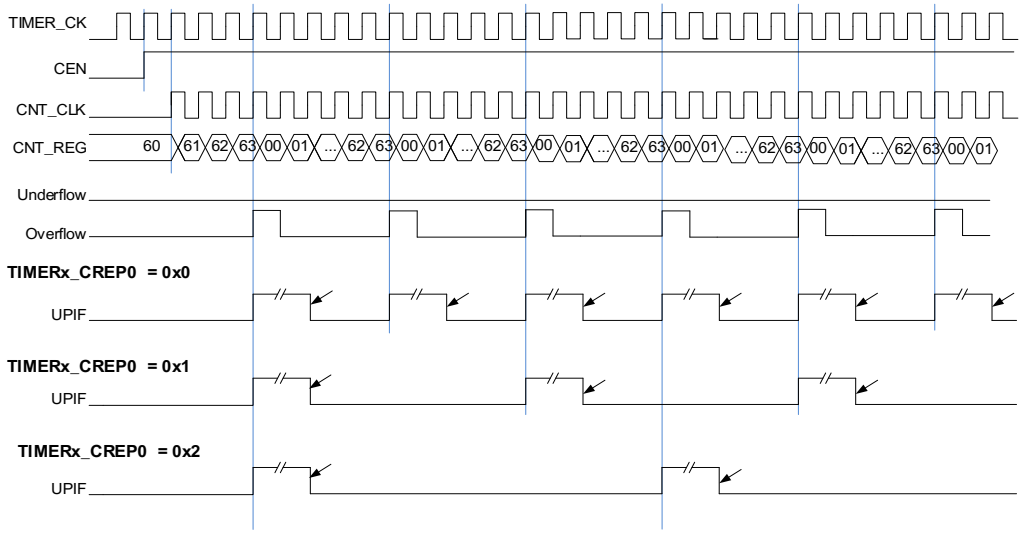
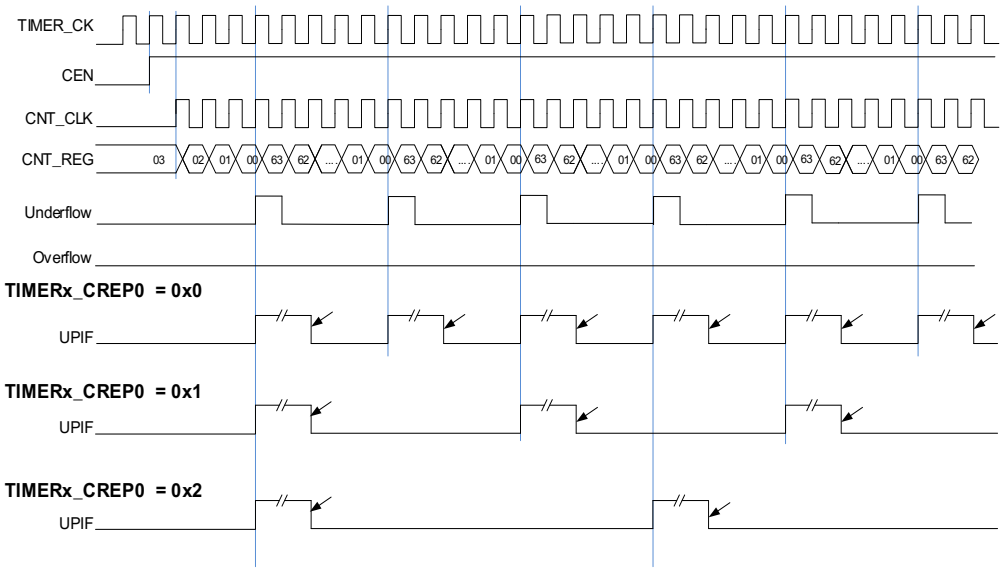


Figure 23-91. Repetition counter timing diagram of down counting mode



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 23-92. Input capture logic for channel 0

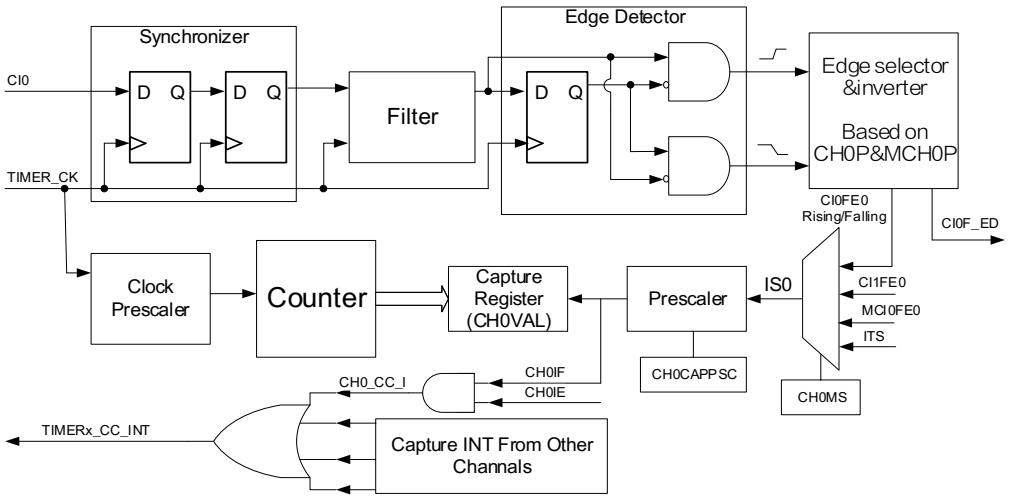
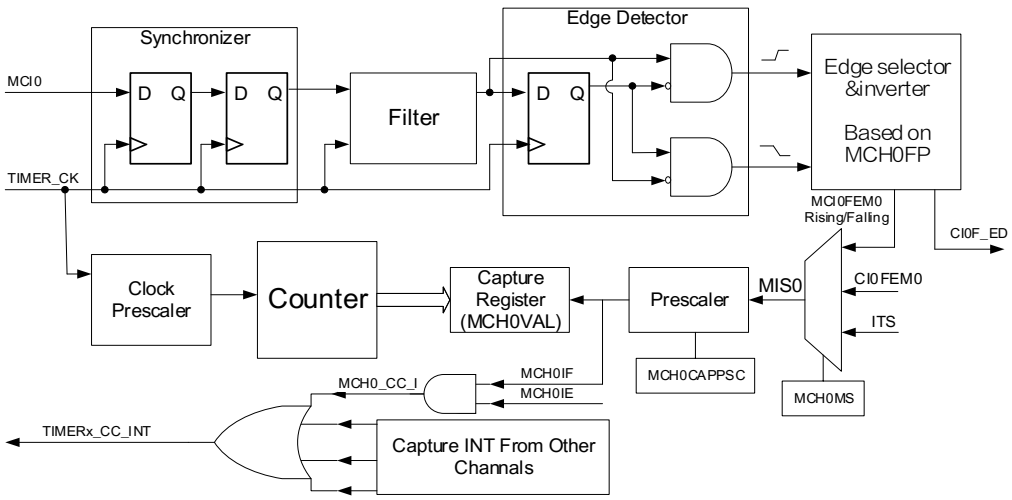


Figure 23-93. Input capture logic for multi mode channel 0



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 23-94. Output compare logic (when MCHxMSEL = 2'00, x=0), *Figure 23-95. Output compare logic (when MCHxMSEL = 2'11, x=0)* and *Figure 23-96. Output compare logic (x=1)* show the logic circuit of output compare mode.

Figure 23-94. Output compare logic (when MCHxMSEL = 2'00, x=0)

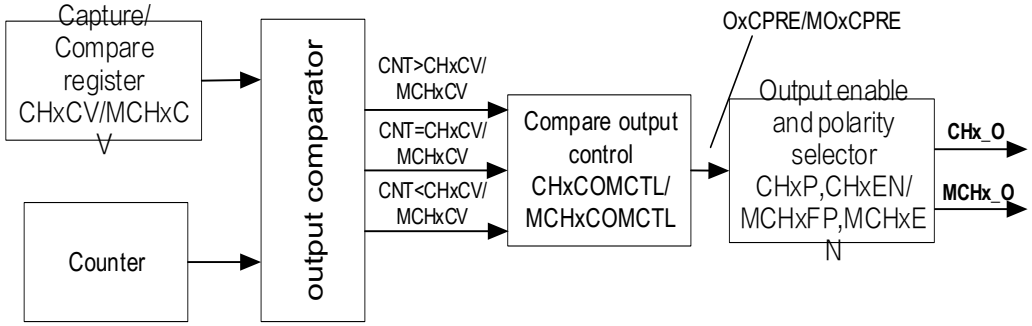


Figure 23-95. Output compare logic (when MCHxMSEL = 2'11, x=0)

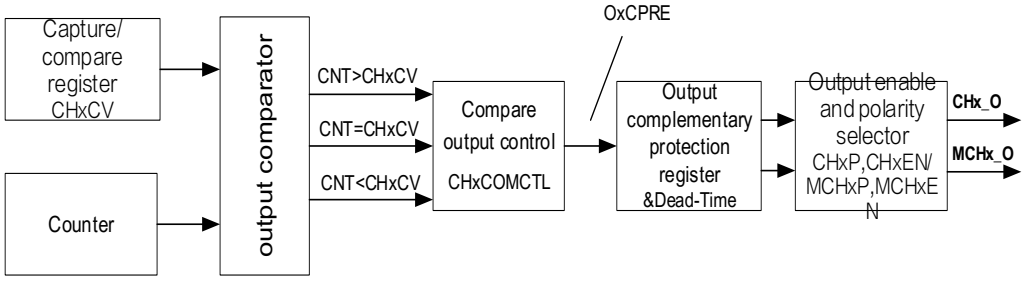
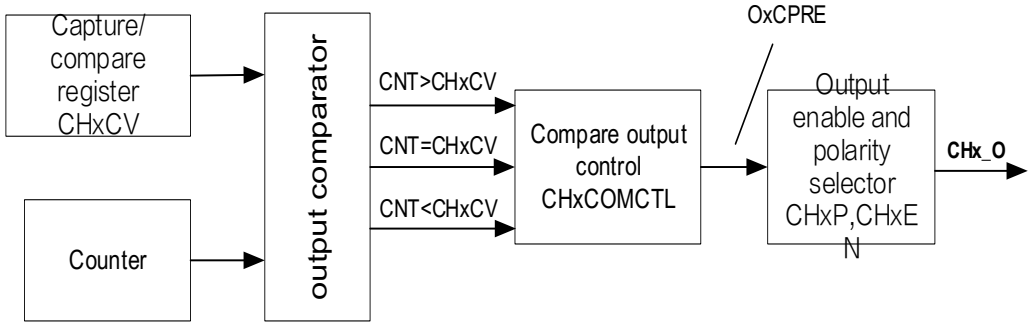


Figure 23-96. Output compare logic (x=1)



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 23-94. 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 23-95. 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.

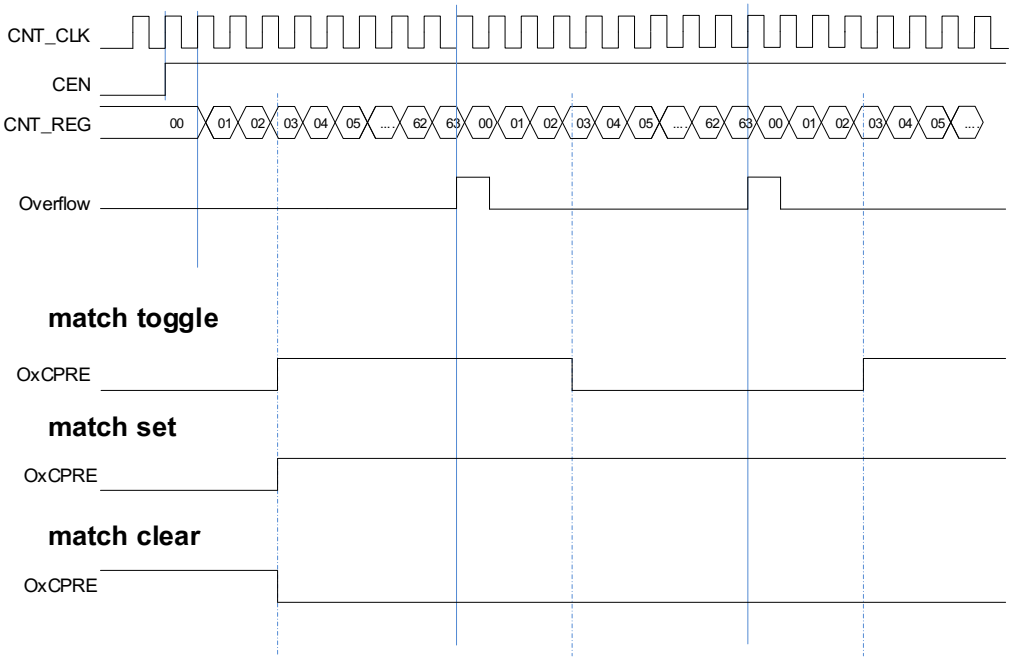
The TIMERx_CHxCV/ TIMERx_MCHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 23-97. Output-compare in three modes shows the three compare modes:

toggle/set/clear. CARL=0x63, CHxVAL=0x3.

Figure 23-97. 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 23-98. 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 23-99. 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 23-98. Timing diagram of EAPWM

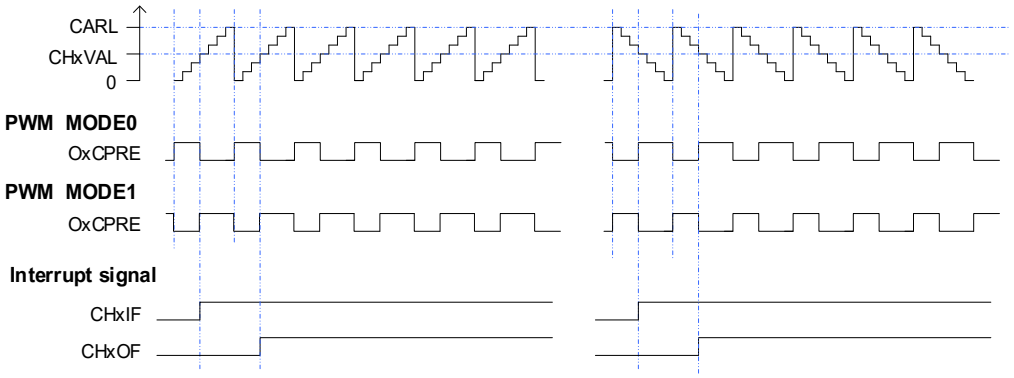
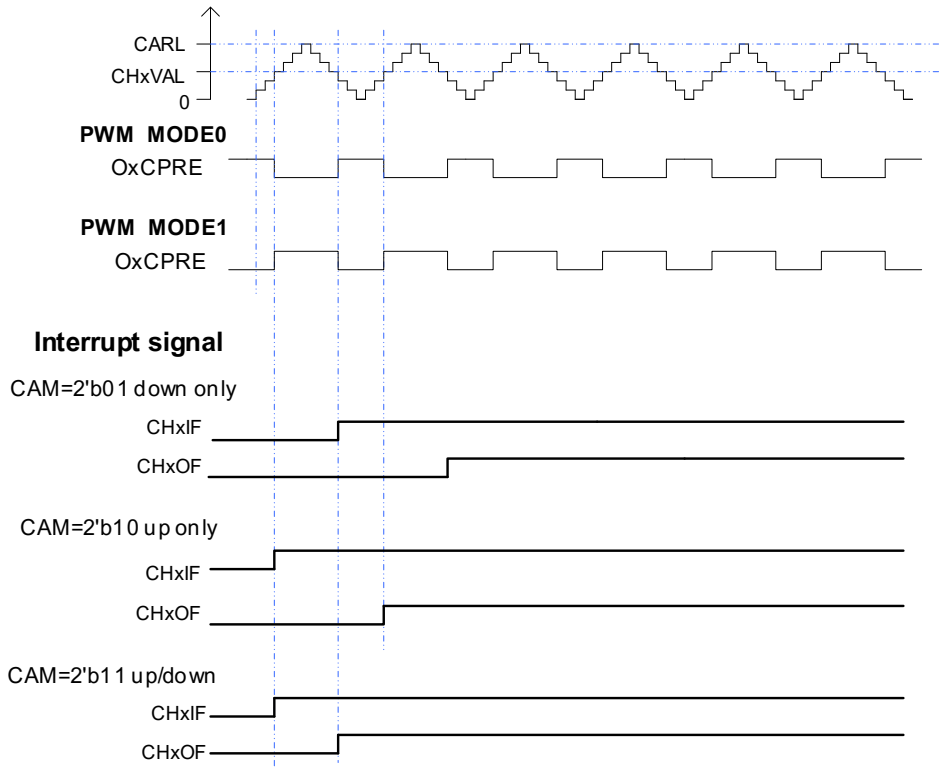


Figure 23-99. Timing diagram of CAPWM



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 23-13.The Composite PWM pulse width***.

Table 23-13.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

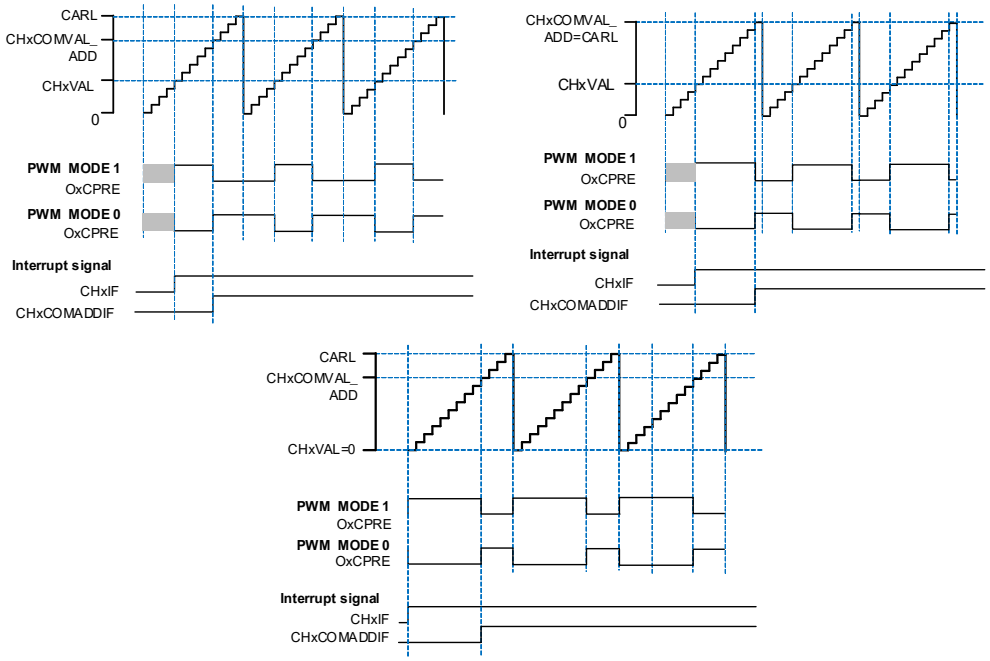
When the counter reaches the value of CHxVAL, the CHxIF bit is set and the channel x interrupt is generated if CHxIE = 1, and the DMA request will be asserted, if CHxDEN=1. When the counter reaches the value of CHxCOMVAL_ADD, the CHxCOMADDIF bit is set (this flag just used in composite PWM mode, when 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:

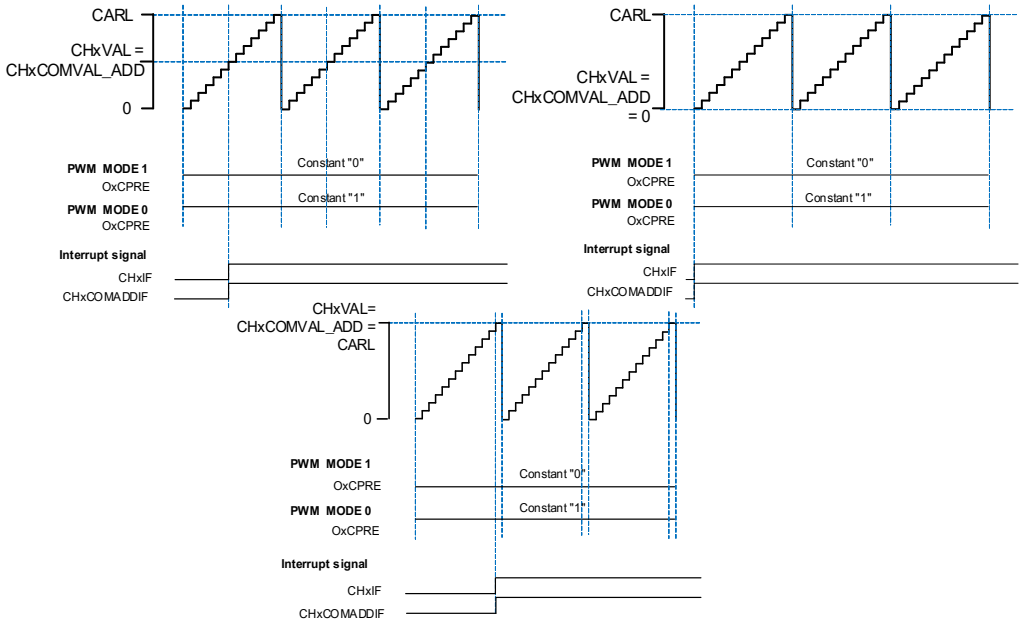
1) CHxVAL < CHxCOMVAL_ADD, and the values of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

Figure 23-100. Channel x output PWM with (CHxVAL < CHxCOMVAL_ADD)



2) CHxVAL = CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD between 0 and CARL.

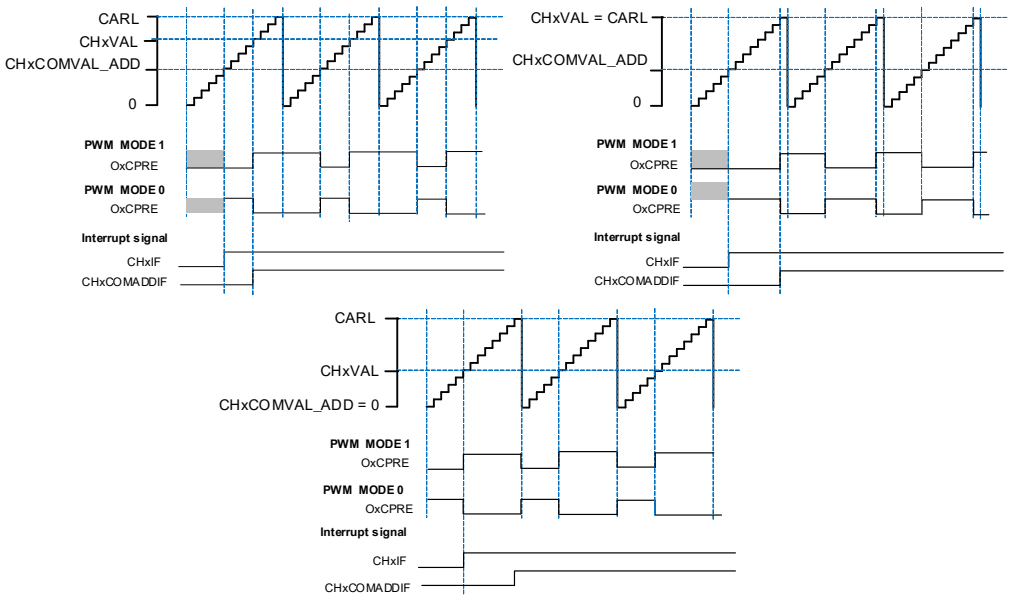
Figure 23-101. Channel x output PWM with (CHxVAL = CHxCOMVAL_ADD)



3) CHxVAL > CHxCOMVAL_ADD, and the value of CHxVAL and CHxCOMVAL_ADD

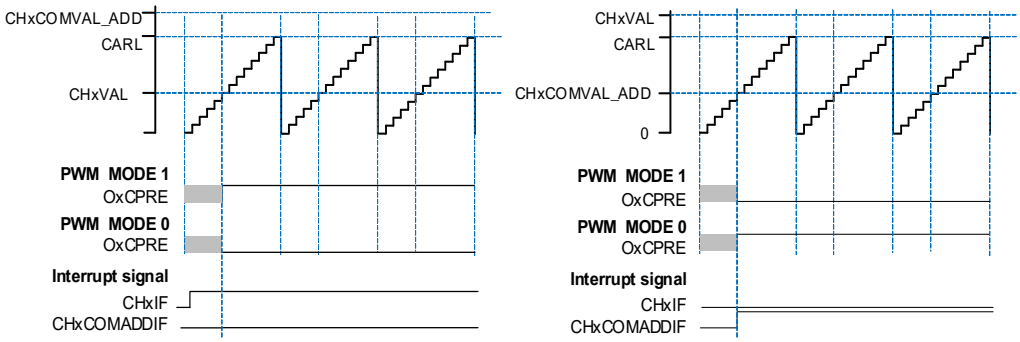
between 0 and CARL.

Figure 23-102. Channel x output PWM with (CHxVAL > CHxCOMVAL_ADD)



4) One of the value of CHxVAL and CHxCOMVAL_ADD exceeds CARL.

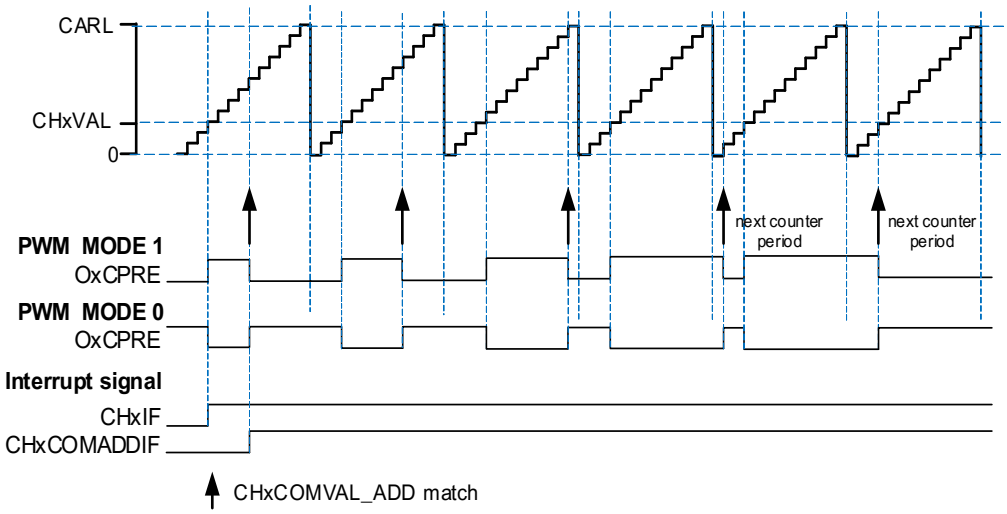
Figure 23-103. Channel x output PWM with CHxVAL or CHxCOMVAL_ADD exceeds CARL



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 23-104. 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 23-104. 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 23-105. CHx_O output with a pulse in edge-aligned mode (CHxOMPSEL!=2'b00)

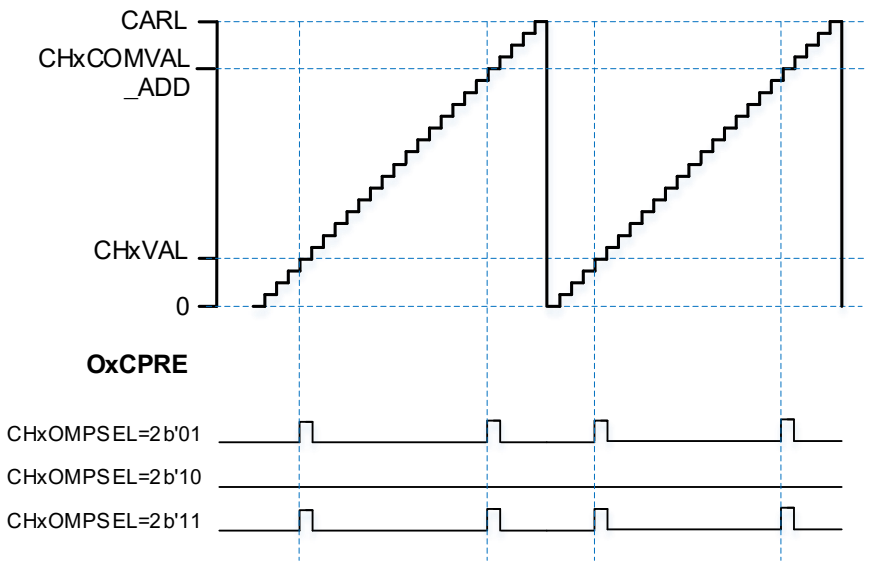
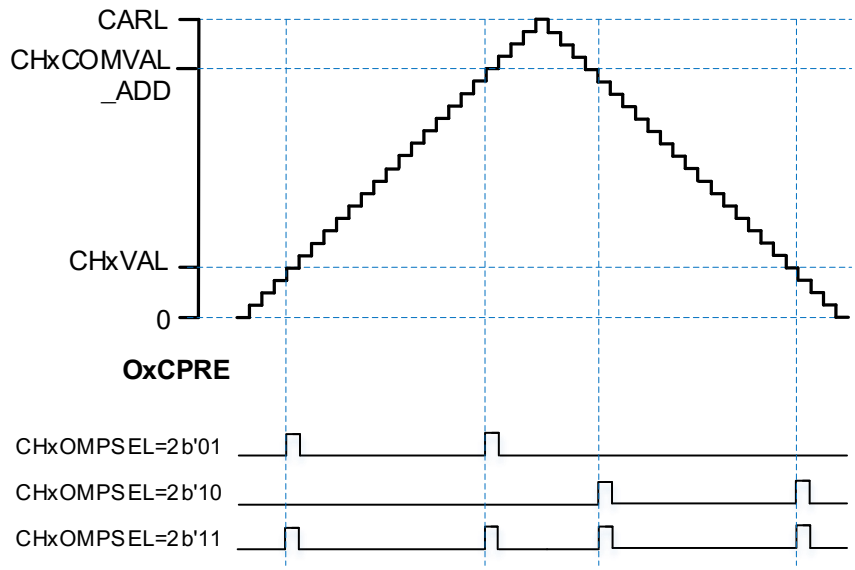


Figure 23-106. CHx_O output with a pulse in center-aligned mode (CHxOMPSEL!=2'b00)



Channel output prepare signal

As is shown in *Figure 23-94. Output compare logic (when MCHxMSEL = 2'00, x=0)*, *Figure 23-95. Output compare logic (when MCHxMSEL = 2'11, x=0)* and *Figure 23-96. 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 23-14. Complementary outputs controlled by parameters (MCHxMSEL =2'b11).***

Table 23-14. 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

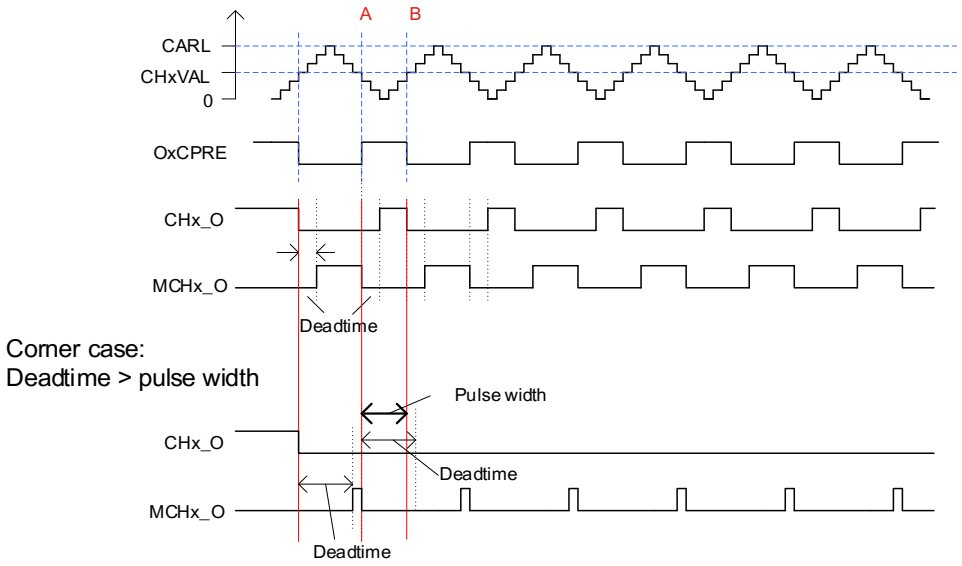
The dead time insertion is enabled when MCHxMSEL=2'b11 and both CHxEN and MCHxEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels. Refer to the *Complementary channel* protection register 0 (TIMERx_CCHP0) 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 23-107. 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 23-107. Complementary output with dead-time insertion***).

Figure 23-107. 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 BREAK0 function. The BREAK0 function can be enabled by setting the BRK0EN bit in the TIMERx_CCHP register. The break input polarity is configured by the BRK0P bit in TIMERx_CCHP register, the input is active on level.

In BREAK0 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.

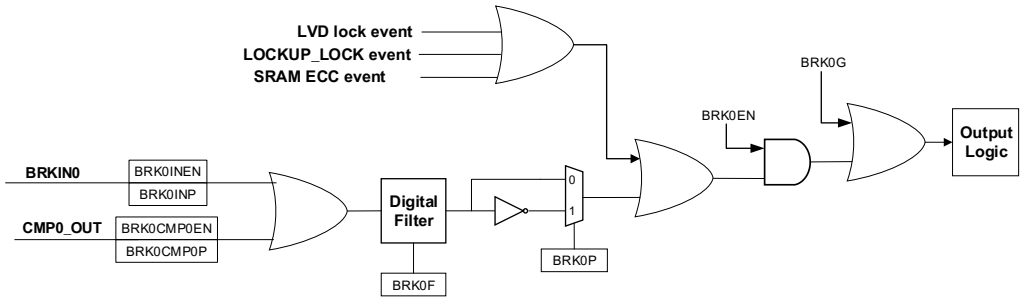
The break event is the result of logic ORed of all sources. The BREAK0 function can handle three types of event sources:

- 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.

BREAK0 events can also be generated by software using BRK0G bit in the TIMERx_SWEVG register.

Refer to **Figure 23-108. BREAK0 function logic diagram**, BRKIN0 can select GPIO pins from the TRIGSEL module.

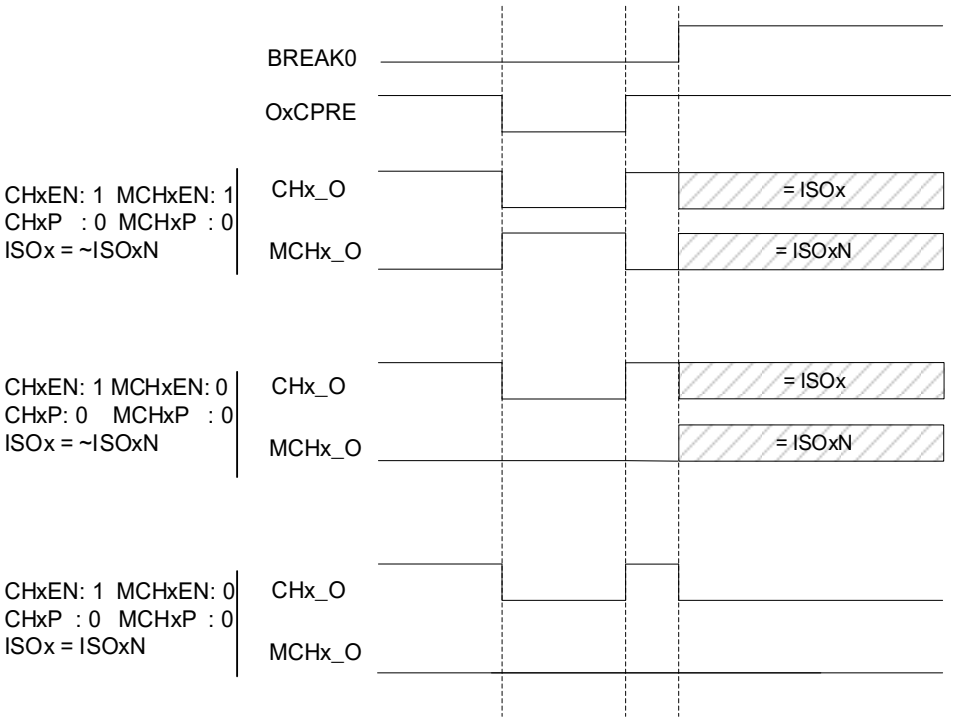
Figure 23-108. BREAK0 function logic diagram



When a BREAK0 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 23-109.* **Output behavior of the channel in response to BREAK0 (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 23-109. Output behavior of the channel in response to BREAK0 (the break input high active and IOS=1)

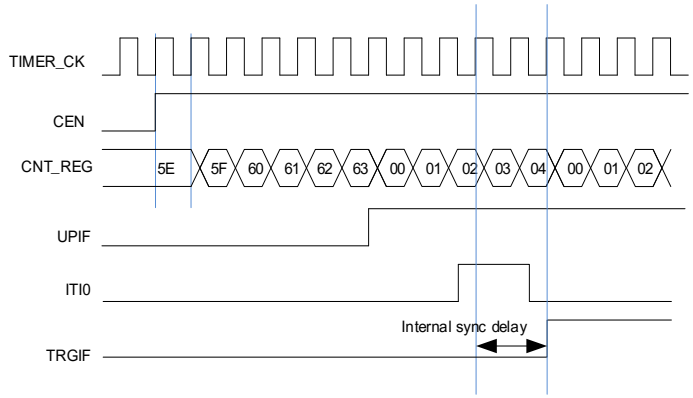


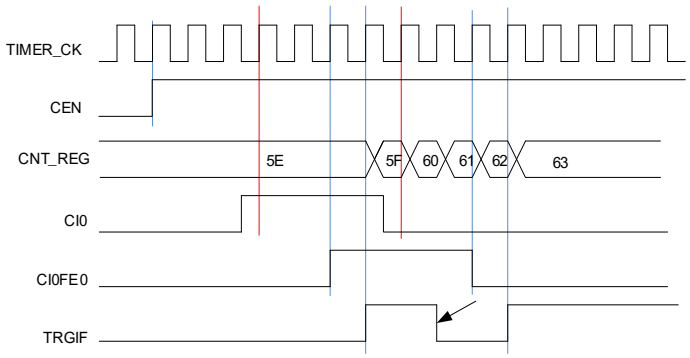
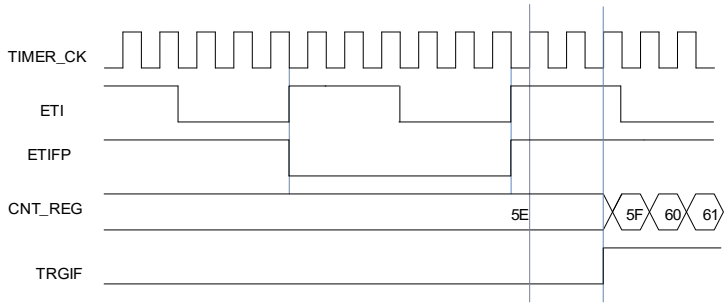
When a break occurs, the BRK0IF bit in the TIMERx_INTF register will be set. If BRKIE is 1, an interrupt will be generated.

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 23-15. 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.
	Figure 23-110. 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.

<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td></td><td colspan="4">Figure 23-111. Pause mode

</td></tr>
<tr><td rowspan="2">Exam3</td><td>Event mode
The counter will start to count when a rising edge of trigger input comes.</td><td>TSCFG5[4:0] =5'b01000, ETIFP is selected.</td><td>ETP = 0, the polarity of ETI does not change.</td><td>ETPSC = 1, ETI is divided by 2.
ETFC = 0, ETI does not filter.</td></tr>
<tr><td colspan="4">Figure 23-112. Event mode

</td></tr>
</table>

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.

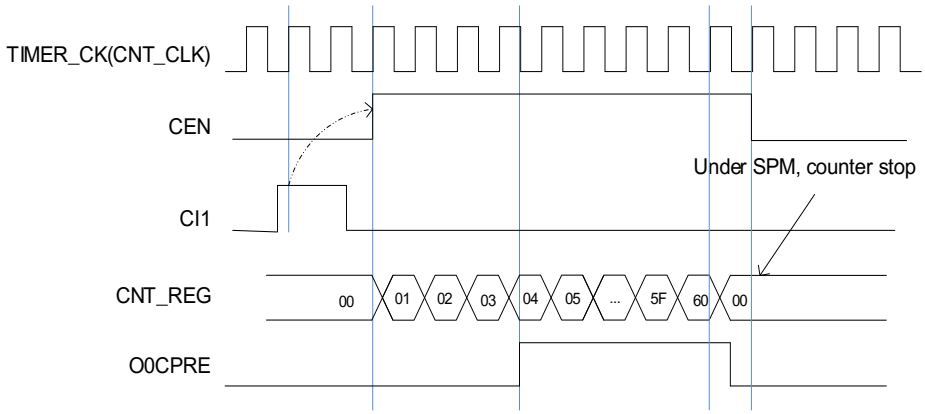
Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs 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 23-113. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timers interconnection

Please refer to **Advanced timer** (TIMERx, x=0, 7, 19)**Timers interconnection**.

Counter synchronization and counter initial direction and value refresh

Please refer to **Advanced timer** (TIMERx, x=0, 7, 19)*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.

23.4.5. Register definition (TIMERx, x=14,40,41,42,43,44)

TIMER14 base address: 0x4001 4000

TIMER40 base address: 0x4001 4C00

TIMER41 base address: 0x4001 4400

TIMER42 base address: 0x4000 E000

TIMER43 base address: 0x4000 E400

TIMER44 base address: 0x4000 E800

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				MMC0[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	MMC0[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	MMC0[3]	Master mode control 0 Refer to MMC0[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 MMC0[2:0] Master mode control 0
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	BRK0IF	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	BRK0IF	BREAK0 interrupt flag This flag is set by hardware when the break input is active, and cleared by software if the BREAK0 input is not at active level. 0: No active level on BREAK0 input has been detected. 1: An active level on BREAK0 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								BRK0G	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	BRK0G	BREAK0 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 BRK0IF flag will be set, related interrupt can occur if enabled. 0: No generate a BREAK0 event 1: Generate a BREAK0 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).

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
Reserved	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		Reserved	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	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	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 When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. 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

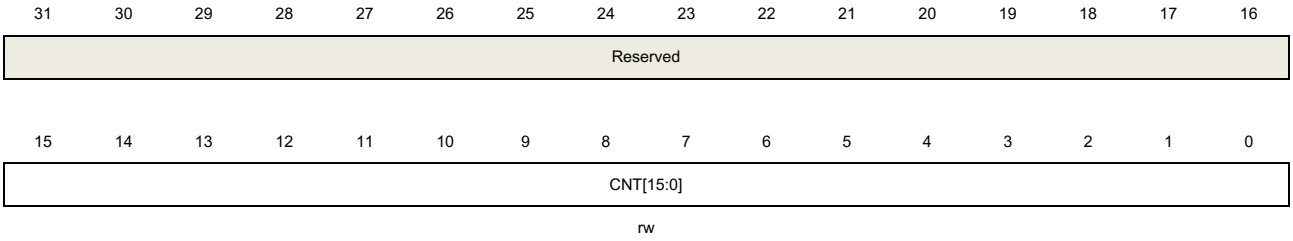
		Refer to CH0P description.
4	CH1EN	Channel 1 capture/compare enable Refer to CH0EN description.
3	MCH0P	Multi mode channel 0 output polarity When Multi mode channel 0 is configured in output mode, and the MCH0MSEL[1:0] = 2'b11, this bit specifies the CH0_ON output signal polarity. 0: Multi mode channel 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).



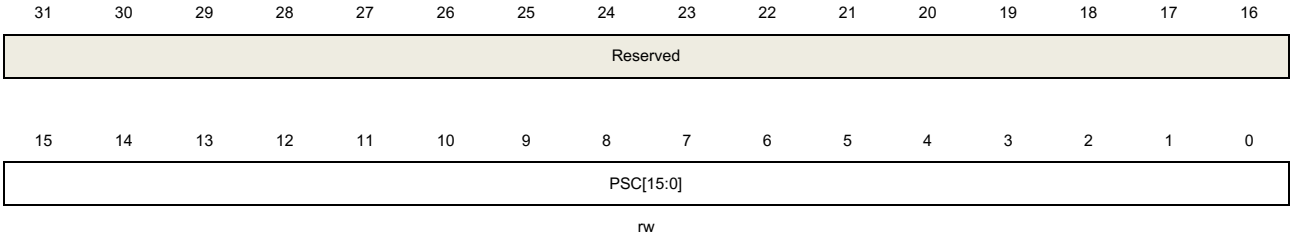
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-field indicates the current counter value. Writing to this bit-field can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

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



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.

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												BRK0F[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRK0P	BRK0EN	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	BRK0F[3:0]	BREAK0 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 BREAK0 input signal and the length of the digital filter applied to BREAK0. 0000: Filter disabled. BREAK0 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	BRK0P	BREAK0 input signal polarity This bit specifies the polarity of the BREAK0 input signal. 0: BREAK0 input active low 1: BREAK0 input active high This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
12	BRK0EN	BREAK0 input signal enable This bit can be set to enable the BREAK0 input signal 0: BREAK0 input disabled 1: BREAK0 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 23-14. 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 23-14. 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 BRK0EN/ BRK0P/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

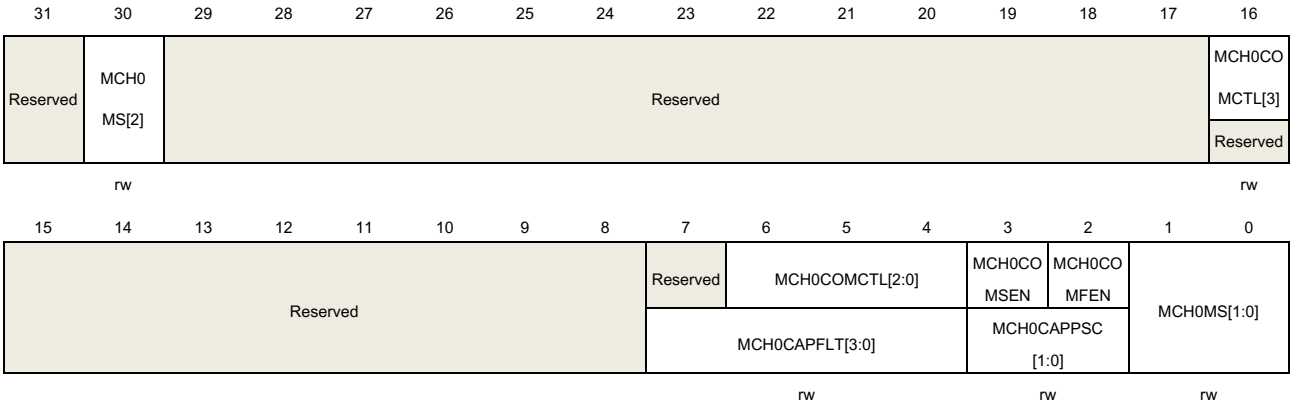
		bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) the ROS/IOS bits in TIMERx_CCHP0 register are writing protected. 11: PROT mode 2. In addition to the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN bits in TIMERx_CHCTL0 and TIMERx_MCHCTL0 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 When this bit is set, the effect of an event on the trigger in input on the capture / compare output will be accelerated if the channel is configured in PWM 0 or PWM 1 mode. The output channel will treat an active edge on the trigger input as a compare match, and MCH0_O is set to the compare level independently from the result of the comparison. 0: Multi mode channel 0 output quickly compare disable. 1: Multi mode channel 0 output quickly compare enable.
1:0	MCH0MS[1:0]	Multi mode channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active (MCH0EN bit in TIMERx_CHCTL2 register is reset). 000: Multi mode channel 0 is 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 [15:0]	Additional compare value of channel 0 When channel 0 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1).

Channel 1 additional compare value register (TIMERx_CH1COMV_ADD)

Address offset: 0x68

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMVAL_ADD[15:0]															
rw															

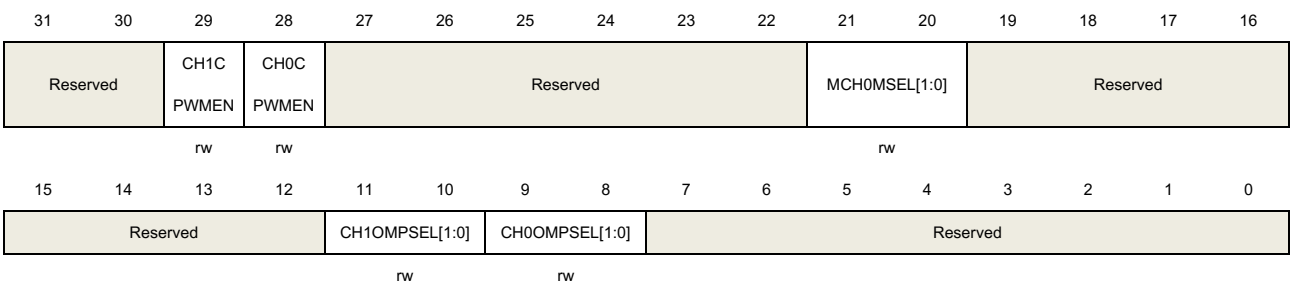
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH1COMVAL_ADD [15:0]	Additional compare value of channel 1 When channel 1 is configured in output mode, this bit-field contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates by every update event. **Note:** This register just used in composite PWM mode(when CH0CPWMEN=1).

Control register 2 (TIMERx_CTL2)

Address offset: 0x74

Reset value: 0x0030 0000

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



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.

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					BRK0CMP1P	BRK0CMP0P	Reserved								BRK0IN0P
					rw	rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK0CMP1EN	BRK0CMP0EN	Reserved								BRK0IN0EN
					rw	rw									rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	BRK0CMP1P	BREAK0 CMP1 input polarity This bit is used to configure the CMP1 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =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.
25	BRK0CMP0P	BREAK0 CMP0 input polarity This bit is used to configure the CMP0 input polarity, and the specific polarity is determined by this bit and the BRK0P bit. 0: CMP0 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: CMP0 input signal will be inverted (BRK0P=0, the input signal is active high;

		BRK0P =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	BRK0IN0P	BREAK0 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 BRK0P bit. 0: BRKIN0 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =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.
15:11	Reserved	Must be kept at reset value.
10	BRK0CMP1EN	BREAK0 CMP1 enable 0: CMP1 input disabled 1: CMP1 input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP0 register is 00.
9	BRK0CMP0EN	BREAK0 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	BRK0IN0EN	BREAK0 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).

31	30	29	28	27	26	25	24	23	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 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.

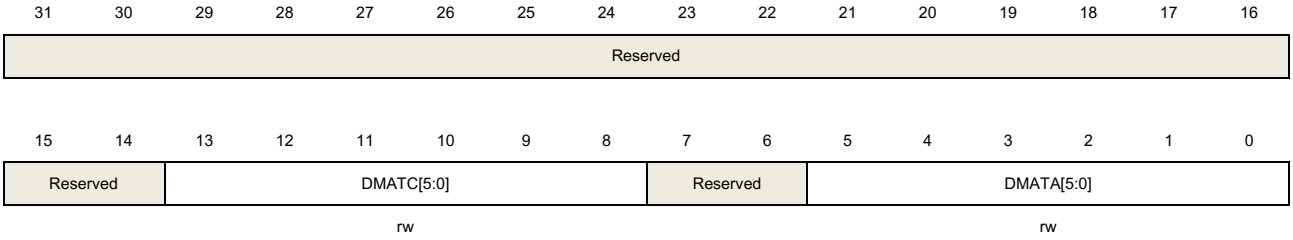
When the CINITVEN bit is 1, when the synchronization event occurs, the counter register can load the initial value which are set in CINITVAL bits-field.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0xE0

Reset value: 0x0000 0000

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



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 … 6'b111111: TIMERx_DMACFG 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.

23.5. Basic timer (TIMERx, x=5,6,50,51)

23.5.1. Overview

The basic timer module(TIMER5/6/50/51) 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.

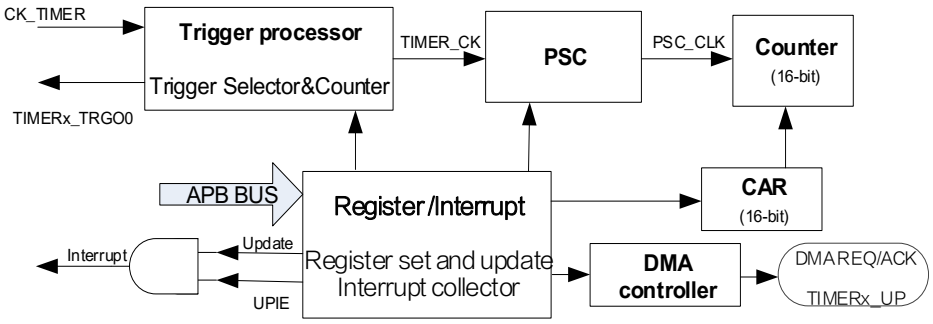
23.5.2. Characteristics

- 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.

23.5.3. Block diagram

Figure 23-114. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 23-114. Basic timer block diagram



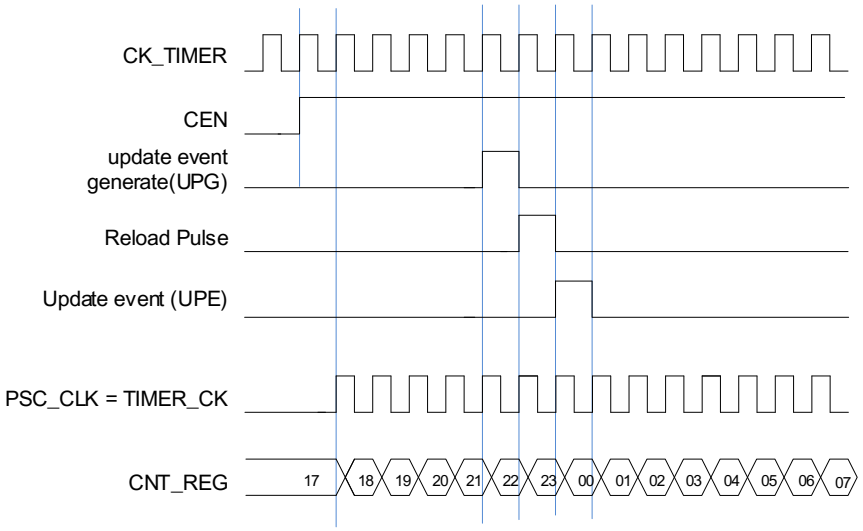
23.5.4. Function overview

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.

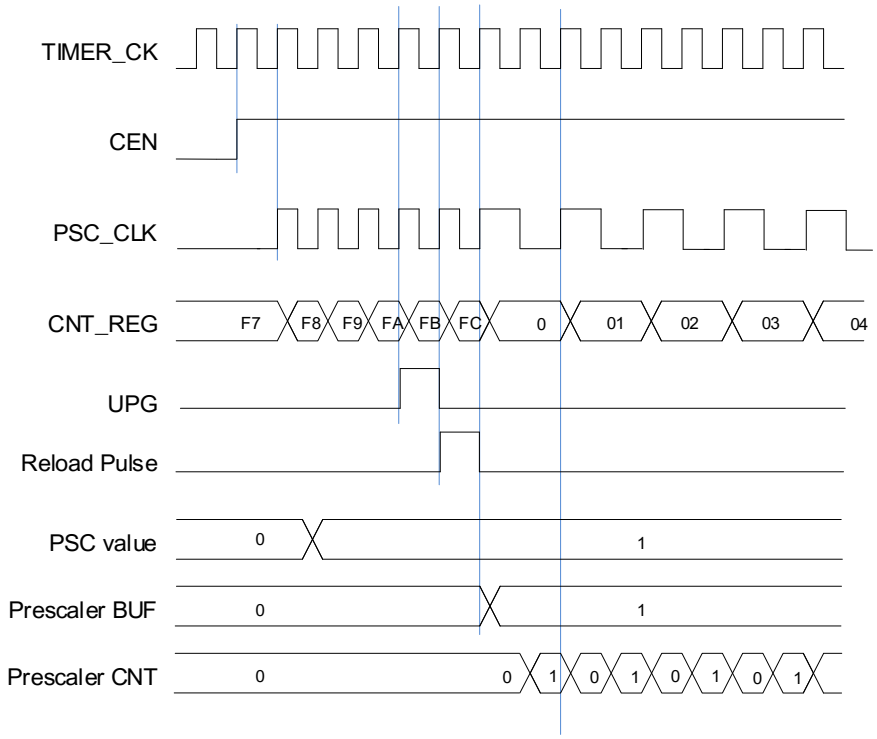
Figure 23-115. Normal mode, internal clock divided by 1



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 23-116. 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 23-117. Timing chart of up counting mode, PSC=0/2

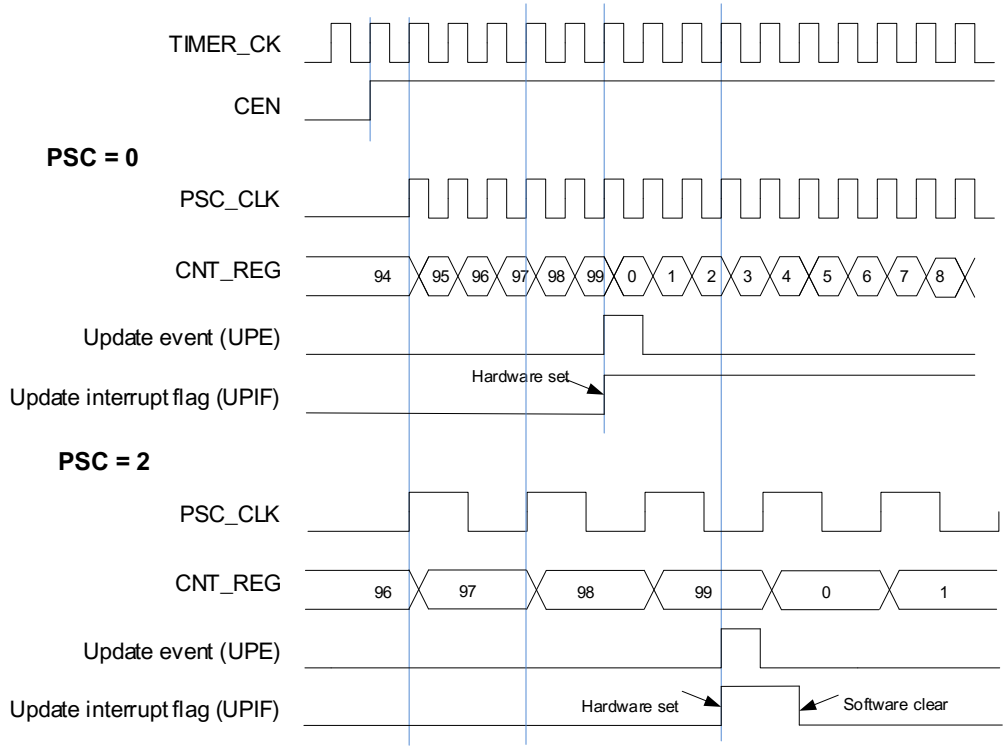
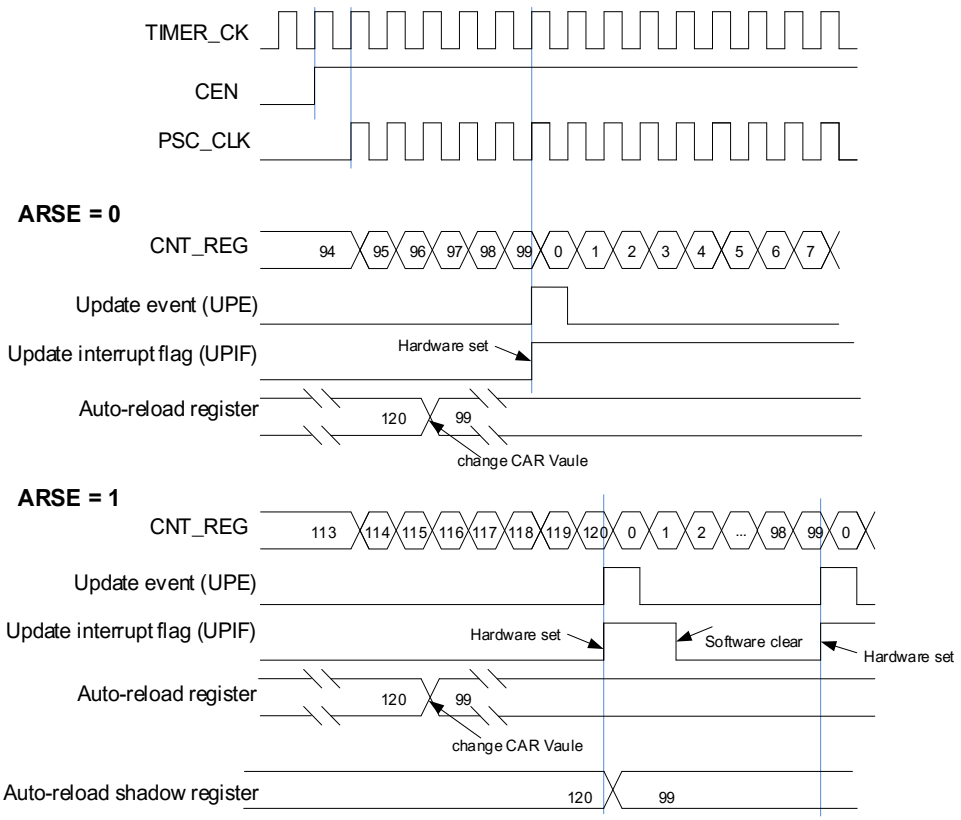


Figure 23-118. Timing chart of up counting mode, change TIMERx_CAR ongoing



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.

23.5.5. Registers definition (TIMERx, x=5,6,50,51)

TIMER5 base address: 0x4000 1000

TIMER6 base address: 0x4000 1400

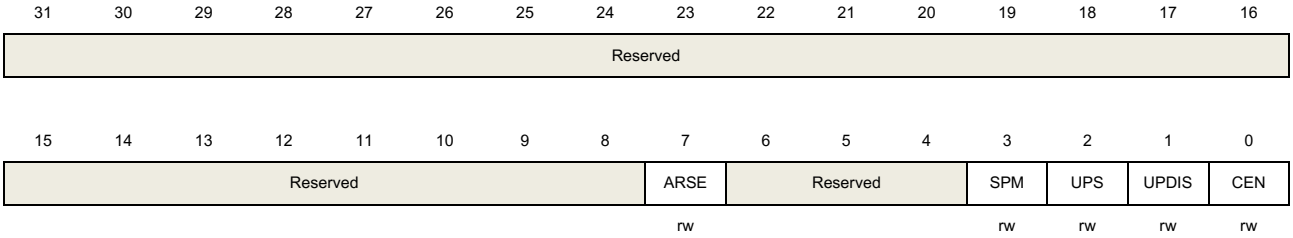
TIMER50 base address: 0x4000 EC00

TIMER51 base address: 0x4000 F000

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Single pulse mode is disabled. Counter continues after an update event. 1: Single pulse mode is enabled. The 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).

31	30	29	28	27	26	25	24	23	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									MMC0[2:0]			Reserved			
									rw						

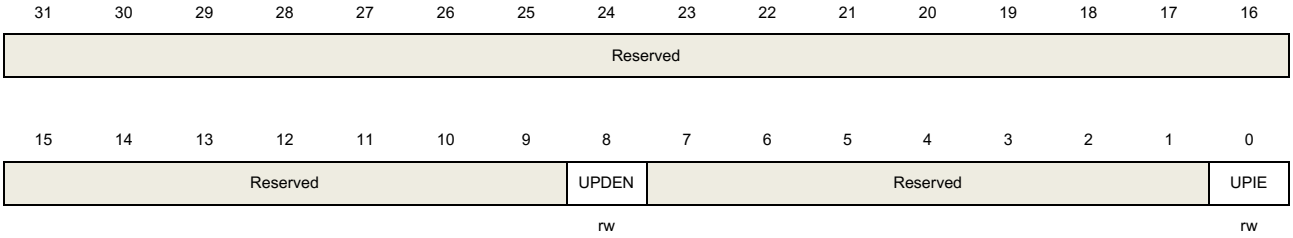
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	MMC0[2:0]	Master mode control 0 These bits control the selection of TRGO signal, which is sent by master timer to slave timer for synchronization function. 000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset 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).

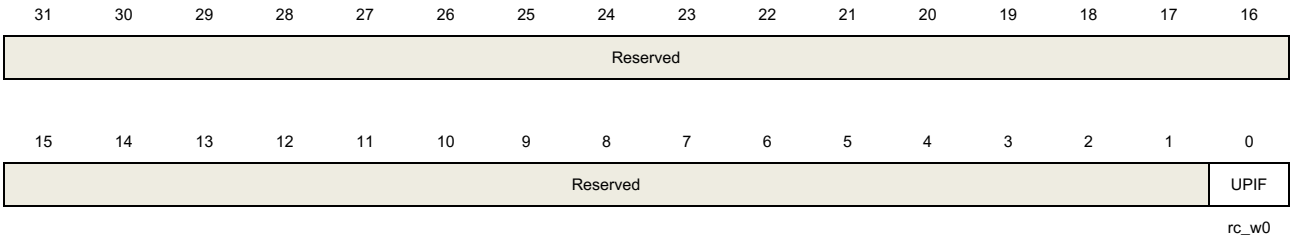


Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	UPDEN	Update DMA request enable 0: Disabled 1: Enabled
7:1	Reserved	Must be kept at reset value.
0	UPIE	Update interrupt enable 0: Disabled 1: Enabled

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

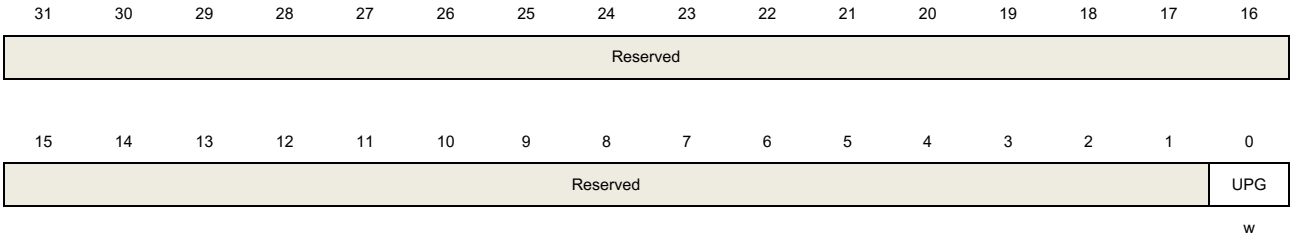
31:1	Reserved	Must be kept at reset value.
0	UPIF	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

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



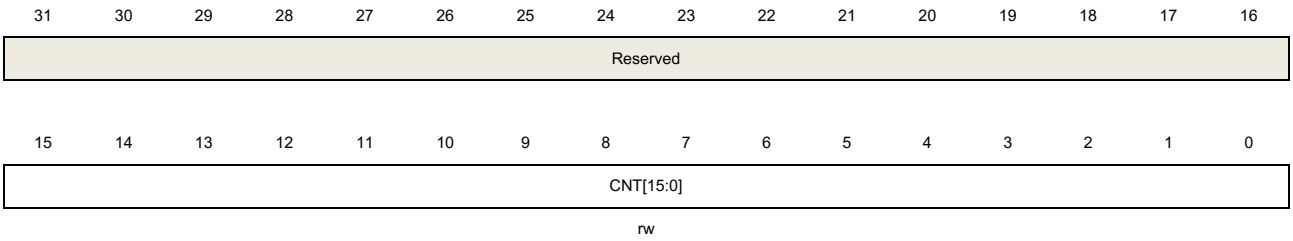
Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	UPG	This bit can be set by software, and automatically cleared by hardware. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

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).

31	30	29	28	27	26	25	24	23	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).

31	30	29	28	27	26	25	24	23	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.

24. Universal synchronous / asynchronous receiver / transmitter (USART)

24.1. Universal synchronous / asynchronous receiver / transmitter (USARTx/UARTx, x=0..4)

24.1.1. Overview

The 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 (CK_APB1 or CK_APB2) 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.

All USART/UART supports DMA function for high-speed data communication(except for UART4).

24.1.2. Characteristics

- NRZ standard format.
- Asynchronous, full duplex communication.
- Half duplex single wire communications.
- Programmable baud-rate generator.
  - Divided from the peripheral clocks, CK_APB2 for USART0, CK_APB1 for USART1/2 and UART3/4.
  - Oversampling by 16 or 8.
  - Maximum speed up to 35 MBits/s (CK_APB2 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.
- Serial port TX can be configured as infrared modulation output, with adjustable modulation signal duty cycle and selectable modulation polarity (only supported by USART2).
- 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.

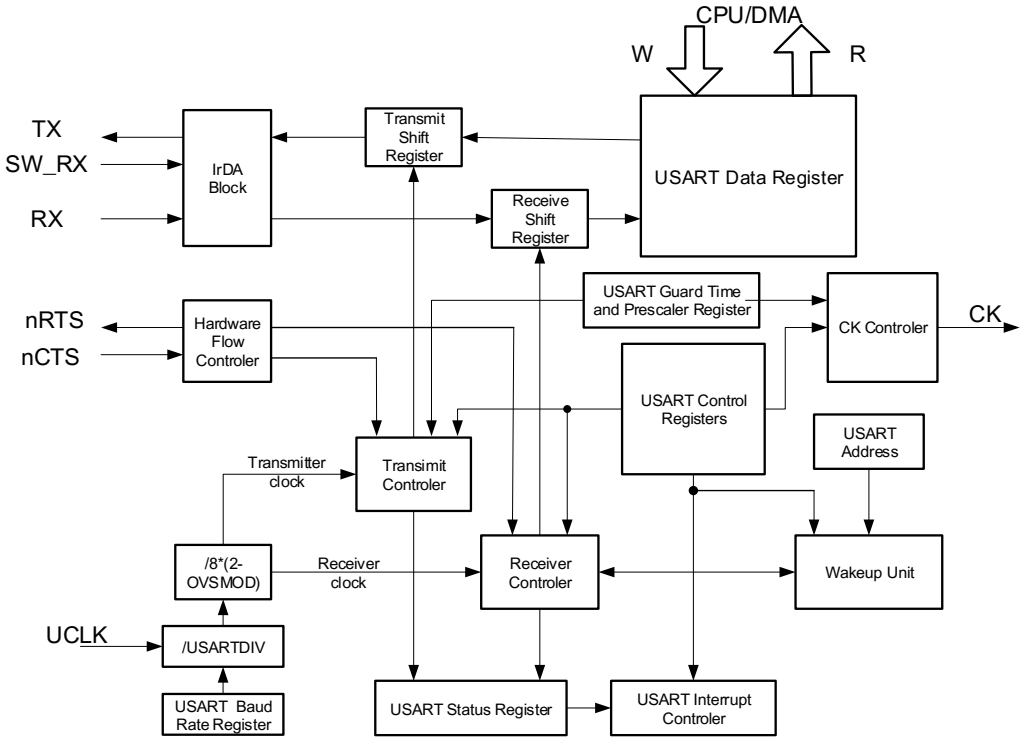
24.1.3. Function overview

The interface is externally connected to another device by the main pins listed in ***Table 24-1. Description of USART important pins***.

Table 24-1. 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

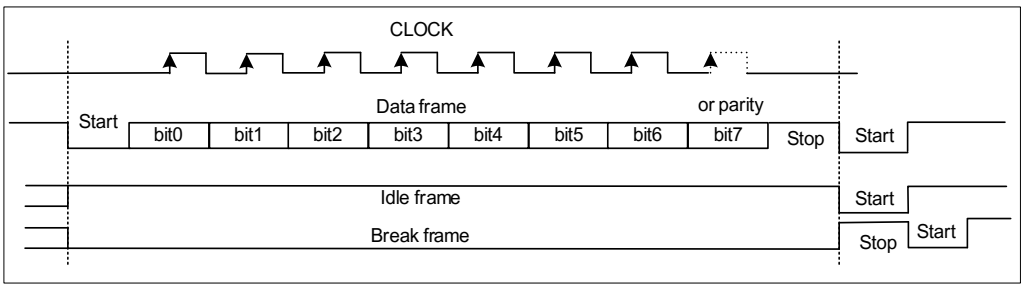
Figure 24-1. USART module block diagram



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 24-2. 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 24-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	Default value
01	0.5	Smartcard mode for receiving

STB[1:0]	stop bit length (bit)	usage description
10	2	Normal USART and single-wire modes
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK, the configuration of the baud rate generator and the oversampling mode.

Baud rate generation

The baud-rate divider is a 16-bit number which consists of a 12-bit integer and a 4-bit fractional part. The number formed by these two values is used by the baud rate generator to determine the bit period. Having a fractional baud-rate divider allows the USART to generate all the standard baud rates.

The baud-rate divider (USARTDIV) has the following relationship with the peripheral clock:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (24\text{-}1)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (24\text{-}2)$$

For example, when oversampled by 16:

1. Get USARTDIV by calculating 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.

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.

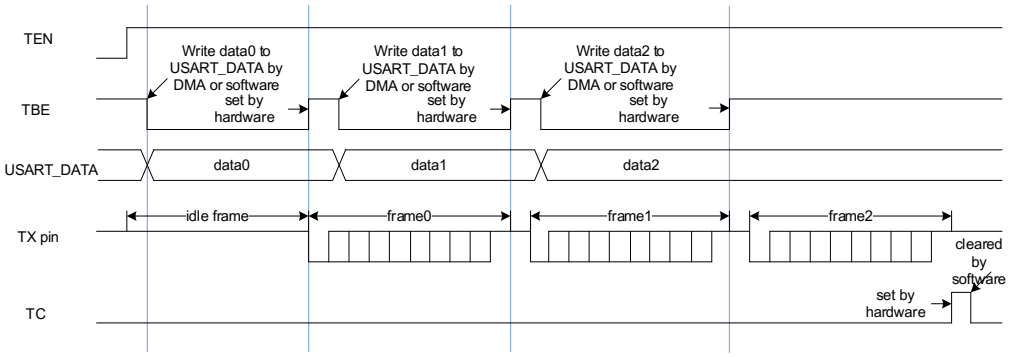
After power on, the TBE bit is high by default. Data can be written to the USART_DATA when the TBE bit in the USART_STAT0 register is asserted. The TBE bit is cleared by writing to the USART_DATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written 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 24-3. 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 24-3. 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.

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 to configure the number of stop bits.
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.

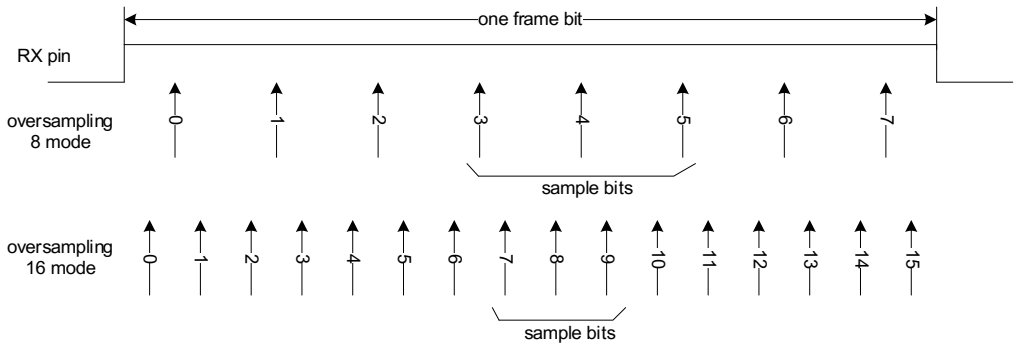
The software can get the received data by reading the USART_DATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_DATA register, whatever it is performed by software directly, or through DMA.

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 two or more samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the value of the three samples of any bit are not the same, whatever it is a data bit, parity bit or stop bit, a noisy error (NERR) 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.

The one bit receive mode can be enabled by setting the OSB bit in USART_CTL2. This mode can be enabled when the RX line is noise free and the NERR bit will never be set in this mode.

Figure 24-4. Receiving a frame bit by oversampling method



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.

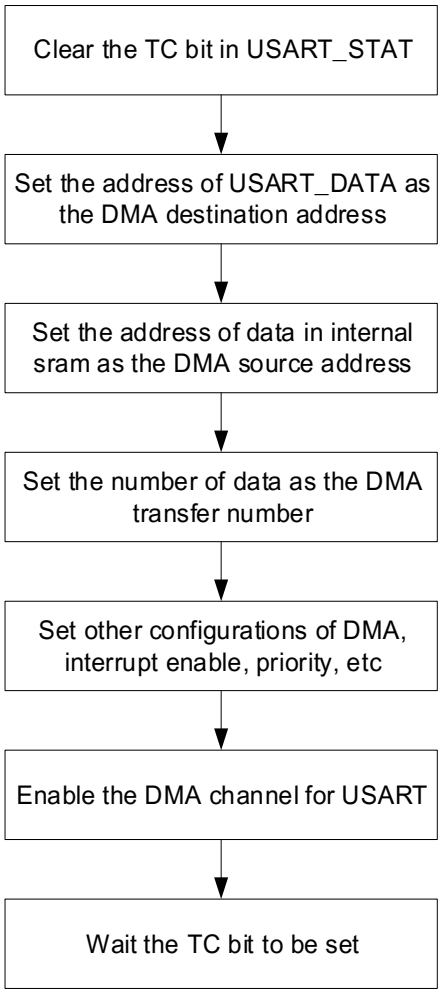
If a noise error (NERR), parity error (PERR), frame error (FERR) or overrun error (ORERR) is generated during a receiving process, then NERR, PERR, FERR or ORERR will be set at same time with RBNE. If DMA is disabled, the software needs to check whether the RBNE interrupt is caused by noise error, parity error, framing error or overflow error when the RBNE interrupt occurs.

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 24-5. Configuration step when using DMA for USART transmission***.

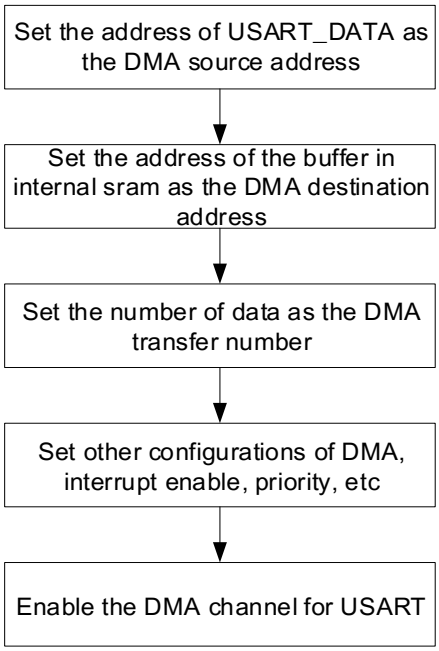
Figure 24-5. Configuration step when using DMA for USART transmission



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 24-6. 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 24-6. Configuration steps when using DMA for USART reception

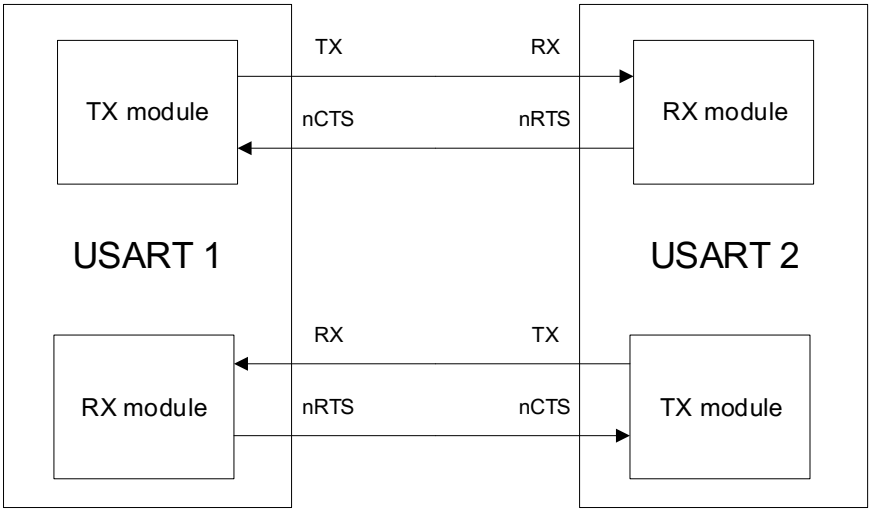


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt will be generated in the DMA module.

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 24-7. Hardware flow control between two USARTs



RTS flow control

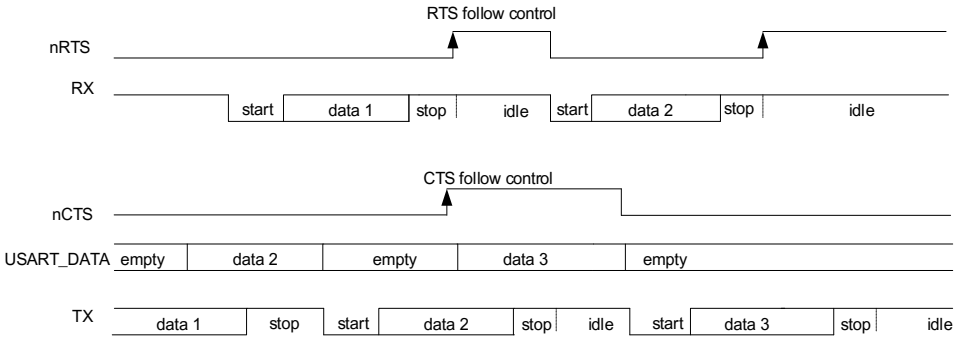
The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When

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 24-8. 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.

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 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, the receiver does not check the parity value of an address frame by default. If the PCEN bit in USART_CTL0 is set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address flag.

LIN mode

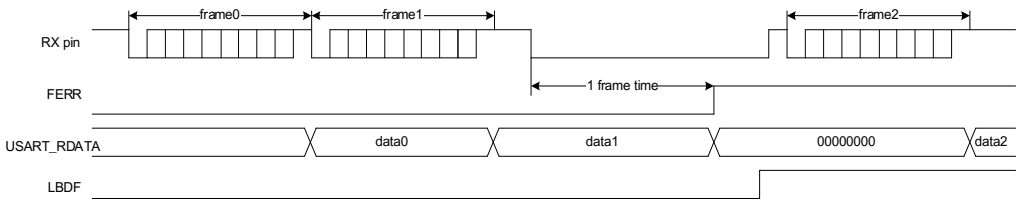
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, WL, STB[1:0] bits in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

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.

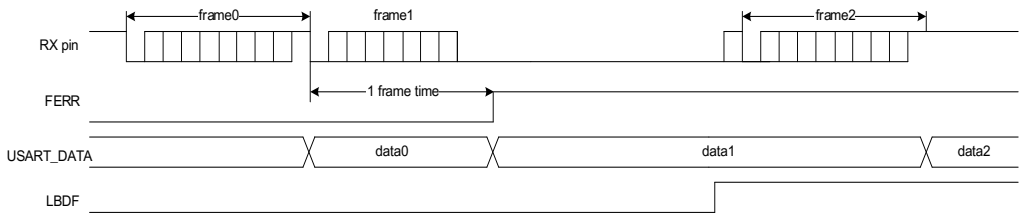
As shown in ***Figure 24-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 24-9. Break frame occurs during idle state



As shown in ***Figure 24-10. 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 24-10. Break frame occurs during a frame



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 24-11. Example of USART in synchronous mode

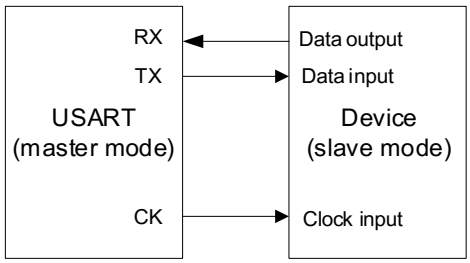
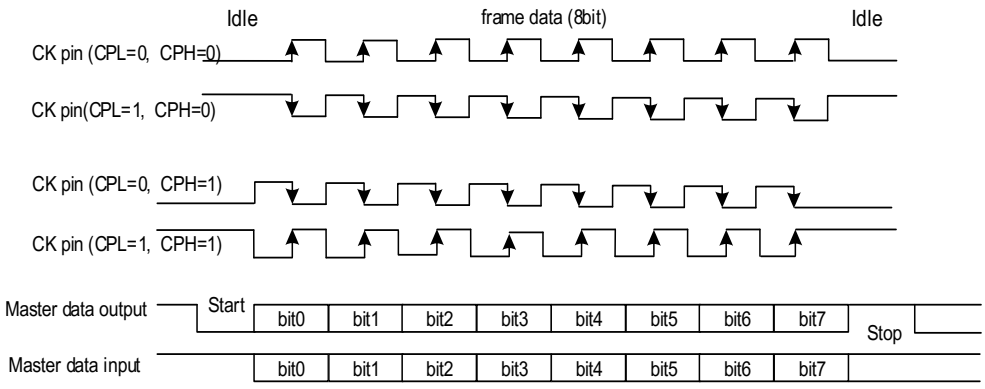


Figure 24-12. 8-bit format USART synchronous waveform (CLEN=1)



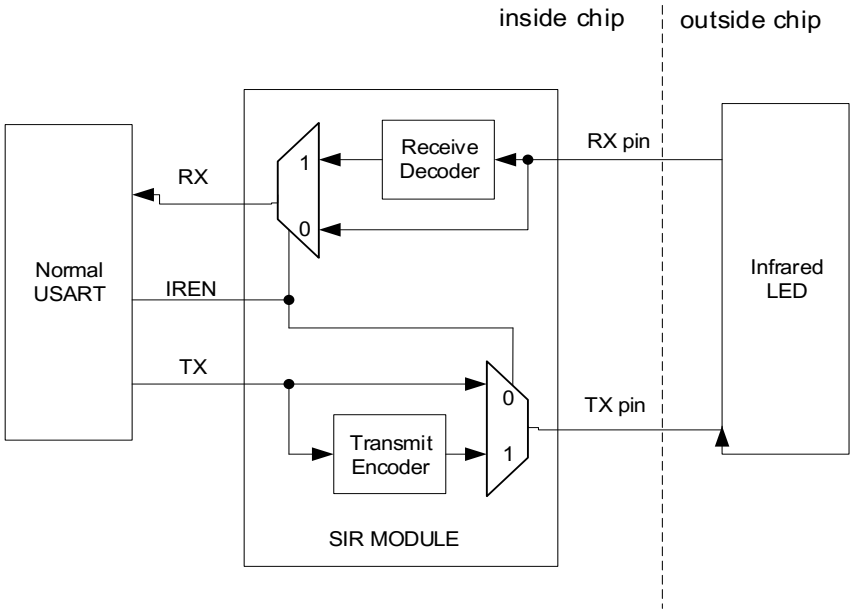
IrDA SIR ENDEC mode

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.

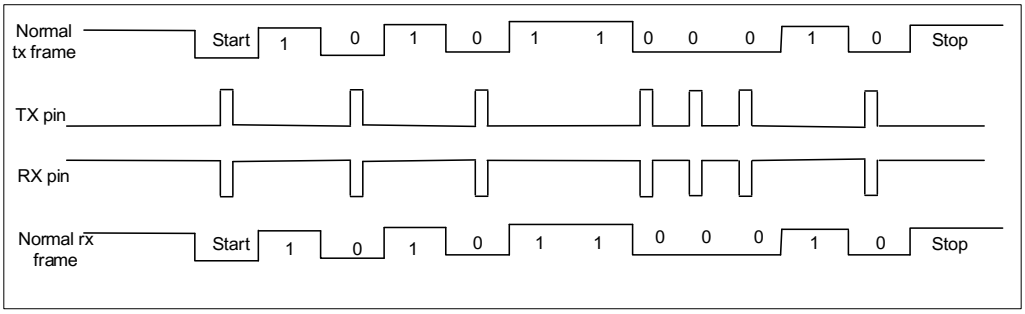
Figure 24-13. IrDA SIR ENDEC module



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 24-14. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The

division factor is configured in bits PSC[7:0] of the USART_GP register, and the USART_BAUD register must be set to 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.

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 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.

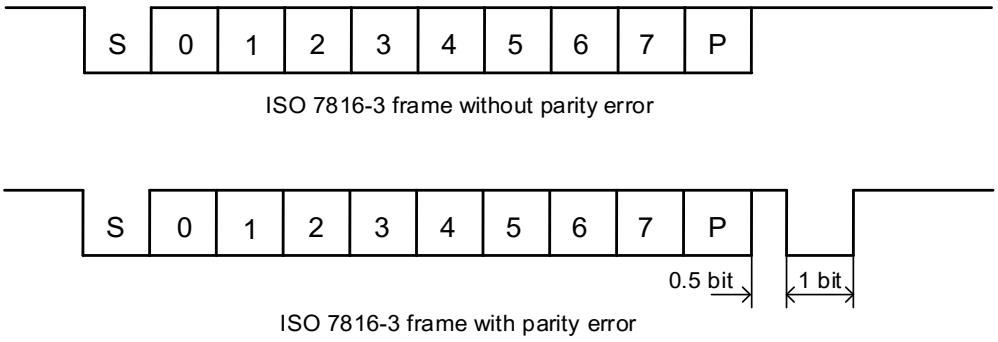
Smartcard (ISO7816-3) mode

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 24-15. ISO7816-3 frame format



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.

USART interrupts

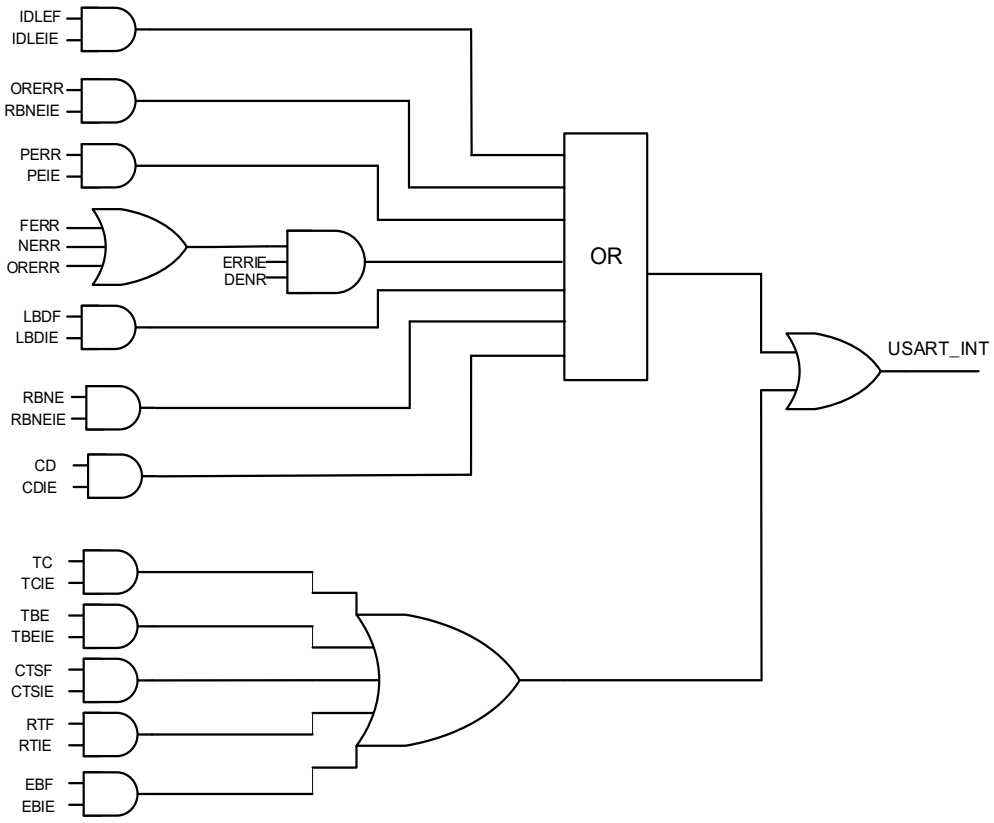
The USART interrupt events and flags are listed in ***Table 24-3. USART interrupt requests***.

Table 24-3. 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 24-16. USART interrupt mapping diagram



Infrared modulation

Infrared modulation is a technology that enables infrared transmission by modulating data signals onto a carrier wave. The TX of USART2 supports infrared modulation output functionality, which can be activated by configuring the IMEN (Infrared Modulation Enable) bit. When using the infrared modulation function of USART2, the polarity of the modulated signal can be flexibly selected, and the duty cycle of the infrared modulation signal can be adjusted according to actual requirements.

Note: When oversampled by 8, the baud rate configuration must satisfy:

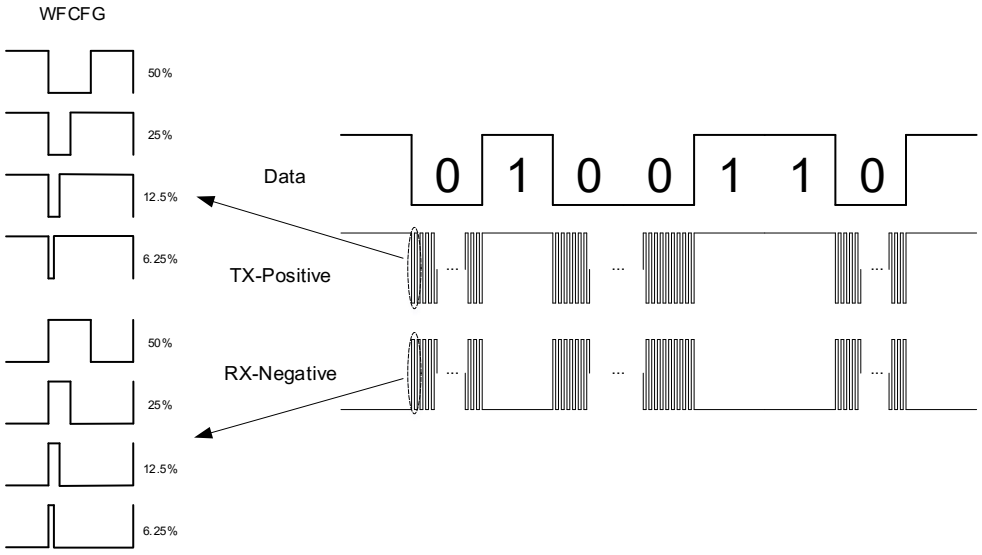
$$USART_BAUD >= CFDIV * 16 * 4 \quad (24\text{-}3)$$

When oversampled by 16:

$$USART_BAUD >= CFDIV * 16 * 2 \quad (24\text{-}4)$$

Otherwise, it may fail to output an effective number within one carrier.

Figure 24-17. Infrared modulation waveform schematic diagram



24.1.4. Register definition

USART0 base address: 0x4001 3800

USART1 base address: 0x4000 4400

USART2 base address: 0x4000 4800

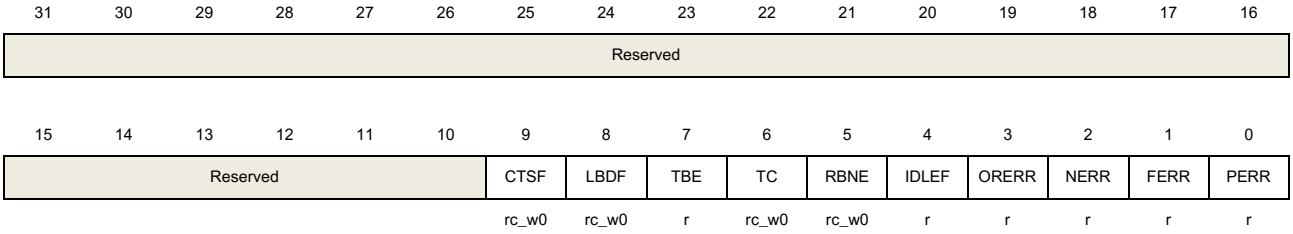
UART3 base address: 0x4000 4C00

UART4 base address: 0x4000 5000

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 reserved 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 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

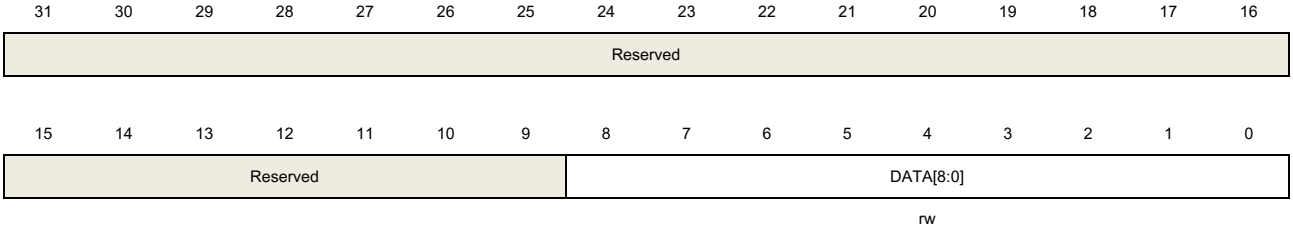
		frame. An interrupt occurs if the ERRIE bit in USART_CTL2 is set. Software can clear this bit by reading the USART_STAT0 and USART_DATA registers one by one. 0: The USART does not detect a framing error. 1: The USART has detected a framing error.
0	PERR	Parity error flag This bit is set when the parity 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.

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.

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]
	rw

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.

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	Oversampe 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.

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 bits 10: 2 Stop bits 11: 1.5 Stop bits 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 disabled. 1: LIN break detected interrupt is enabled.
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.

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. 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.

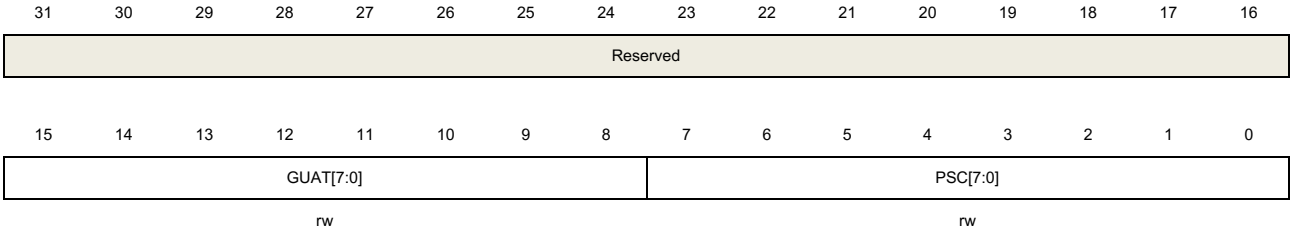
		This bit is reserved for UART3/4.
3	HDEN	Half-duplex enable This bit enables the half-duplex USART mode. 0: Half duplex mode is disabled. 1: Half duplex mode is enabled.
2	IRLP	IrDA low-power This bit selects low-power mode of IrDA mode. 0: Normal mode. 1: Low-power mode.
1	IREN	IrDA mode enable This 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.

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 reserved 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 (CK_APB1/CK_APB2) to generate the low-power frequency.

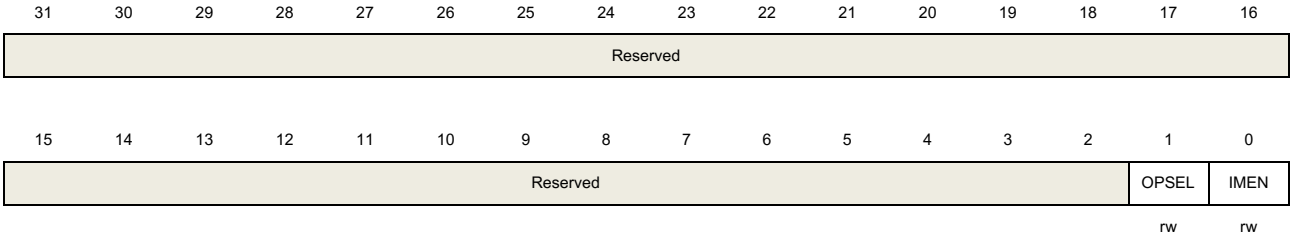
00000000: Reserved - never program this value
00000001: divides by 1
00000010: divides by 2
...
11111111: divides by 255
When the USART works in IrDA normal mode, these bits must be set to 00000001.
When the USART smartcard mode is enabled, the PSC [4:0] bits specify the division factor that is used 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.

Infrared modulation control register (USART_IMCTL)

Address offset: 0x20
Reset value: 0x0000 0000

This register is only available for USART2.

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept the reset value.
1	OPSEL	Infrared modulation output polarity selection 0: Positive polarity 1: Negative polarity
0	IMEN	Infrared modulation function enable 0: Disable infrared modulation function of TX output. 1: Enable infrared modulation function of TX output.

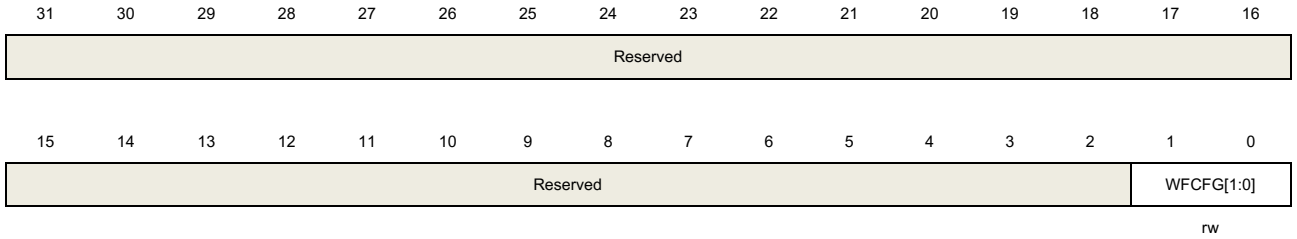
Infrared modulation pulse width register (USART_IMPW)

Address offset: 0x24

Reset value: 0x0000 0000

This register is only available for USART2.

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



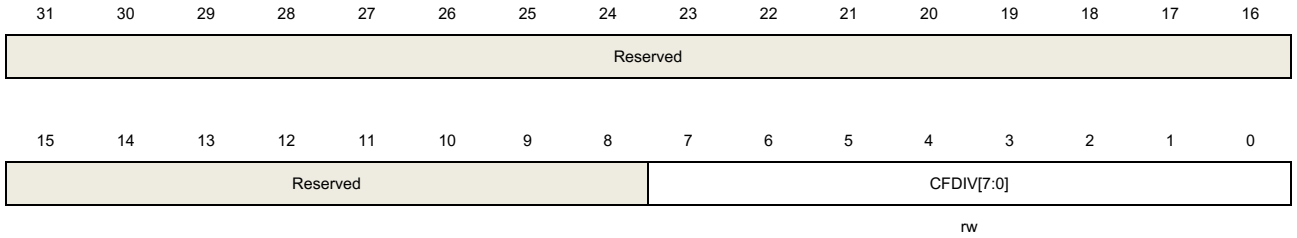
Bits	Fields	Descriptions
31:2	Reserved	Must be kept the reset value.
1:0	WFCFG[1:0]	Infrared modulation waveform duty cycle configuration 00: 50% 01: 25% 10: 12.5% 11: 6.25%

Infrared modulation prescaler register (USART_IMPRE)

Address offset: 0x28
Reset value: 0x0000 0000

This register is only available for USART2.

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



Bits	Fields	Descriptions
31:8	Reserved	Must be kept the reset value.
7:0	CFDIV[7:0]	Infrared modulation clock frequency division configuration Infrared modulation signal frequency is: $F = \frac{UCLK}{(CFDIV[7:0] + 1) * 16}$

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 USARTx_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.

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

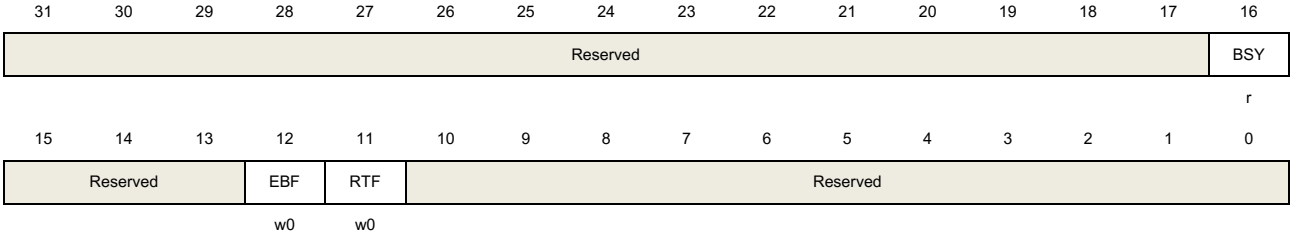
Status register 1 (USART_STAT1)

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

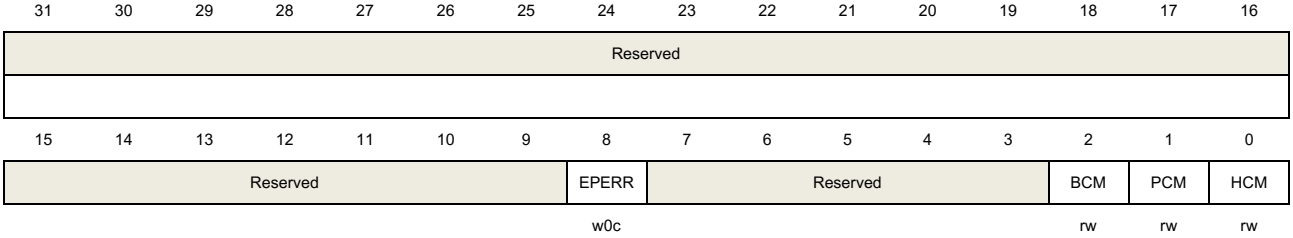
		This bit is set when the RX pin is in idle state for longer than RT bits time. An interrupt occurs if the RTIE bit in USART_CTL3 is set. Software can clear this bit by writing 0 to it. 0: Receiver timeout event does not occur. 1: Receiver timeout event has occurred.
10:0	Reserved	Must be kept the reset value.

Coherence control register (USART_CHC)

Address offset: 0xC0

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	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.

24.2. Universal synchronous / asynchronous receiver / transmitter (USARTx, x=5)

24.2.1. 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(CK_USART5) to produce a dedicated wide range 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. The data frame can be transferred from LSB or MSB bit. The polarity of the TX/RX pins can be configured independently and flexibly.

The USART support DMA function for high-speed data communication.

24.2.2. Characteristics

- NRZ standard format.
- Asynchronous, full duplex communication.
- Half duplex single wire communications.
- Receive / Transmit FIFO function.
- Address match feature in the receiver to reduce address mark wakeup ISR overhead.
- Dual clock domain:
  - Asynchronous PCLK and USART clock.
  - Baud rate programming independent from the PCLK reprogramming.
- Programmable baud-rate generator allowing speed up to 35 MBits/s when the clock frequency is 280 MHz and oversampling is by 8.
- Fully programmable serial interface characteristics:
  - A data word (7 / 8 / 9 or 10 bits) LSB or MSB first.
  - Even, odd or no-parity bit generation / detection.
  - 0.5, 1, 1.5 or 2 stop bit generation.
- Swappable Tx / Rx pin.
- Configurable data polarity.
- Hardware modem operations (CTS / RTS) and RS485 drive enable.
- Configurable multibuffer communication using centralized DMA.
- Separate enable bits for transmitter and receiver.
- Parity control:

- Transmits parity bit.
- Checks parity of received data byte.

- LIN break generation and detection.
- IrDA support.
- Synchronous mode and transmitter clock output for synchronous transmission.
- ISO 7816-3 compliant smartcard interface:
  - Character mode (T = 0).
  - Block mode (T = 1).
  - Direct and inverse convention.
- Multiprocessor communication:
  - Enter into mute mode if address match does not occur.
  - Wake up from mute mode by idle line or address match detection.
- Support for ModBus communication:
  - Timeout feature.
  - CR / LF character recognition.
- Wake up from deep-sleep mode:
  - By standard RBNE interrupt.
  - By WUF interrupt.
- Various status flags:
  - Flags for transfer detection: receive buffer not empty (RBNE), receive FIFO full (RFF), receive FIFO empty (RFE), receive FIFO threshold reached(RFT), receive FIFO not empty(RFNE), transmit buffer empty (TBE), transfer complete (TC), transmit FIFO not full (TFNF), transmit FIFO threshold reached (TFT), transmit FIFO empty (TFE), transmit FIFO threshold reached(TFTIF).
  - 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) and send break (SBF).
  - Flag for multiprocessor communication: IDLE frame detected (IDLEF).
  - Flag for ModBus communication: address/character match (AMF0 / AMF1) and receiver timeout (RTF).
  - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF).
  - Wakeup from deep-sleep mode flag (WUF).
  - Receiver wakeup from mute mode(RWU).
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set.
- Smartcard mode.
- IrDA SIR ENDEC block.
- LIN mode.
- Dual clock domain and wakeup from deep-sleep mode.
- Receiver timeout interrupt.
- ModBus communication.
- Synchronous mode.

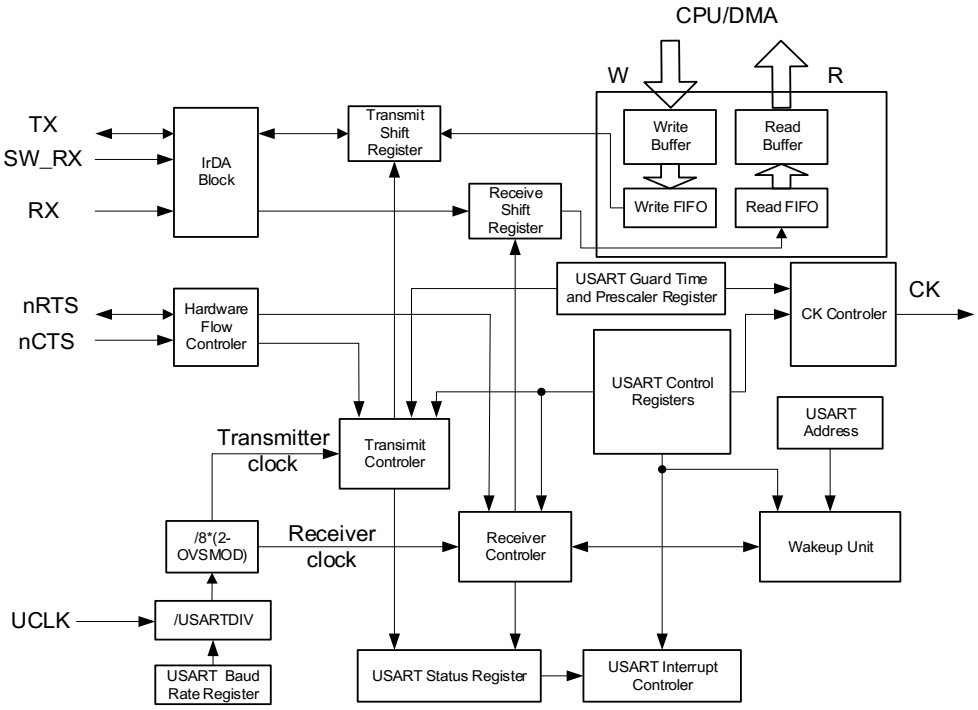
24.2.3. Function overview

The interface is externally connected to another device by the main pins listed in ***Table 24-4. Description of USART important pins***.

Table 24-4. 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 24-18. USART module block diagram

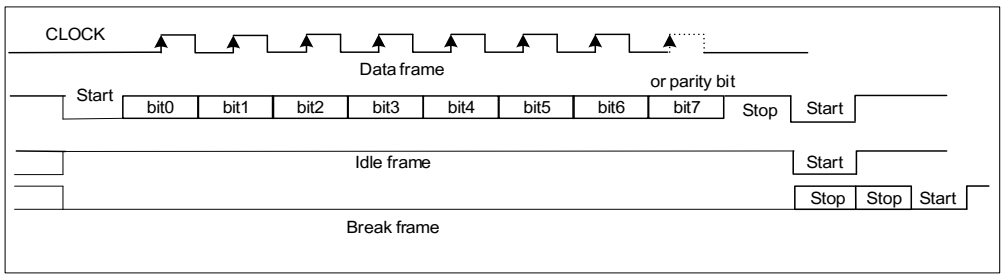


USART frame format

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 WL0 bit and WL1 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 WL1 bit and WL0 bit are reset, the parity bit is the 7th bit. When the WL1 bit is reset and WL0 bit is set, the parity bit is the 8th bit. When WL1 bit is set and the WL0 bit is reset, the parity bit is the 6th bit. When WL1 bit is set and the WL0 bit is set, the parity bit is the 9th bit. The method of calculating the parity bit is selected by the PM bit in

USART_CTL0 register.

Figure 24-19. 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 24-5. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	Default value
01	0.5	Smartcard mode for receiving
10	2	Normal USART and single-wire modes
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK, the configuration of the baud rate generator and the oversampling mode.

Baud rate generation

The baud-rate divider is a 20-bit number which consists of a 16-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 UCLK:

In case of oversampling by 16, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (24\text{-}5)$$

In case of oversampling by 8, the equation is:

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (24\text{-}6)$$

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.

USART transmitter

If the transmit enable bit (TEN) in USART_CTL0 register is set, when the transmit data buffer is not empty, the transmitter shifts out the transmit data frame through the TX pin. The polarity of the TX pin can be configured by the TINV bit in the USART_CTL1 register. Clock pulses can output through the CK pin.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while the transmission is ongoing.

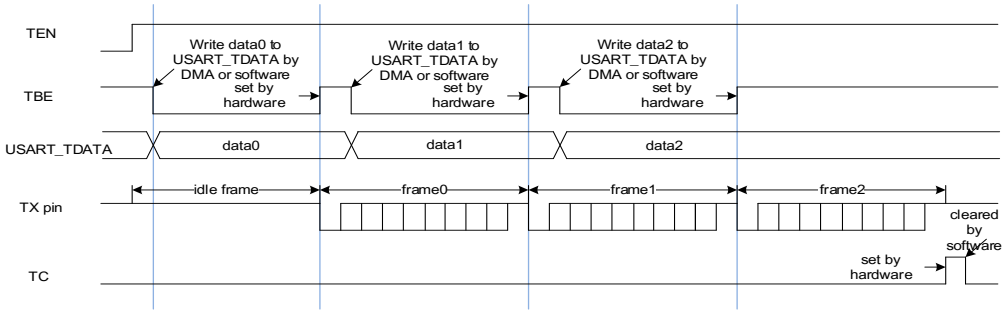
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit in the USART_STAT register is asserted. The TBE bit is cleared by writing USART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the USART_TDATA register while a transmission is ongoing, it will be firstly stored in the transmit buffer, and transferred to the transmit shift register after the current transmission is done. If a data is written to the USART_TDATA register while no transmission is ongoing, the TBE bit will be cleared and set soon, because the data will be transferred to the transmit shift register immediately.

If a frame is transmitted and the TBE bit is asserted, the TC bit of the USART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in ***Figure 24-20. USART transmit procedure***. The software operating process is as follows:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the TEN bit in USART_CTL0.
7. If FIFO is not enabled, wait for TBE to be asserted;
8. Write the data to the USART_TDATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 24-20. 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. The TC bit can be cleared by writing 1 to TCC bit in USART_INTC register.

The break frame is sent when the SBKCMD bit is set, and SBKCMD bit is reset after the transmission.

Note: When FIFO mode is enabled, sending the break character has priority on sending data even if the TXFIFO is full.

USART receiver

After power on, the USART receiver can be enabled by the following procedure:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1.
3. Enable DMA (DENR bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the REN bit in USART_CTL0.

After being enabled, the receiver receives a bit stream after a valid start pulse has been detected. Detection on noisy error, parity error, frame error and overrun error is performed during the reception of a frame.

When a frame is received, the RBNE bit in USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

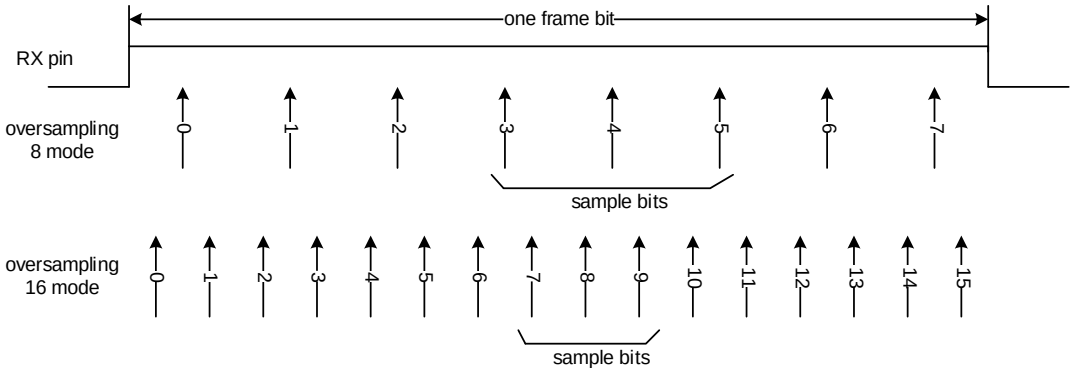
The software can get the received data by reading the USART_RDATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_RDATA register, whatever it is performed by software directly, or through DMA.

The REN bit should not be disabled when reception is ongoing, or the current frame will be lost.

By default, the receiver gets three samples to evaluate the value of a frame bit. If the oversampling 8 mode is enabled, the 3rd, 4th and 5th samples are used, while in the

oversampling 16 mode, the 7th, 8th, and 9th samples are used. If two or more samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the value of the three samples of any bit are not the same, whatever it is a data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame. An interrupt will be generated, If the receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set. If the OSB bit in USART_CTL2 register is set, the receiver gets only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 24-21. Oversampling method of a receive frame bit (OSB=0)



If the parity check function is enabled by setting the PCEN bit in the USART_CTL0 register, the receiver calculates the expected parity value while receiving a frame. The received parity bit will be compared with this expected value. If they are not the same, the parity error (PERR) bit in USART_STAT register will be set. An interrupt is generated, if the PERRIE bit in USART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in USART_STAT register will be set. An interrupt is generated, If the receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set.

When a frame is received, if the RBNE bit is not cleared yet, the last frame will not be stored in the receive data buffer. The overrun error (ORERR) bit in USART_STAT register will be set. An interrupt is generated, if the receive DMA is enabled and the ERRIE bit in USART_CTL2 register is set, or if the RBNEIE is set.

If a noise error (NERR), parity error (PERR), frame error (FERR) or overrun error (ORERR) occurs during reception, NERR, PERR, FERR or ORERR will be set at the same time with RBNE. If the receive DMA is not enabled, when the RBNE interrupt occurs, software need to check whether there is a noise error, parity error, frame error or overrun error.

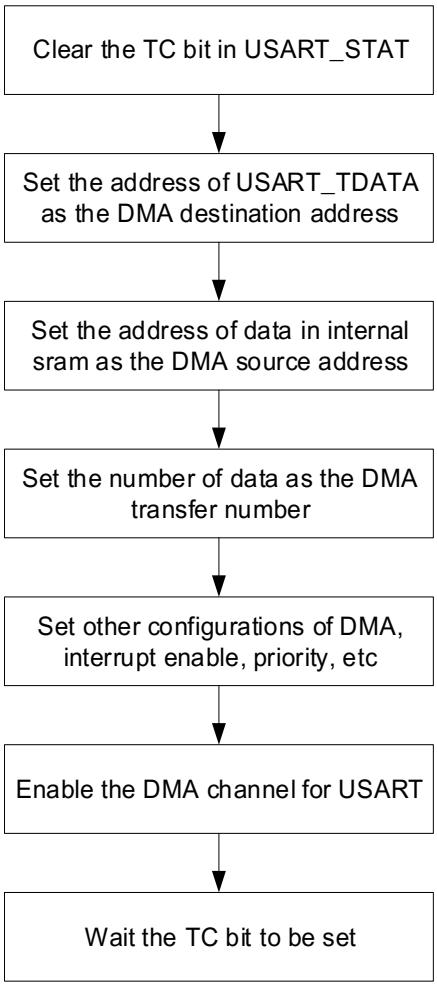
Use DMA for data buffer access

To reduce the burden of the processor, DMA can be used to access the transmitting and receiving data buffer. The DENT bit in USART_CTL2 is used to enable the DMA transmission, and the DENR bit in USART_CTL2 is used to enable the DMA reception.

When DMA is used for USART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the USART. The configuration step are shown in *Figure 24-22.*

Configuration step when using DMA for USART transmission.

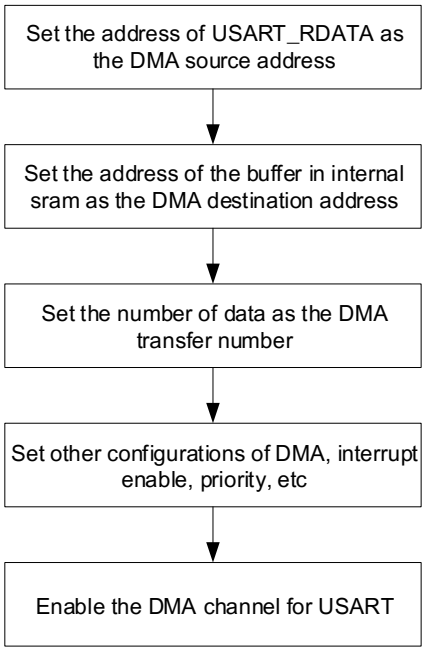
Figure 24-22. Configuration step when using DMA for USART transmission



After all of the data frames are transmitted, the TC bit in USART_STAT is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

When DMA is used for USART reception, DMA transfers data from the receive data buffer of the USART to the internal SRAM. The configuration steps are shown in *Figure 24-23. Configuration step when using DMA for USART reception*. If the ERRIE bit in USART_CTL2 is set, interrupts can be generated by the error status bits (FERR, ORERR and NERR) in USART_STAT.

Figure 24-23. Configuration step when using DMA for USART reception

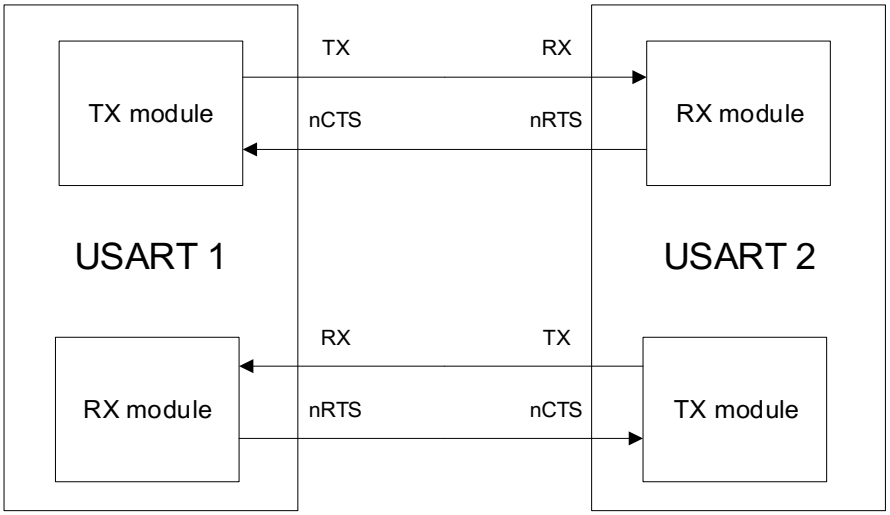


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

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 24-24. Hardware flow control between two USARTs



RTS flow control

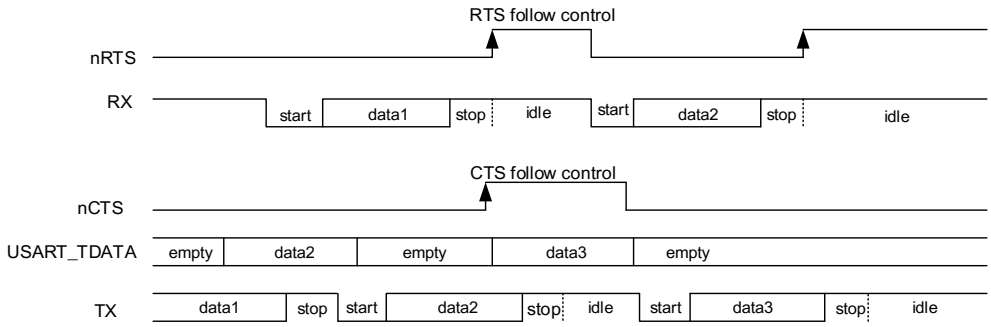
The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When

data frame is received, the nRTS signal goes high to prevent the transmitter from sending next frame. The nRTS signal keeps high when the receive buffer is full.

CTS flow control

The USART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in USART_STAT is '0' and the nCTS signal is low, the transmitter transmits the data frame. When the nCTS signal goes high during a transmission, the transmitter stops after the current transmission is accomplished.

Figure 24-25. Hardware flow control



RS485 Driver Enable

The driver enable feature, which is enabled by setting bit DEM in the USART_CTL2 control register, allows the user to activate the external transceiver control, through the DE (Driver Enable) signal. The assertion time, which is programmed using the DEA [4:0] bits field in the USART_CTL0 control register, is the time between the activation of the DE signal and the beginning of the START bit. The de-assertion time, which is programmed using the DED [4:0] bits field in the USART_CTL0 control register, is the time between the end of the last stop bit and the de-activation of the DE signal. The polarity of the DE signal can be configured using the DEP bit in the USART_CTL2 control register.

Multi-processor communication

In multiprocessor communication, several USARTs are connected as a network. It will be a big burden for a device to monitor all of the messages on the RX pin. To reduce the burden of a device, software can put an USART module into a mute mode by writing 1 to the MMCMD bit in USART_CMD register.

If a USART is in mute mode, all of the receive status bits cannot be set. The USART can also be wake up by hardware by one of the two methods: idle frame method and address match method.

The idle frame wake up method is selected by default. If the RWU bit is reset, an idle frame is detected on the RX pin, the IDLEF bit in USART_STAT will be set. If the RWU bit is set, an idle frame is detected on the RX pin, the hardware clears the RWU bit and exits the mute mode. When it is woken up by an idle frame, the IDLEF bit in USART_STAT will not be set.

When the WM bit of in USART_CTL0 register is set, the MSB bit of a frame is detected as the address flag. If the address flag is high, the frame is treated as an address frame. If the address flag is low, the frame is treated as a data frame. If the LSB 4 or 7 bits, which are configured by the ADDM0 bit of the USART_CTL1 register or ADDM1 bit of the USART_CTL2, of an address frame is the same as the ADDR0 bits in the USART_CTL1 register or ADDR1 bits in the USART_CTL2 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_STAT register. If the LSB 4/7 bits of an address frame defers from the ADDR0 bits in the USART_CTL1 register or ADDR1 bits in the USART_CTL2 register, the hardware sets the RWU bit and enters mute mode automatically. In this situation, the RBNE bit is not set.

If the PCEN bit in USART_CTL0 is set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address bit. If the ADDM0 or ADDM1 bit is set and the receive frame is a 8bit data, the LSB 7 bits will be compared with ADDR0[6:0] or ADDR1[6:0]. If the ADDM0 or ADDM1 bit is set and the receive frame is a 9bit data, the LSB 8 bits will be compared with ADDR0[7:0] or ADDR1[7:0].

Match address operation function in the same way for both ADDR0 and ADDR1. Note that the position of the address mark is the same as the Parity Bit When parity is enabled for 8 bit and 9 bit data formats.

If only one of AMEN0 and AMEN1 bit is set, a match address is compared only with the associate ADDR0 or ADDR1 and data is transferred to the receive data buffer only on a match.

If AMEN0 and AMEN1 are all set, a match address is compared with both ADDR0 and ADDR1 and data is transferred only on a match with either ADDR0 or ADDR1. So, a second match address can be used for broadcast or general call address to serial bus.

Note: If the MEN bit is set, the WM bit is reset and the RWU bit is reset, an idle frame is detected on the RX pin, the IDLEF bit will be set. If the RWU bit is set, the IDLEF is not set. Otherwise, AMEN0 or AMEN1 must be set when using address match method to wakeup USART from mute mode.

LIN mode

The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

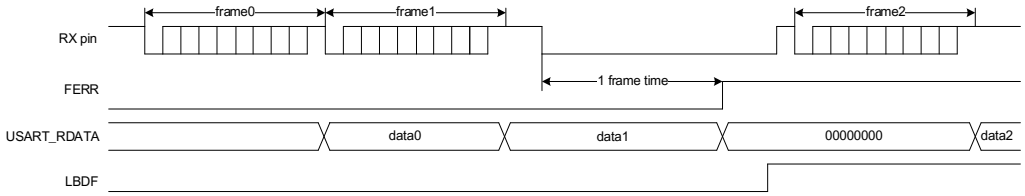
When transmitting a normal data frame, the transmission procedure is the same as the normal USART mode. The data bits length can only be 8. And the break frame is 13-bit '0', followed by 1 stop bit.

The break detection function is totally independent of the normal USART receiver. So a break frame can be detected during the idle state or during a frame. The expected length of a break frame can be selected by configuring LBLEN in USART_CTL1. When the RX pin is detected

at low state for a time that is equal to or longer than the expected break frame length (10 bits when LBLEN=0, or 11 bits when LBLEN=1), the LBDF bit in USART_STAT is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

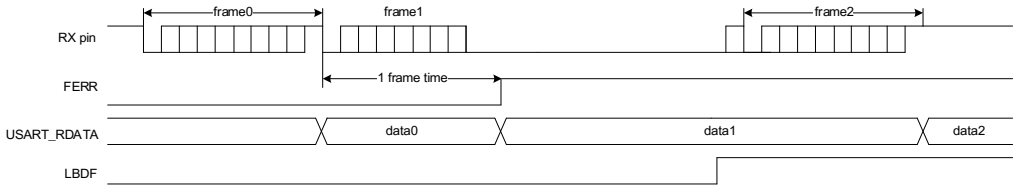
As shown in ***Figure 24-26. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all '0' frame, with an asserted FERR status.

Figure 24-26. Break frame occurs during idle state



As shown in ***Figure 24-27. 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 24-27. Break frame occurs during a frame



Synchronous mode

The USART can be used for full-duplex synchronous serial communications only in master mode, by setting the CKEN bit in USART_CTL1. The LMEN bit in USART_CTL1 and SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in synchronous mode. The CK pin is the clock output of the synchronous USART transmitter, and can be only activated when the TEN bit is enabled. No clock pulse will be sent through the CK pin during the transmission of the start bit and stop bit. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the last (address flag) bit transmission. The clock output is also not activated during idle and break frame sending. The CPH bit in USART_CTL1 can be used to determine whether data is captured on the first or the second clock edge. The CPL bit in USART_CTL1 can be used to configure the clock polarity in the USART synchronous idle state.

The CPL, CPH and CLEN bits in USART_CTL1 determine the waveform on the CK pin. Software can only change them when the USART is disabled (UEN=0).

The clock is synchronized with the data transmitted. The receiver in synchronous mode samples the data on the transmitter clock without any oversampling.

Figure 24-28. Example of USART in synchronous mode

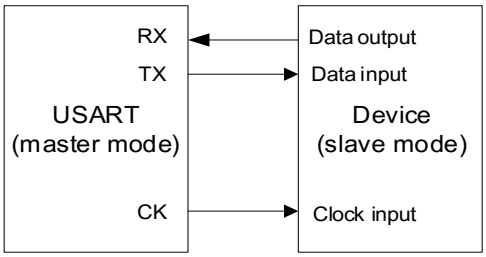
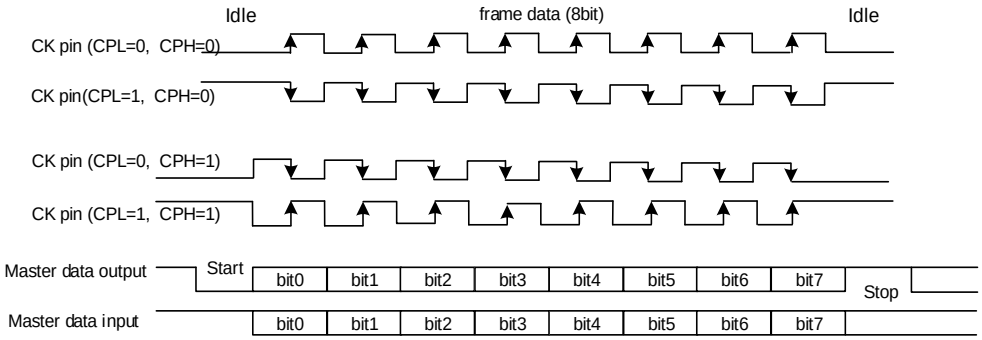


Figure 24-29. 8-bit format USART synchronous waveform (CLEN=1)

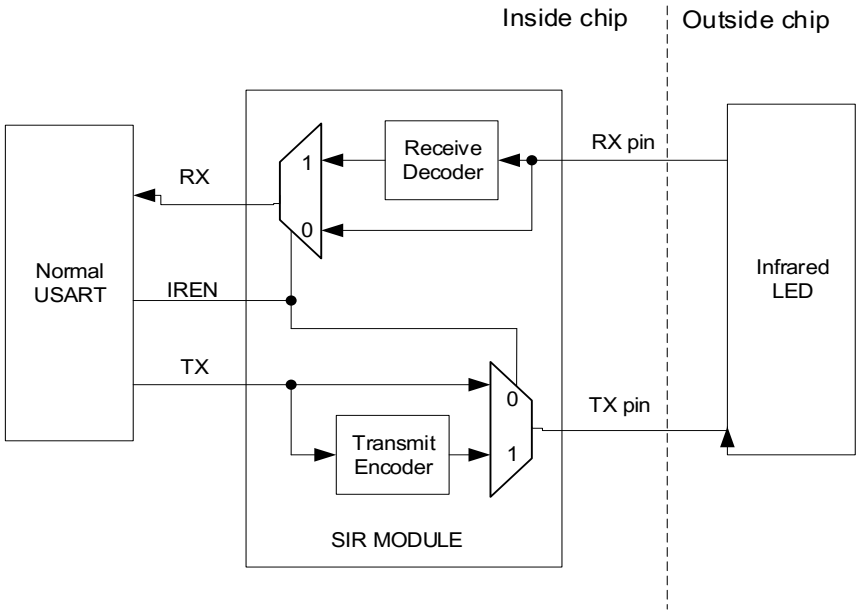


IrDA SIR ENDEC mode

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.

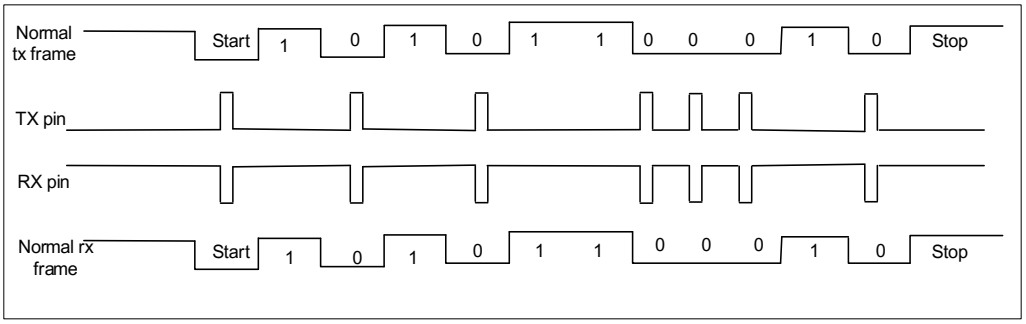
Figure 24-30. IrDA SIR ENDEC module



In IrDA mode, the polarity of the TX and RX pins is different. The TX pin is usually at low state, while the RX pin is usually at high state. The IrDA pins keep stable to represent the logic '1', while an infrared light pulse on the IrDA pins (a Return to Zero signal) represents the logic '0'. The pulse width should be 3/16 of a bit period. The IrDA could not detect any pulse if the pulse width is less than 1 PSC clock. While it can detect a pulse by chance if the pulse width is greater than 1 but smaller than 2 times of PSC clock.

Because the IrDA is a half-duplex protocol, the transmission and the reception should not be carried out at the same time in the IrDA SIR ENDEC block.

Figure 24-31. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit encoder is driven by a low speed clock, which is divided from the PCLK. The division ratio is configured by the PSC[7:0] bits in USART_GP register. The pulse width on the TX pin is 3 cycles of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

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 The TX and RX pins are connected together internally, and the RX pin is no longer used. The TX pin should be configured as open drain mode and communication conflicts are handled by software. When the TEN bit is set, the data in the data register will be sent.

Smartcard (ISO7816-3) mode

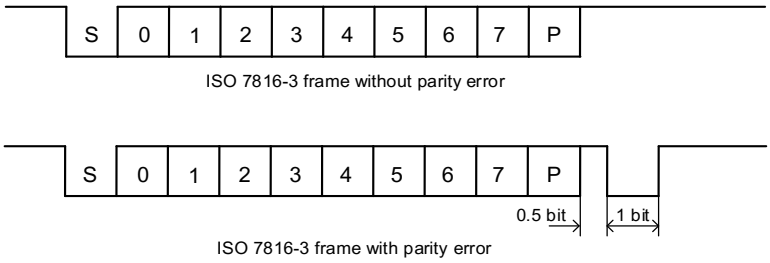
The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T=0) mode and the block (T=1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be reset in smartcard mode.

A clock is provided to the smartcard if the CKEN bit is set. The clock can be divided for other use.

The frame consists of 1 start bit, 9 data bits (1 parity bit included) and 1.5 stop bits.

The smartcard mode is a half-duplex communication protocol. When connected to a smartcard, the TX pin must be configured as open drain mode, and drives a bidirectional line that is also driven by the smartcard.

Figure 24-32. ISO7816-3 frame format



Character (T=0) mode

Compared to the timing in normal operation, the transmission time from transmit shift register to the TX pin is delayed by half baud clock, and the TC flag assertion time is delayed by a guard time that is configured by the GUAT[7:0] bits in USART_GP. In Smartcard mode, the internal guard time counter starts counting up after the stop bits of the last data frame, and the GUAT[7:0] bits should be configured as the character guard time (CGT) in ISO7816-3 protocol minus 12. The TC status is forced reset while the guard time counter is counting up. When the counter reaches the programmed value TC is asserted high.

During USART transmission, if a parity error event is detected, the smartcard may NACK the current frame by pulling down the TX pin during the last 1 bit time of the stop bits. The USART

can automatically resend data according to the protocol for SCRTNUM times. An interframe gap of 2 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the frame error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE/receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the Smartcard mode.

Block (T=1) mode

In block (T=1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. A timeout interrupt will be generated, if no answer is received from the card before the expiration of this period. If the first character is received before the expiration of the period, it is signaled by the RBNE interrupt. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

In order to allow the automatic check of the maximum wait time between two consecutive characters, the USART_RT register must be programmed to the CWT (character wait time) - 11 value, which is expressed in baudtime units, after the reception of the first character (RBNE interrupt). The USART signals to the software through the RT flag and interrupt (when RTIE bit is set), if the smartcard doesn't send a new character in less than the CWT period after the end of the previous character.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE=0), to count the number of received characters. The length of the block, which must be programmed in the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). This register field must be programmed to the minimum value (0x0), before the start of the block, when using DMA mode. With this value, an interrupt is generated after the 4th received character. The software must read the third byte as block length from the receive buffer.

In interrupt driven receive mode, the length of the block may be checked by software or by programming the BL value. However, before the start of the block, the maximum value of BL (0xFF) may be programmed. The real value will be programmed after the reception of the

third character.

The total block length (including prologue, epilogue and information fields) equals BL+4. The end of the block is signaled to the software through the EBF flag and interrupt (when EBIE bit is set). The RT interrupt may occur in case of an error in the block length.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

The direct convention is defined as: LSB first, logical bit value of 1 corresponds to high state of the line and parity is even. In this case, the following control bits must be programmed: MSBF=0, DINV=0 (default values).

The inverse convention is defined as: MSB first, logical bit value 1 corresponds to an low state on the signal line and parity is even. In this case, the following control bits must be programmed: MSBF=1, DINV=1.

ModBus communication

The USART offers basic support for the implementation of ModBus/RTU and ModBus/ASCII protocols by implementing an end of block detection.

In the ModBus/RTU mode, the end of one block is recognized by an idle line for more than 2 characters time. This function is implemented through the programmable timeout function.

To detect the idle line, the RTEN bit in the USART_CTL1 register and the RTIE in the USART_CTL0 register must be set. The USART_RT register must be set to the value corresponding to a timeout of 2 characters time. After the last stop bit is received, when the receive line is idle for this duration, an interrupt will be generated, informing the software that the current block reception is completed.

In the ModBus/ASCII mode, the end of a block is recognized by a specific (CR/LF) character sequence. The USART manages this mechanism using the character match function by programming the LF ASCII code in the ADDR field and activating the address match interrupt (AMIE=1). When a LF has been received or can check the CR/LF in the DMA buffer, the software will be informed.

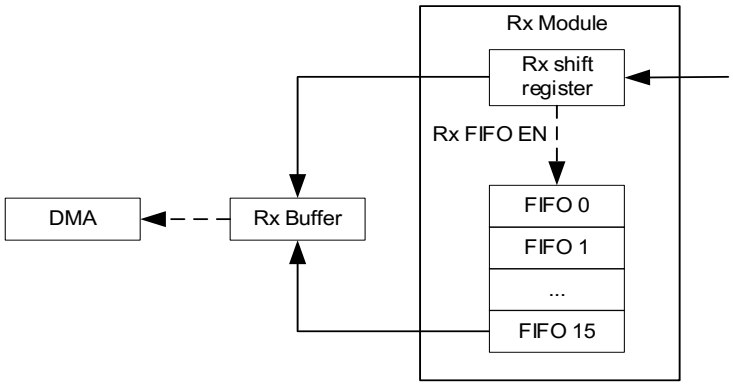
Receive / Transmitter FIFO

Receive FIFO

The receive FIFO can be enabled by setting the FEN bit of the USART_RFCS register to avoid the overrun error when the CPU can't serve the RBNE interrupt immediately. Up to 17 frames receive data can be stored in the receive FIFO and receive buffer. The RFF flag will be set when the receive FIFO is full, an interrupt is generated if the RFFIE bit is set. The RFT flag will be set when the receive FIFO reaches the threshold configured in the RFTCFG[2:0] bits., an interrupt is generated if the RFTIE bit is set. An interrupt will be generated when

receive FIFO is not empty if RFNEIE bit is set.

Figure 24-33. USART receive FIFO structure

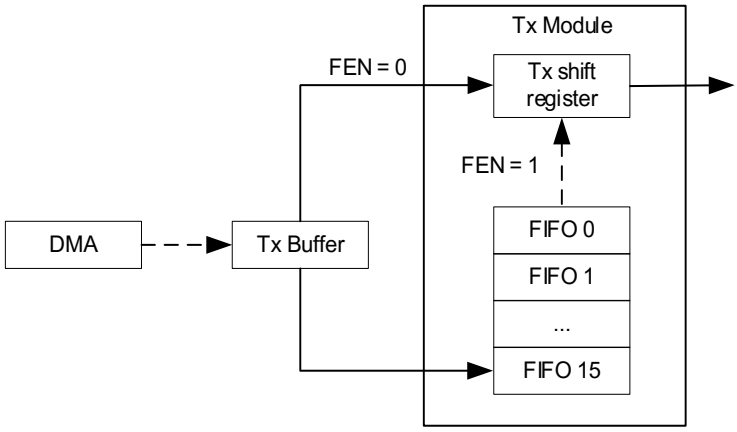


If the software read receive data buffer in the routing of the RBNE interrupt, the RBNEIE bit should be reset at the beginning of the routing and set after all of the receive data is read out. The PERR / NERR / FERR / EBF flags should be cleared before reading a receive data out.

Transmit FIFO

The Transmitter FIFO can be enabled by setting the FEN bit of the USART_RFCS register Up to 17 frames transmitter data can be stored in the transmitter FIFO and transmitter buffer. The TFE flag will be set when the transmitter FIFO is empty, an interrupt is generated if the TFEIE bit is set. The TFTIF flag will be set (or TFT flag will be clear) when the available space in transmitter FIFO reaches the threshold configured in the TFTCFG[2:0] bits, an interrupt is generated if the TFTIE bit is set. TFNF flag will be set when transmitter FIFO is not full, an interrupt will be generated if TFNFIE bit is set.

Figure 24-34. USART transmitter FIFO structure



Wakeup from deep-sleep mode

The USART is able to wake up the MCU from deep-sleep mode and deep-sleep 1 mode by the standard RBNE interrupt or the WUM interrupt.

The UESM bit must be set and the USART clock must be set to IRC8M or LXTAL (refer to the reset and clock unit RCU section).

When using the standard RBNE interrupt, the RBNEIE bit must be set before entering deep-sleep mode and deep-sleep 1 mode.

When using the WUIE interrupt, the source of WUIE interrupt may be selected through the WUM bit fields.

DMA must be disabled before entering deep-sleep mode and deep-sleep 1 mode. Before entering deep-sleep mode, software must check that the USART is not performing a transfer, by checking the BSY flag in the USART_STAT register. The REA bit must be checked to ensure the USART is actually enabled.

When the wakeup event is detected, the WUF flag is set by hardware and a wakeup interrupt is generated if the WUIE bit is set, independently of whether the MCU is in stop or active mode.

USART interrupts

The USART interrupt events and flags are listed in *Table 24-6. USART interrupt requests*.

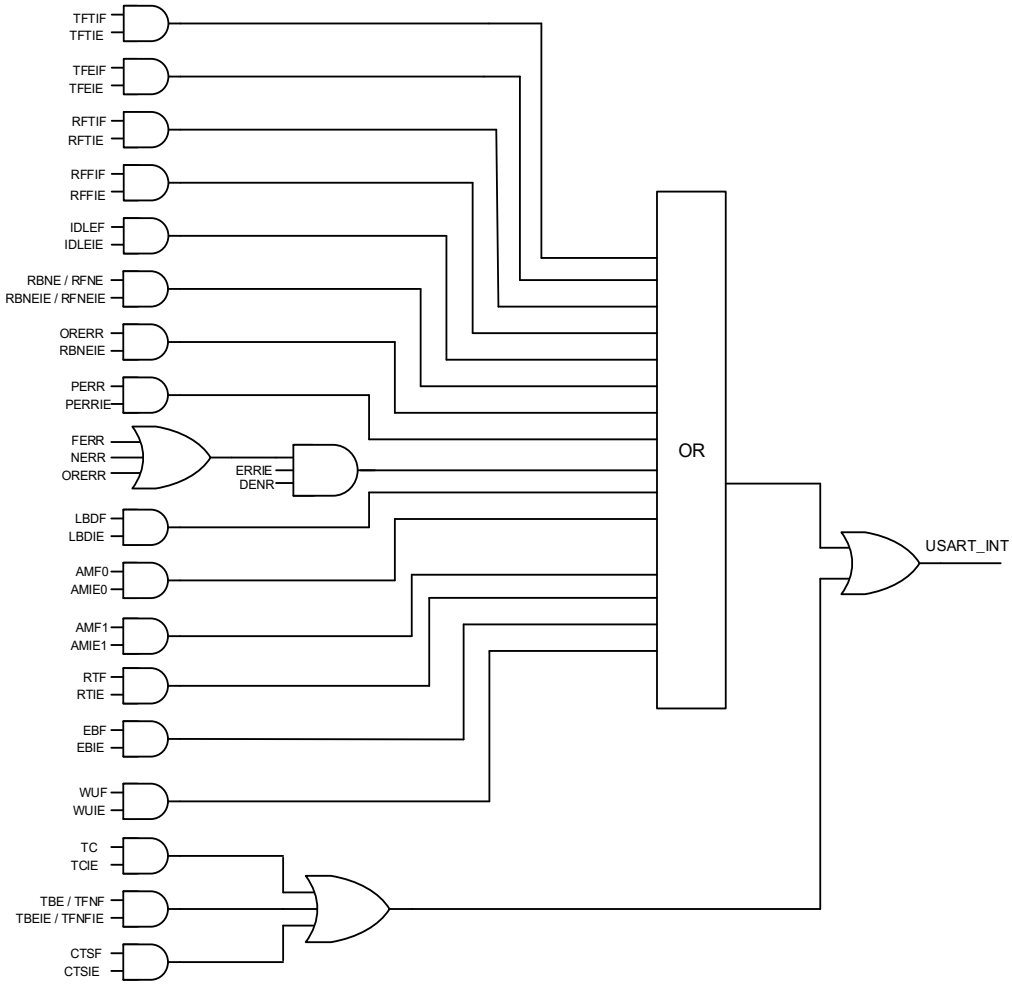
Table 24-6. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty or Transmit FIFO not full	TBE / TFNF	TBEIE / TFNFIE
CTS flag	CTSF	CTSIE
Transmission complete	TC	TCIE
Received data ready to be read or Receive FIFO not empty	RBNE / RFNE	RBNEIE / RFNEIE
Overrun error detected	ORERR	
Receive FIFO full	RFFIF	RFFIE
Receive FIFO threshold reaches	RFTIF	RFTIE
Transmit FIFO empty	TFEIF	TFEIE
Transmit FIFO threshold reaches	TFTIF	TFTIE
Idle line detected	IDLEF	IDLEIE
Parity error flag	PERR	PERRIE
Break detected flag in LIN mode	LBDF	LBDIE
Reception errors (noise flag, overrun error, framing error)	NERR or ORERR or FERR	ERRIE
ADDR0 match	AMF0	AMIE0
ADDR1 match	AMF1	AMIE1

Interrupt event	Event flag	Enable Control bit
Receiver timeout error	RTF	RTIE
End of Block	EBF	EBIE
Wakeup from Deep-sleep mode	WUF	WUIE

All of the interrupt events are ORed together before being sent to the interrupt controller, so the USART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine.

Figure 24-35. USART interrupt mapping diagram



24.2.4. Register defintion

USART5 base address: 0x4001 7000

Control register 0 (USART_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AMIE1	Reserved		WL1	EBIE	RTIE	DEA[4:0]					DED[4:0]				
rw			rw	rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE0	MEN	WL0	WM	PCEN	PM	PERRIE	TBEIE / TFNFIE	TCIE	RBNEIE / RFNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	AMIE1	ADDR1 character match interrupt enable 0: ADDR1 character match interrupt is disabled 1: ADDR1 character match interrupt is enabled
30:29	Reserved	Must be kept at reset value.
28	WL1	Word length 1 This bit and WL0 bit determine the word length 00: 8 Data bits 01: 9 Data bits 10: 7 Data bits 11: 10 Data bits This bit field cannot be written when the USART is enabled (UEN = 1). **Note:** 1. In 7 Data bits data length mode, the Smartcard mode and LIN master mode are not supported. 2. In 10 Data bits data length mode address mark is not supported
27	EBIE	End of block interrupt enable 0: End of block interrupt is disabled 1: End of block interrupt is enabled
26	RTIE	Receiver timeout interrupt enable 0: Receiver timeout interrupt is disabled 1: Receiver timeout interrupt is enabled
25:21	DEA[4:0]	Driver enable assertion time These bits are used to define the time between the activation of the DE (driver

		enable) signal and the beginning of the start bit. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN = 1).
20:16	DED[4:0]	Driver enable de-assertion time These bits are used to define the time between the end of the last stop bit, in a transmitted message, and the de-activation of the DE (driver enable) signal. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN = 1).
15	OVSMOD	Oversample mode 0: Oversampling by 16 1: Oversampling by 8 This bit must be kept cleared in LIN, IrDA and smartcard modes. This bit field cannot be written when the USART is enabled (UEN = 1).
14	AMIE0	ADDR0 character match interrupt enable 0: ADDR0 character match interrupt is disabled 1: ADDR0 character match interrupt is enabled
13	MEN	Mute mode enable 0: Mute mode disabled 1: Mute mode enabled
12	WL0	Word length 0 This bit and WL1 bit determine the word length This bit field cannot be written when the USART is enabled (UEN = 1).
11	WM	Wakeup method in mute mode 0: Idle Line 1: Address match This bit field cannot be written when the USART is enabled (UEN = 1).
10	PCEN	Parity control enable 0: Parity control disabled 1: Parity control enabled This bit field cannot be written when the USART is enabled (UEN = 1).
9	PM	Parity mode 0: Even parity 1: Odd parity This bit field cannot be written when the USART is enabled (UEN = 1).
8	PERRIE	Parity error interrupt enable 0: Parity error interrupt is disabled 1: An interrupt will occur whenever the PERR bit is set in USART_STAT

7	TBEIE	When FIFO is disabled: Transmitter register empty interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the TBE bit is set in USART_STAT
	TFNFIE	When FIFO is enabled: Transmit FIFO not full interrupt enable 0: Transmit FIFO not full interrupt is disabled 1: An interrupt will occur whenever the TFNF bit is set in USART_STAT
6	TCIE	Transmission complete interrupt enable If this bit is set, an interrupt occurs when the TC bit in USART_STAT is set. 0: Transmission complete interrupt is disabled 1: Transmission complete interrupt is enabled
5	RBNEIE	When FIFO is disabled: Read data buffer not empty interrupt and overrun error interrupt enable 0: Read data register not empty interrupt and overrun error interrupt disabled 1: An interrupt will occur whenever the ORERR bit is set or the RBNE bit is set in USART_STAT.
	RFNEIE	When FIFO is enabled: Receive FIFO not empty interrupt and overrun error interrupt enable 0: Receive FIFO not empty interrupt and overrun error interrupt is disabled 1: An interrupt will occur whenever the ORERR bit is set or the RFNE bit is set in USART_STAT.
4	IDLEIE	IDLE line detected interrupt enable 0: IDLE line detected interrupt disabled 1: An interrupt will occur whenever the IDLEF bit is set in USART_STAT.
3	TEN	Transmitter enable 0: Transmitter is disabled 1: Transmitter is enabled
2	REN	Receiver enable 0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit
1	UESM	USART enable in Deep-sleep mode 0: USART not able to wake up the MCU from Deep-sleep mode. 1: USART able to wake up the MCU from Deep-sleep mode. Providing that the clock source for the USART must be CK_IRC8M or CK_LXTAL.
0	UEN	USART enable 0: USART prescaler and outputs disabled 1: USART prescaler and outputs enabled

Control register 1 (USART_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[7:0]								RTEN	Reserved			MSBF	DINV	TINV	RINV
rw								rw				rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	Reserved	LBDIE	LBLEN	ADDM0	Reserved		LBEN	AMEN0
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw			rw	rw

Bits	Fields	Descriptions
31:24	ADDR0[7:0]	Address 0 of the USART terminal These bits give the address 0 of the USART terminal. In multiprocessor communication during mute mode or deep-sleep mode, this is used for wakeup with address match detection. The received frame, the MSB of which is equal to 1, will be compared to these bits. When the ADDM0 bit is reset, only the ADDR0[3:0] bits are used to compare. In normal reception, these bits are also used for character detection. The whole received character (8-bit) is compared to the ADDR0[7:0] value and AMF0 flag is set on matching. This bit field cannot be written when both reception (REN = 1) and USART (UEN = 1) are enabled.
23	RTEN	Receiver timeout enable 0: Receiver timeout function disabled 1: Receiver timeout function enabled
22:20	Reserved	Must be kept at reset value.
19	MSBF	Most significant bit first 0: Data is transmitted / received with the LSB first 1: Data is transmitted / received with the MSB first This bit field cannot be written when the USART is enabled (UEN = 1).
18	DINV	Data bit level inversion 0: Data bit signal values are not inverted 1: Data bit signal values are inverted This bit field cannot be written when the USART is enabled (UEN = 1).
17	TINV	TX pin level inversion 0: TX pin signal values are not inverted 1: TX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN = 1).

16	RINV	RX pin level inversion 0: RX pin signal values are not inverted 1: RX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN = 1).
15	STRP	Swap TX / RX pins 0: The TX and RX pins functions are not swapped 1: The TX and RX pins functions are swapped This bit field cannot be written when the USART is enabled (UEN = 1).
14	LMEN	LIN mode enable 0: LIN mode disabled 1: LIN mode enabled This bit field cannot be written when the USART is enabled (UEN = 1).
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit This bit field cannot be written when the USART is enabled (UEN = 1).
11	CKEN	CK pin enable 0: CK pin disabled 1: CK pin enabled This bit field cannot be written when the USART is enabled (UEN = 1).
10	CPL	Clock polarity 0: Steady low value on CK pin outside transmission window in synchronous mode 1: Steady high value on CK pin outside transmission window in synchronous mode This bit field cannot be written when the USART is enabled (UEN = 1).
9	CPH	Clock phase 0: The first clock transition is the first data capture edge in synchronous mode 1: The second clock transition is the first data capture edge in synchronous mode This bit field cannot be written when the USART is enabled (UEN = 1).
8	CLEN	CK length 0: The clock pulse of the last data bit (MSB) is not output to the CK pin in synchronous mode 1: The clock pulse of the last data bit (MSB) is output to the CK pin in synchronous mode This bit field cannot be written when the USART is enabled (UEN = 1)
7	Reserved	Must be kept at reset value.
6	LBDIE	LIN break detection interrupt enable 0: LIN break detection interrupt is disabled

		1: An interrupt will occur whenever the LBDF bit is set in USART_STAT
5	LBLEN	LIN break frame length 0: 10 bit break detection 1: 11 bit break detection This bit field cannot be written when the USART is enabled (UEN = 1).
4	ADDM0	Address 0 detection mode This bit is used to select between 4-bit address detection and full-bit address 0 detection. 0: 4-bit address detection 1: Full-bit address detection. In 7-bit, 8-bit and 9-bit data modes, the address detection is done on 6-bit, 7-bit and 8-bit address (ADDR0[5:0], ADDR0[6:0] and ADDR0[7:0]) respectively This bit field cannot be written when the USART is enabled (UEN = 1).
3:2	Reserved	Must be kept at reset value.
1	LBEN	Loopback mode 0: Loopback mode disable 1: Loopback mode enable This bit field cannot be written when the USART is enabled (UEN = 1).
0	AMEN0	Address 0 match mode enable 0: Address 0 match mode disable 1: Address 0 match mode enable. This bit field cannot be written when the USART is enabled (UEN = 1).

Control register 2 (USART_CTL2)

Address offset: 0x08

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[7:0]								ADDM1	WUIE	WUM[1:0]		SCRTNUM[2:0]			AMEN1
rw								rw	rw	rw		rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	ADDR1[7:0]	Address 1 of the USART terminal These bits give the address 1 of the USART terminal. In multiprocessor communication during mute mode or Deep-sleep mode, this is used for wakeup with address mark detection. The received frame, the MSB of

		which is equal to 1, will be compared to these bits. When the ADDM1 bit is reset, only the ADDR1 [3:0] bits are used to compare. In normal reception, these bits are also used for character detection. The whole received character (8-bit) is compared to the ADDR1[7:0] value and AMF1 flag is set on matching. This bit field cannot be written when both reception (REN = 1) and USART are enabled (UEN = 1).
23	ADDM1	Address 1 detection mode This bit is used to select between 4-bit address detection and full-bit address 1 detection. 0: 4-bit address detection 1: Full-bit address detection. In 7-bit, 8-bit and 9-bit data modes, the address detection is done on 6-bit, 7-bit and 8-bit address (ADDR1[5:0], ADDR1[6:0] and ADDR1[7:0]) respectively. This bit field cannot be written when the USART is enabled (UEN = 1).
22	WUIE	Wakeup from deep-sleep mode interrupt enable 0: Wakeup from deep-sleep mode interrupt is disabled 1: Wakeup from deep-sleep mode interrupt is enabled
21:20	WUM[1:0]	Wakeup mode from deep-sleep mode These bits are used to specify the event which activates the WUF (wakeup from deep-sleep mode flag) in the USART_STAT register. 00: WUF active on address match, which is defined by ADDR0 and ADDM0 or ADDR1 and ADDM1. 01: Reserved 10: WUF active on start bit 11: WUF active on RBNE This bit field cannot be written when the USART is enabled (UEN = 1).
19:17	SCRTNUM[2:0]	Smartcard auto-retry number In smartcard mode, these bits specify the number of retries in transmission and reception. In transmission mode, a transmission error (FERR bit set) will occur after this number of automatic retransmission retries. In reception mode, reception error (RBNE and PERR bits set) will occur after this number or erroneous reception trials. When these bits are configured as 0x0, there will be no automatic retransmission in transmit mode. This bit field is only can be cleared to 0 when the USART is enabled (UEN = 1), to stop retransmission.
16	AMEN1	Address 1 match mode enable 0: Address 1 match mode disable 1: Address 1 match mode enable

This bit field cannot be written when the USART is enabled (UEN = 1).

15 DEP
Driver enable polarity mode
0: DE signal is active high
1: DE signal is active low
This bit field cannot be written when the USART is enabled (UEN = 1)

14 DEM
Driver enable mode
This bit is used to activate the external transceiver control, through the DE signal, which is output on the RTS pin.
0: DE function is disabled
1: DE function is enabled
This bit field cannot be written when the USART is enabled (UEN = 1).

13 DDRE
Mask DMA request on reception error.
0: DMA is not disabled in case of reception error. The DMA request is not asserted to make sure the erroneous data is not transferred, but the next correct received data will be transferred. The RBNE is kept 0 to prevent overrun when reception error, but the corresponding error flag is set. This mode can be used in smartcard mode
1: The DMA request is not asserted in case of reception error until the error flag is cleared. The RBNE flag and corresponding error flag will be set. The software must first disable the DMA request (DENR = 0) or clear RBNE / RFNE in USART_STAT when FIFO mode is enabled before clearing the error flag
This bit field cannot be written when the USART is enabled (UEN = 1).

12 OVRD
Overrun disable
0: Overrun functionality is enabled. The ORERR error flag will be set when received data is not read before receiving new data, and the new data will be lost
1: Overrun functionality is disabled. The ORERR error flag will not be set when received data is not read before receiving new data, and the new received data overwrites the previous content of the USART_RDATA register. When FIFO mode is enabled, the data is written in USART_RDATA directly and Receive FIFO is bypassed. Even if FIFO is enabled, the RBNE bit is to be used.
This bit field cannot be written when the USART is enabled (UEN = 1).

11 OSB
One sample bit method
0: Three sample bit method
1: One sample bit method
This bit field cannot be written when the USART is enabled (UEN = 1).

10 CTSIE
CTS interrupt enable
0: CTS interrupt is disabled
1: An interrupt will occur whenever the CTS bit is set in USART_STAT

9 CTSEN
CTS enable
0: CTS hardware flow control disabled

		1: CTS hardware flow control enabled This bit field cannot be written when the USART is enabled (UEN = 1).
8	RTSEN	RTS enable 0: RTS hardware flow control disabled 1: RTS hardware flow control enabled, data can be requested only when there is space in the receive buffer This bit field cannot be written when the USART is enabled (UEN = 1).
7	DENT	DMA enable for transmission 0: DMA mode is disabled for transmission 1: DMA mode is enabled for transmission
6	DENR	DMA enable for reception 0: DMA mode is disabled for reception 1: DMA mode is enabled for reception
5	SCEN	Smartcard mode enable 0: Smartcard mode disabled 1: Smartcard mode enabled This bit field cannot be written when the USART is enabled (UEN = 1).
4	NKEN	NACK enable in Smartcard mode 0: Disable NACK transmission when parity error 1: Enable NACK transmission when parity error This bit field cannot be written when the USART is enabled (UEN = 1).
3	HDEN	Half-duplex enable 0: Half duplex mode is disabled 1: Half duplex mode is enabled This bit field cannot be written when the USART is enabled (UEN = 1).
2	IRLP	IrDA low-power 0: Normal mode 1: Low-power mode This bit field cannot be written when the USART is enabled (UEN = 1).
1	IREN	IrDA mode enable 0: IrDA disabled 1: IrDA enabled This bit field cannot be written when the USART is enabled (UEN = 1).
0	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: An interrupt will occur whenever the FERR bit or the ORERR bit or the NERR bit is set in USART_STAT in multibuffer communication

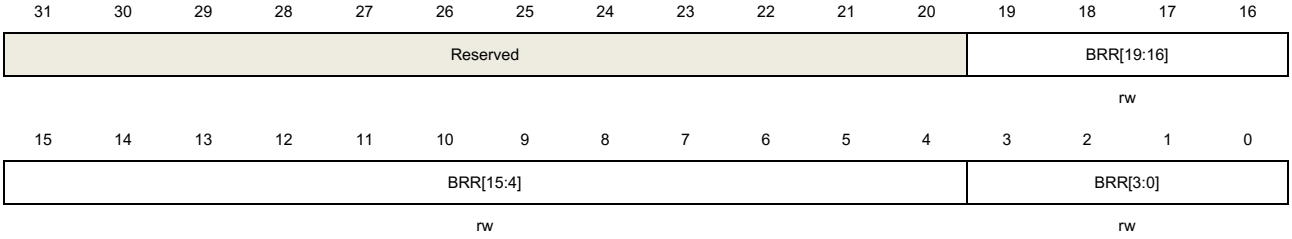
Baud rate generator register (USART_BAUD)

Address offset: 0x0C

Reset value: 0x0000 0000

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

This register cannot be written when the USART is enabled (UEN=1).



Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:4	BRR[19:4]	Integer of baud-rate divider INTDIV = BRR[19:4]
3:0	BRR[3:0]	Fraction of baud-rate divider If OVSMOD = 0, FRADIV = BRR [3:0]; If OVSMOD = 1, FRADIV = BRR [2:0], BRR [3] must be reset.

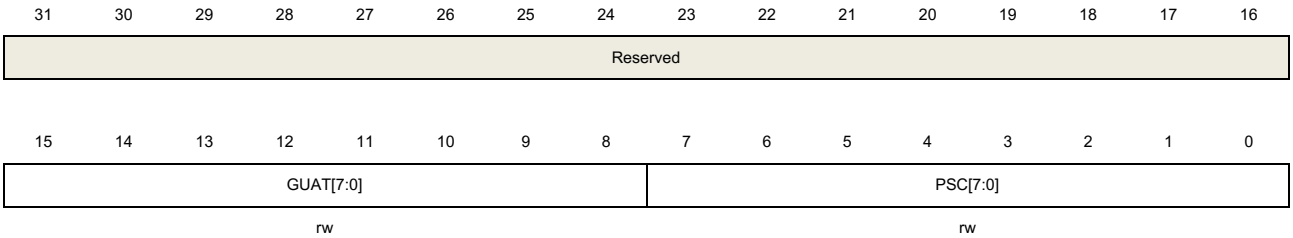
Prescaler and guard time configuration register (USART_GP)

Address offset: 0x10

Reset value: 0x0000 0000

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

This register cannot be written when the USART is enabled (UEN=1).



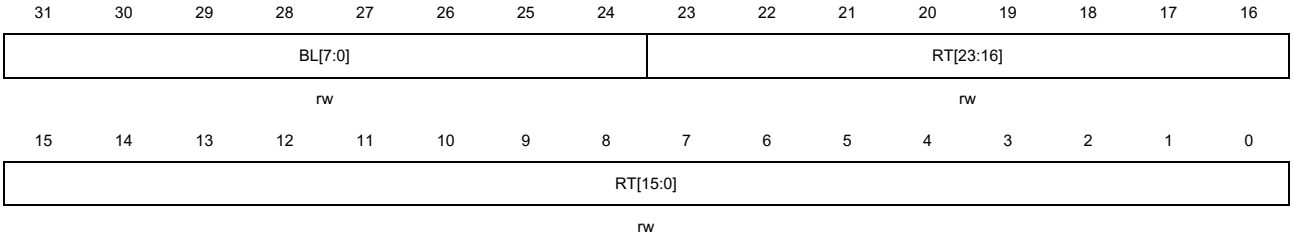
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	GUAT[7:0]	Guard time value in smartcard mode. This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock. In IrDA Low-power mode, the division factor is the prescaler value.

00000000: Reserved - do not program this value.
00000001: divides the source clock by 1.
00000010: divides the source clock by 2.
...
In IrDA normal mode,
00000001: can be set this value only.
In smartcard mode, the prescaler value for dividing the system clock is stored in PSC[4:0] bits. And the bits of PSC[7:5] must be kept at reset value. The division factor is twice as the prescaler value.
00000: Reserved - do not program this value.
00001: divides the source clock by 2.
00010: divides the source clock by 4.
00011: divides the source clock by 6.
...
This bit field cannot be written when the USART is enabled (UEN=1).

Receiver timeout register (USART_RT)

Address offset: 0x14
Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:24	BL[7:0]	Block Length. These bits specify the block length in smartcard T=1 reception. Its value equals the number of information characters + the length of the Epilogue Field (1-LEC/2-CRC) - 1. This value, which must be programmed only once per received block, can be programmed after the start of the block reception (using the data from the LEN character in the Prologue Field). The block length counter is reset when TBE=0 or TFE =0 (FIFO is enabled) in smartcard mode. In other modes, when REN=0 (receiver disabled) and/or when the EBC bit is written to 1, the Block length counter is reset.
23:0	RT[23:0]	Receiver timeout threshold. These bits are used to specify receiver timeout value in terms of number of baud clocks.

In standard mode, the RTF flag is set if no new start bit is detected for more than the RT value after the last received character.

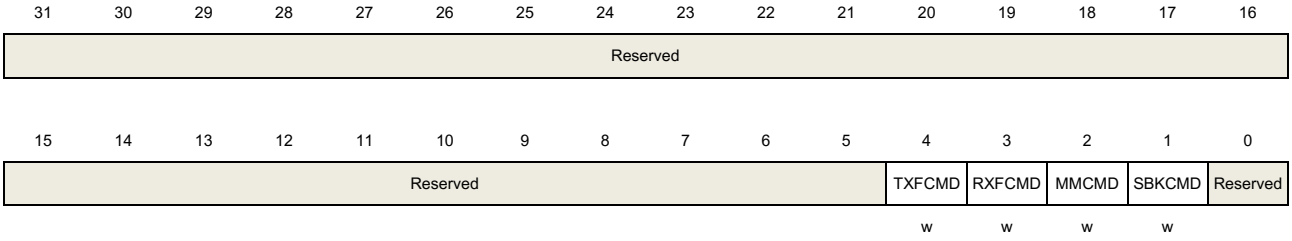
In smartcard mode, the CWT and BWT are implemented by this value. In this case, the timeout measurement is started from the start bit of the last received character. These bits can be written on the fly. The RTF flag will be set if the new value is lower than or equal to the counter. These bits must only be programmed once per received character.

Command register (USART_CMD)

Address offset: 0x18

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TXFCMD	Transmit data flush request When FIFO is disabled, writing 1 to this bit sets the TBE flag, to discard the transmit data. When FIFO is enabled, writing 1 to this bit flushes the whole Transmit FIFO and sets TFE flag in USART_RFCS. **Note:** When FIFO is enabled, during the flush command, TFNF is reset until Transmit FIFO is empty.
3	RXFCMD	Receive data flush command When FIFO is disabled, writing 1 to this bit clears the RBNE flag to discard the received data without reading it. When FIFO is enabled, writing 1 to this bit empties the Receive FIFO and sets RFNE flag in USART_RFCS.
2	MMCMD	Mute mode command Writing 1 to this bit makes the USART into mute mode and sets the RWU flag.
1	SBKCMD	Send break command Writing 1 to this bit sets the SBF flag and makes the USART send a BREAK frame, as soon as the transmit machine is idle.
0	Reserved	Must be kept at reset value.

Status register (USART_STAT)

Address offset: 0x1C

Reset value: 0x0000 00C0

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									REA	TEA	WUF	RWU	SBF	AMF0	BSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		AMF1	EBF	RTF	CTS	CTSF	LBDF	TBE / TFNF	TC	RBNE / RFNE	IDLEF	ORERR	NERR	FERR	PERR
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	REA	Receive enable acknowledge flag This bit, which is set / reset by hardware, reflects the receive enable state of the USART core logic. 0: The USART core receiving logic has not been enabled 1: The USART core receiving logic has been enabled
21	TEA	Transmit enable acknowledge flag This bit, which is set / reset by hardware, reflects the transmit enable state of the USART core logic. 0: The USART core transmitting logic has not been enabled 1: The USART core transmitting logic has been enabled
20	WUF	Wakeup from deep-sleep mode and deep-sleep 1 flag 0: No wakeup from deep-sleep and deep-sleep 1 mode 1: Wakeup from deep-sleep and deep-sleep 1 mode. An interrupt is generated if WUFIE = 1 in the USART_CTL2 register and the MCU is in deep-sleep mode and deep-sleep1 mode. This bit is set by hardware when a wakeup event, which is defined by the WUM bit field, is detected. Cleared by writing a 1 to the WUC in the USART_INTC register. This bit can also be cleared when UESM is cleared.
19	RWU	Receiver wakeup from mute mode This bit is used to indicate if the USART is in mute mode. 0: Receiver in active mode 1: Receiver in mute mode It is cleared/set by hardware when a wakeup/mute sequence (address or IDLEIE) is recognized, which is selected by the WM bit in the USART_CTL0 register. This bit can only be set by writing 1 to the MMCMD bit in the USART_CMD register

		when wakeup on IDLEIE mode is selected. **Note:** RWU is only effective when UEN=1, and this bit remains 0 when UEN=0.
18	SBF	Send break flag 0: No break character is transmitted 1: Break character will be transmitted This bit indicates that a send break character was requested. Set by software, by writing 1 to the SBKCMD bit in the USART_CMD register. Cleared by hardware during the stop bit of break transmission.
17	AMF0	ADDR0 character match flag 0: ADDR0 character does not match the received character 1: ADDR0 character matches the received character, an interrupt is generated if AMIE0=1 in the USART_CTL0 register. Set by hardware, when the character defined by ADDR0[7:0] is received. Cleared by writing 1 to the AMC in the USART_INTC register.
16	BSY	Busy flag 0: USART reception path is idle 1: USART reception path is working
15:14	Reserved	Must be kept at reset value.
13	AMF1	ADDR1 character match flag 0: ADDR1 character does not match the received character 1: ADDR1 character matches the received character, an interrupt is generated if AMIE1 = 1 in the USART_CTL0 register. Set by hardware, when the character defined by ADDR1[7:0] is received. Cleared by writing 1 to the AMC in the USART_INTC register.
12	EBF	End of block flag 0: End of Block not reached 1: End of Block (number of characters) reached. An interrupt is generated if the EBIE = 1 in the USART_CTL1 register Set by hardware when the number of received bytes (from the start of the block, including the prologue) is equal or greater than BLEN + 4. Cleared by writing 1 to EBC bit in USART_INTC register.
11	RTF	Receiver timeout flag 0: Timeout value not reached 1: Timeout value reached without any data reception. An interrupt is generated if RTIE bit in the USART_CTL1 register is set. Set by hardware when the RT value, programmed in the USART_RT register has lapsed without any communication. Cleared by writing 1 to RTC bit in USART_INTC register. The timeout corresponds to the CWT or BWT timings in smartcard mode.

10	CTS	CTS level This bit equals to the inverted level of the nCTS input pin. 0: nCTS input pin is in high level 1: nCTS input pin is in low level
9	CTSF	CTS change flag 0: No change occurred on the nCTS status line 1: A change occurred on the nCTS status line. An interrupt will occur if the CTSIE bit is set in USART_CTL2 Set by hardware when the nCTS input toggles. Cleared by writing 1 to CTSC bit in USART_INTC register.
8	LBDF	LIN break detected flag 0: LIN Break is not detected 1: LIN Break is detected. An interrupt will occur if the LBDIE bit is set in USART_CTL1 Set by hardware when the LIN break is detected. Cleared by writing 1 to LBDC bit in USART_INTC register.
7	TBE	When FIFO is disabled: Transmit data register empty 0: Data is not transferred to the shift register 1: Data is transferred to the shift register. An interrupt will occur if the TBEIE bit is set in USART_CTL0 Set by hardware when the content of the USART_TDATA register has been transferred into the transmit shift register or writing 1 to TXFCMD bit of the USART_CMD register. Cleared by a write to the USART_TDATA.
	TFNF	When FIFO is enabled: Transmit FIFO not full 0: Transmit FIFO is full 1: Transmit FIFO is not full. An interrupt will occur if the TFNFIE bit is set in USART_CTL0. Set by hardware when the transmit FIFO is not full. The flag is cleared when the transmit FIFO is full. **Note:** The TFNF bit keep reset during the TXFCMD set until Transmit FIFO is empty.
6	TC	Transmission completed 0: Transmission is not completed 1: Transmission is complete. An interrupt will occur if the TCIE bit is set in USART_CTL0. Set by hardware if the transmission of a frame containing data is completed and if the TBE or TFE bit in USART_RFCS is set. Cleared by writing 1 to TCC bit in USART_INTC register.

		Note: The TC bit is set immediately when TEN is reset and no transmission is on going.
5	RBNE	When FIFO is disabled: Read data buffer not empty 0: Data is not received 1: Data is received and ready to be read. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. Set by hardware when the content of the receive shift register has been transferred to the USART_RDATA. Cleared by reading the USART_RDATA or writing 1 to RXFCMD bit of the USART_CMD register.
	RFNE	When FIFO is enabled: Receive FIFO not empty 0: Receive FIFO is empty 1: Receive FIFO is not empty. An interrupt will occur if the RFNEIE bit is set in USART_CTL0. Set by hardware when the receive FIFO is not empty. The flag is cleared when the receive FIFO is empty. The bit can also be reset by setting RXFCMD bit in USART_CMD.
4	IDLEF	IDLE line detected flag 0: No Idle line is detected 1: Idle line is detected. An interrupt will occur if the IDLEIE bit is set in USART_CTL0 Set by hardware when an Idle line is detected. It will not be set again until the RBNE bit or RFNE bit has been set itself. Cleared by writing 1 to IDLEC bit in USART_INTC register.
3	ORERR	Overrun error 0: No overrun error is detected 1: Overrun error is detected. An interrupt will occur if the RBNEIE or RFNEIE bit is set in USART_CTL0. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when the word in the receive shift register is ready to be transferred into the USART_RDATA register while the RBNE bit or RFF bit in USART_RFCS is set. Cleared by writing 1 to OREC bit in USART_INTC register.
2	NERR	Noise error flag 0: No noise error is detected 1: Noise error is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when noise error is detected on a received frame. Cleared by writing 1 to NEC bit in USART_INTC register. **Note:** When this bit and RBNE or RFNE bit appears at the same time, it does not

		generate an interrupt. When FIFO is enabled, the error is associated with the data in the USART_RDATA.
1	FERR	Frame error flag 0: No framing error is detected 1: Frame error flag or break character is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when a de-synchronization, excessive noise or a break character is detected. This bit will be set when the maximum number of transmit attempts is reached without success (the card NACKs the data frame), when USART transmits in smartcard mode. Cleared by writing 1 to FEC bit in USART_INTC register. **Note:** When FIFO is enabled, the error is associated with the data in the USART_RDATA.
0	PERR	Parity error flag 0: No parity error is detected 1: Parity error flag is detected. An interrupt will occur if the PERRIE bit is set in USART_CTL0. Set by hardware when a parity error occurs in receiver mode. Cleared by writing 1 to PEC bit in USART_INTC register. **Note:** When FIFO is enabled, the error is associated with the data in the USART_RDATA.

Interrupt status clear register (USART_INTC)

Address offset: 0x20

Reset value: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											WUC	Reserved		AMC0	AMC1
											w			w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EBC	RTC	Reserved	CTSC	LBDC	Reserved	TCC	Reserved	IDLEC	OREC	NEC	FEC	PEC
			w	w		w	w		w		w	w	w	w	w

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	WUC	Wakeup from deep-sleep mode clear Writing 1 to this bit clears the WUF bit in the USART_STAT register.
19:18	Reserved	Must be kept at reset value.
17	AMC0	ADDR0 character match clear

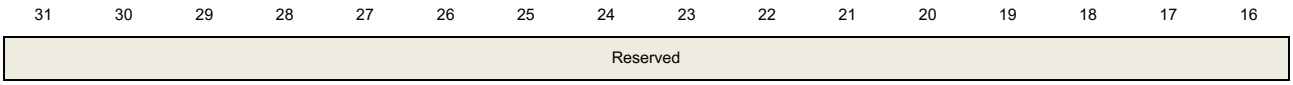
		Writing 1 to this bit clears the AMF0 bit in the USART_STAT register.
16	AMC1	ADDR1 character match clear Writing 1 to this bit clears the AMF1 bit in the USART_STAT register.
15:13	Reserved	Must be kept at reset value.
12	EBC	End of block clear Writing 1 to this bit clears the EBF bit in the USART_STAT register.
11	RTC	Receiver timeout clear Writing 1 to this bit clears the RTF flag in the USART_STAT register.
10	Reserved	Must be kept at reset value.
9	CTSC	CTS change clear Writing 1 to this bit clears the CTSF bit in the USART_STAT register.
8	LBDC	LIN break detected clear Writing 1 to this bit clears the LBDF flag in the USART_STAT register.
7	Reserved	Must be kept at reset value.
6	TCC	Transmission complete clear Writing 1 to this bit clears the TC bit in the USART_STAT register.
5	Reserved	Must be kept at reset value.
4	IDLEC	Idle line detected clear Writing 1 to this bit clears the IDLEF bit in the USART_STAT register.
3	OREC	Overrun error clear Writing 1 to this bit clears the ORERR bit in the USART_STAT register.
2	NEC	Noise detected clear Writing 1 to this bit clears the NERR bit in the USART_STAT register.
1	FEC	Frame error flag clear Writing 1 to this bit clears the FERR bit in the USART_STAT register
0	PEC	Parity error clear Writing 1 to this bit clears the PERR bit in the USART_STAT register.

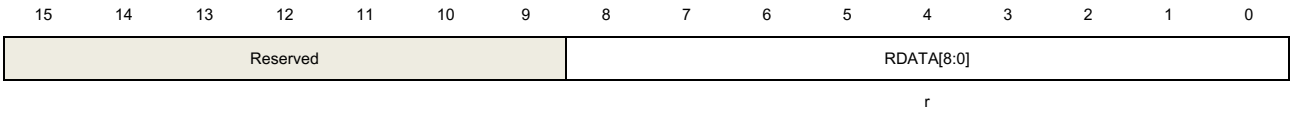
Receive data register (USART_RDATA)

Address offset: 0x24

Reset value: Undefined

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





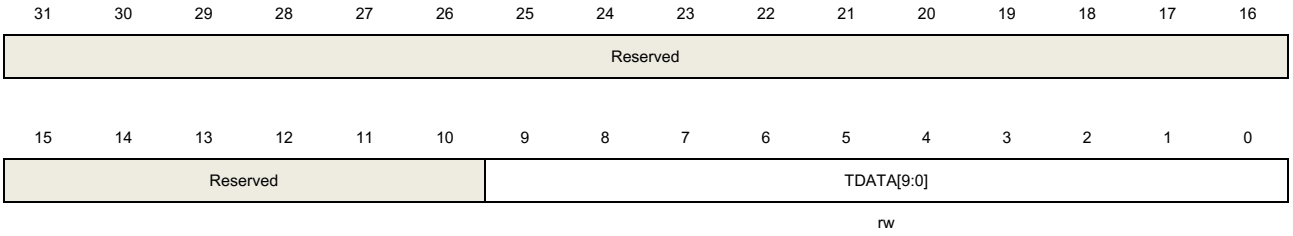
Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8:0	RDATA[8:0]	Receive data value The received data character is contained in these bits. The value read in the MSB (bit 7 or bit 8 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).

Transmit data register (USART_TDATA)

Address offset: 0x28

Reset value: Undefined

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



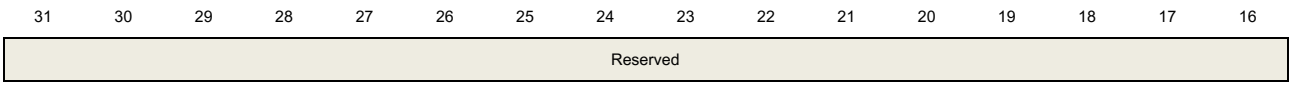
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9:0	TDATA[9:0]	Transmit Data value The transmit data character is contained in these bits. The value written in the MSB (bit6, bit7, bit8 or bit9 depending on the data length) will be replaced by the parity, when transmitting with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register). This register must be written only when TBE bit in USART_STAT register is set.

USART coherence control register (USART_CHC)

Address offset: 0xC0

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
Reserved							EPERR	Reserved							HCM
							rc_w0								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.
7:1	Reserved	Must be kept at reset value.
0	HCM	Hardware flow control coherence mode 0: nRTS signal equals to the RBNE in status register 1: nRTS signal is set when the last data bit (parity bit when pce is set) has been sampled.

USART FIFO control and status register (USART_FCS)

Address offset: 0xD0
Reset value: 0x0300 0440

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TFEIE	Reserved	TFTIE	Reserved	RFTIE	TFEC	TFTIF	TFEIF	Reserved	RFTIF	TFTCFG[2:0]			RFTCFG[2:0]		
rw		rw		rw	rw	r	r		rc_w0	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFIF	RFCNT[2:0]			RFF	RFE	RFFIE	FEN	TFF	TFE	TFT	RFT	Reserved	RFCNT[4:3]		ELNACK
rc_w0	r			r	r	rw	rw	r	r	r	r		r		rw

Bits	Fields	Descriptions
31	TFEIE	Transmit FIFO empty interrupt enable If this bit is set, an interrupt occurs whenever the TFE bit is set. 0: Transmit FIFO empty interrupt is disabled 1: Transmit FIFO empty interrupt is enabled
30	Reserved	Must be kept at reset value.
29	TFTIE	Transmit FIFO threshold interrupt enable If this bit is set, an interrupt will occur whenever the available space in transmit FIFO reached the threshold configured in TFTCFG[2:0]. 0: Transmit FIFO threshold interrupt is disabled 1: Transmit FIFO threshold interrupt is enabled

28	Reserved	Must be kept at reset value.
27	RFTIE	Receive FIFO threshold interrupt enable If this bit is set, an interrupt will occur whenever the receive FIFO reached the threshold configured in RFTCFG. 0: Receive FIFO threshold interrupt is disabled 1: Receive FIFO threshold interrupt is enabled
26	TFEC	Transmit FIFO empty flag clear Writing 1 to this bit clears the TFE flag.
25	TFTIF	Transmit FIFO threshold interrupt flag The bit is valid when TFTIE bit is set.. 0: The available space in transmit FIFO does not reach the programmed threshold 1: The available space in transmit FIFO reached the programnmed threshold. An interrupt will occur if the TFTIE bit is set. Set by hardware when the the available space in transmit FIFO reaches the threshold configured in TFTCFG[2:0].
24	TFEIF	Transmit FIFO empty interrupt flag The bit is valid when TFEIE bit is set. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty
23	Reserved	Must be kept at reset value.
22	RFTIF	Receive FIFO threshold interrupt flag The bit is valid when RFTIE bit is set. 0: Receive FIFO does not reach the programmed threshold 1: Receive FIFO reached the programmed threshold
21:19	TFTCFG[2:0]	Transmit FIFO threshold configuration 000:Transmit FIFO reaches 1/8 of its depth 001:Transmit FIFO reaches 1/4 of its depth 010:Transmit FIFO reaches 1/2 of its depth 011:Transmit FIFO reaches 3/4 of its depth 100:Transmit FIFO reaches 7/8 of its depth 101:Transmit FIFO becomes empty 11x: Reserved
18:16	RFTCFG[2:0]	Receive FIFO threshold configuration 000:Receive FIFO reaches 1/8 of its depth 001:Receive FIFO reaches 1/4 of its depth 010:Receive FIFO reaches 1/2 of its depth 011:Receive FIFO reaches 3/4 of its depth 100:Receive FIFO reaches 7/8 of its depth 101:Receive FIFO becomes full

		11x: Reserved
15	RFFIF	Receive FIFO full interrupt flag The bit is valid when RFFIE bit is set. 0: Receive FIFO is not full 1: Receive FIFO is full
14:12	RFCNT[2:0]	Receive FIFO counter number These bits and RFCNT[4:3] bits determine the receive FIFO counter number.
11	RFF	Receive FIFO full flag 0: Receive FIFO is not full 1: Receive FIFO is full. An interrupt will occur if the RFFIE bit is set. Set by hardware when the number of received data is RXFIFO size + 1.
10	RFE	Receive FIFO empty flag 0: Receive FIFO is not empty 1: Receive FIFO is empty.
9	RFFIE	Receive FIFO full interrupt enable If this bit is set, an interrupt occurs when the RFF bit is set. 0: Receive FIFO full interrupt is disable 1: Receive FIFO full interrupt is enable
8	FEN	FIFO enable 0: FIFO is disable 1: FIFO is enable This bit field cannot be written when the USART is enabled (UEN = 1). Note: Do not change the FEN bit when receiving or transmitting data is not accomplished. When UEN is cleared and reconfigure the UEN without changing the FEN bit, please flush the FIFO if do not need the pervious FIFO value.
7	TFF	Transmit FIFO full flag 0: Transmit FIFO is not full 1: Transmit FIFO is full
6	TFE	Transmit FIFO empty flag 0: Transmit FIFO is not empty 1: Transmit FIFO is empty. An interrupt will occur if the TFEIE bit is set. Set by hardware when the transmit FIFO is empty. The flag is cleared when the transmit FIFO has at least one data. The bit can also be set by setting TXFCMD bit in USART_CMD.
5	TFT	Transmit FIFO threshold flag 0: The available space in transmit FIFO reached the programmed threshold. 1: The available space in transmit FIFO does not reach the programmed threshold. Reset by hardware when the available space in transmit FIFO reaches the threshold

		configured in TFTCFG[2:0].
4	RFT	Receive FIFO threshold flag 0: Receive FIFO does not reach the programmed threshold 1: Receive FIFO reached the programmed threshold. An interrupt will occur if the RFTIE bit is set. Set by hardware when the receive FIFO reaches the threshold configured in RFTCFG[2:0]. This means that there are (RFTCFG[2:0] - 1) data in the Receive FIFO and one data in the USART_RDATA register **Note:** When the RFTCFG[2:0] = 0b101 and 16 data are available, RFT flag will be set.
3	Reserved	Must be kept at reset value.
2:1	RFCNT[4:3]	Receive FIFO counter number These bits and RFCNT[2:0] bits determine the receive FIFO counter number.
0	ELNACK	Early NACK when smartcard mode is selected. The NACK pulse occurs 1/16 bit time earlier when the parity error is detected. 0:Early NACK disable when smartcard mode is selected 1:Early NACK enable when smartcard mode is selected

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, 1 with configurable mask).
- Programmable setup time and hold time.
- Multi-master capability.
- Supports standard mode (up to 100 kHz), 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 of the internal configuration of the I2C interface.

Figure 25-1. I2C module block diagram

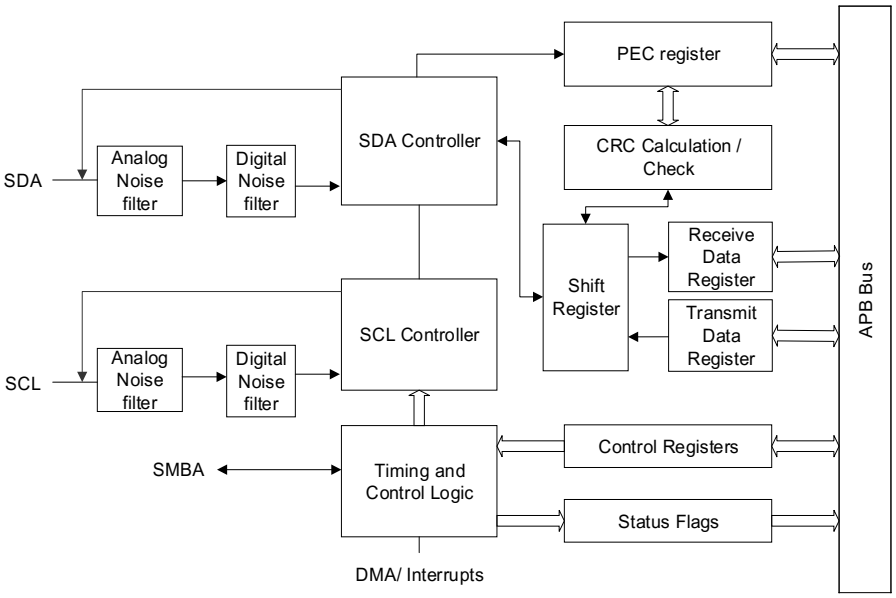


Table 25-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

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 I2C2 clock is independent of the PCLK frequency, so that the I2C2 can be operated independently.

This I2C2 clock (I2C2CLK) can be selected from the following three clock sources:

- CK_APB1: APB1 clock (default value)
- CK_IRC8M: internal 8 MHz RC
- CK_SYS: system clock

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 150ns. 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

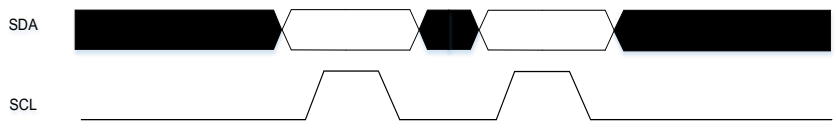
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- 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

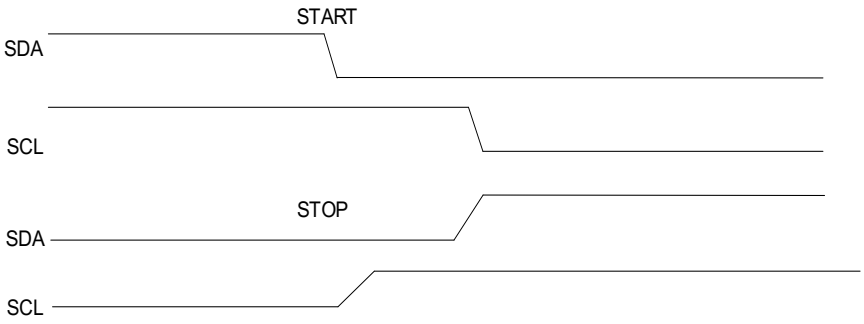


START and STOP signal

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.

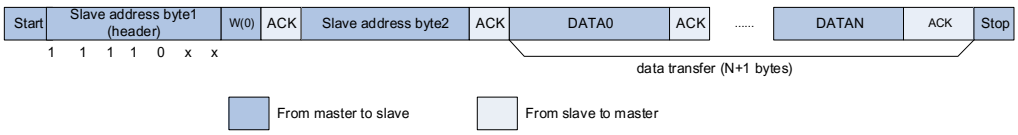
Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the START signal contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

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)

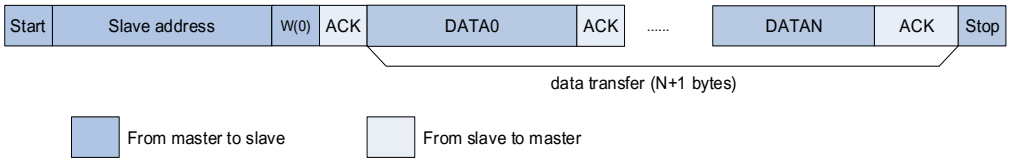
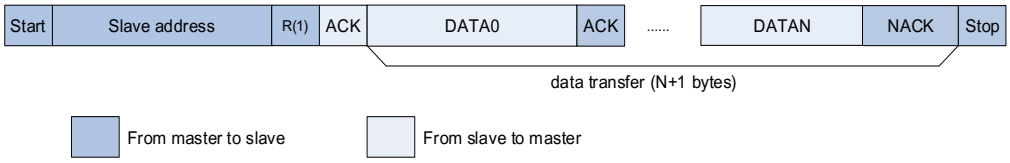


Figure 25-6. I2C communication flow with 7-bit address (Master Receive)



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)

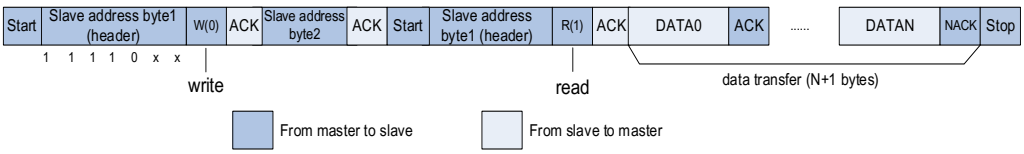
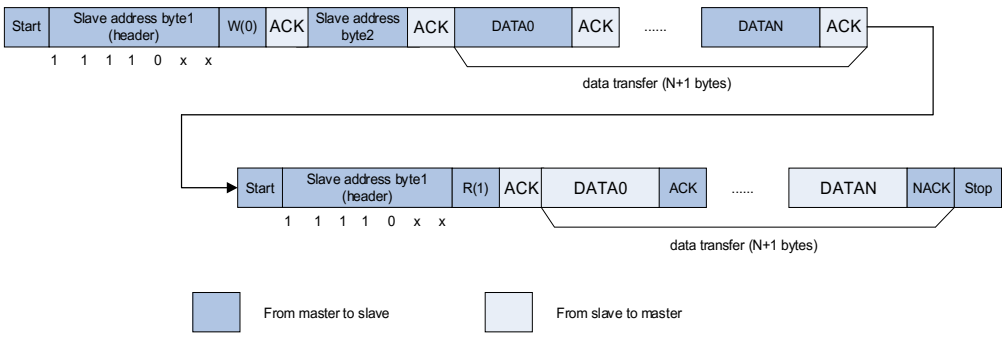


Figure 25-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



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.

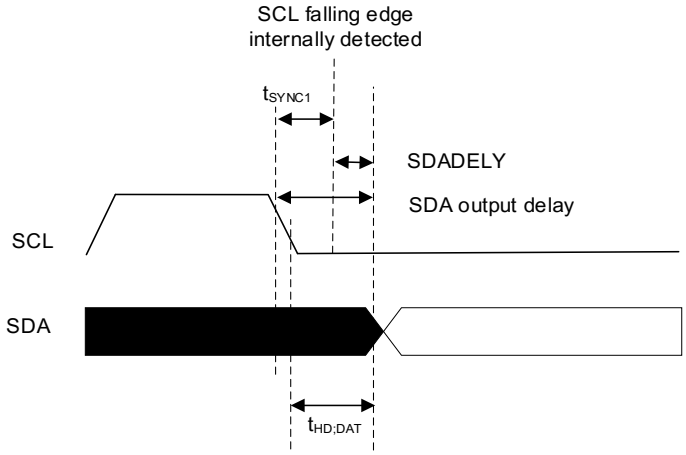
The digital noise filter can be used by configuring the DNF [3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will be changed if the level is stable for more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF [3:0].

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.

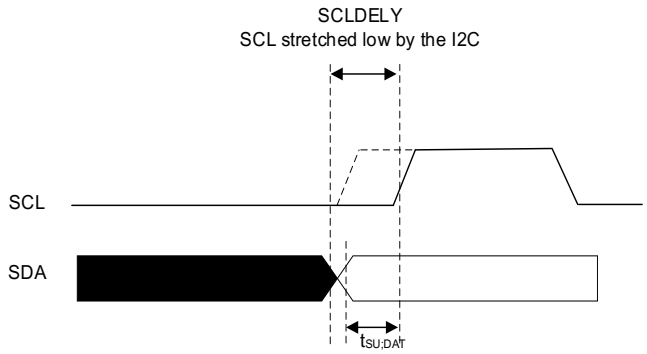
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 25-9. Data hold time***.

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+1)*(PSC+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:

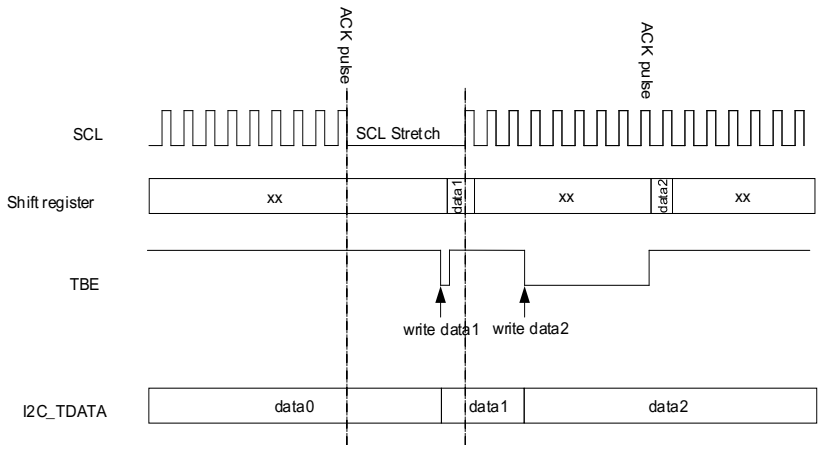
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

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.

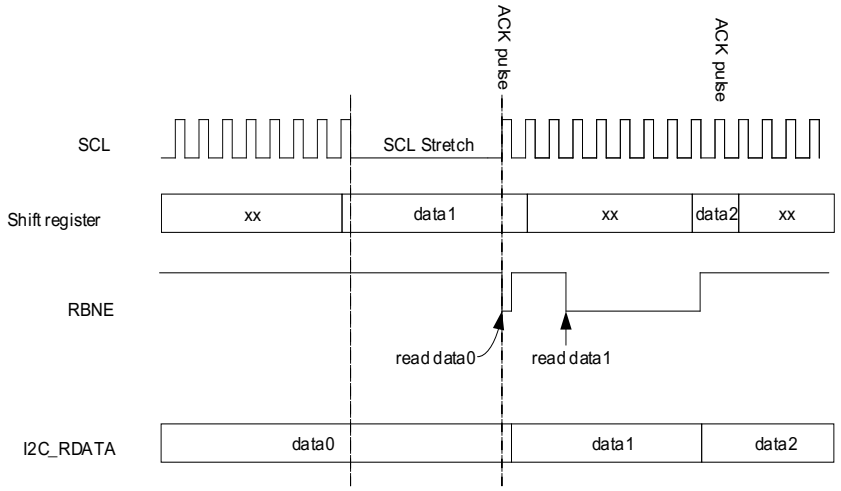
Figure 25-11. Data transmission



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

Working mode	Action
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

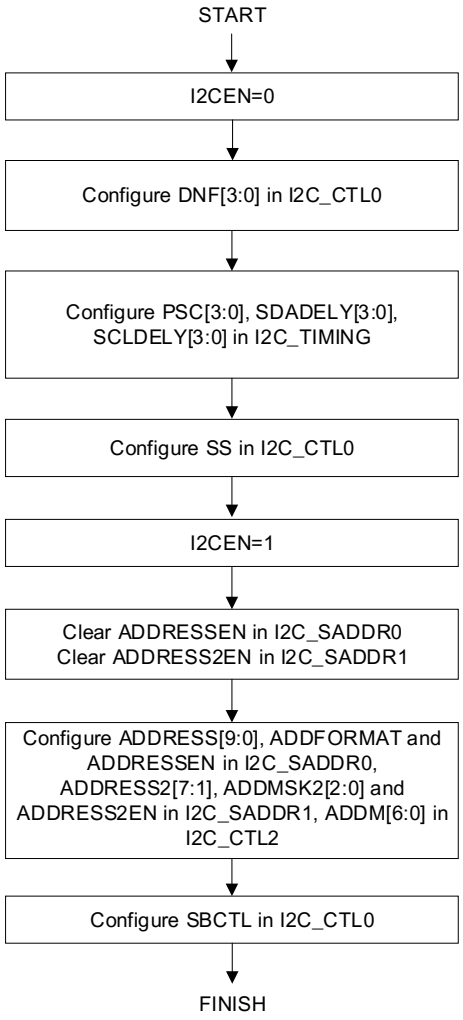
two data bytes.

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 operates in byte control mode. BYTENUM[7:0] must be configured in the ADDSEND interrupt service routine. However, when acting as an SMBus slave transmitter with the PEC function enabled, BYTENUM must be configured before the START bit is set. 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

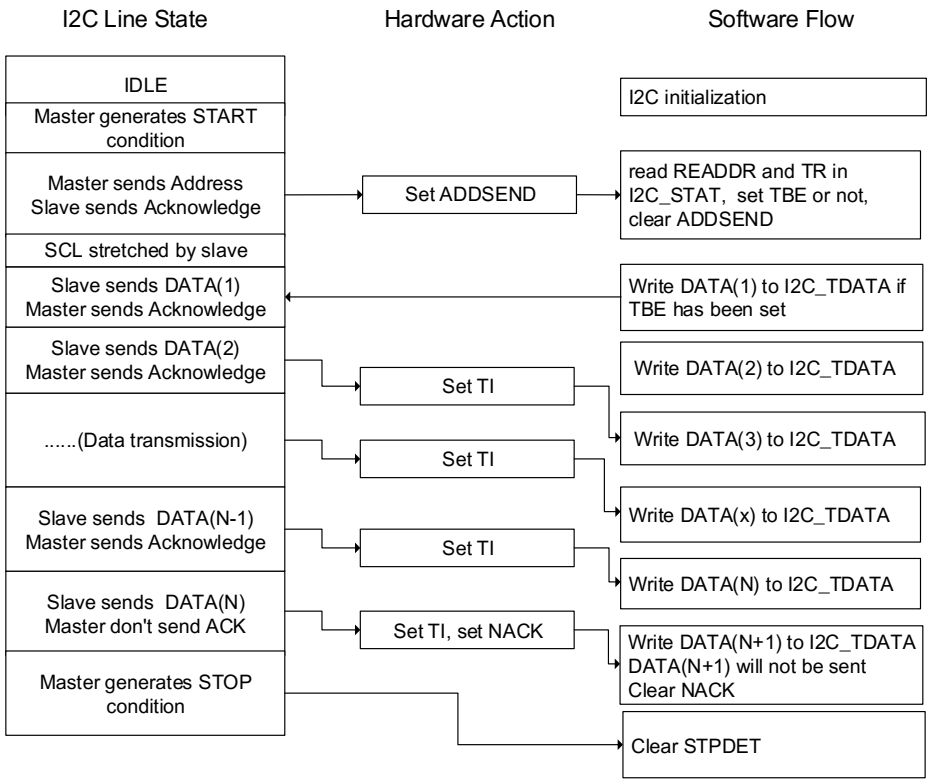
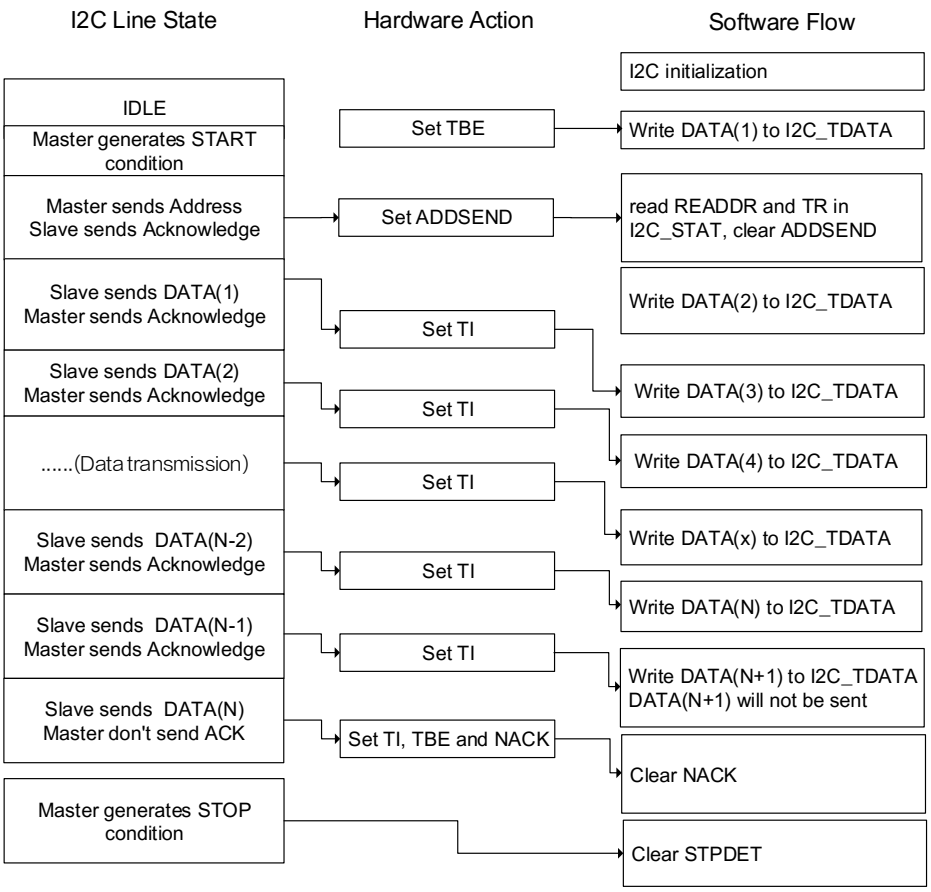


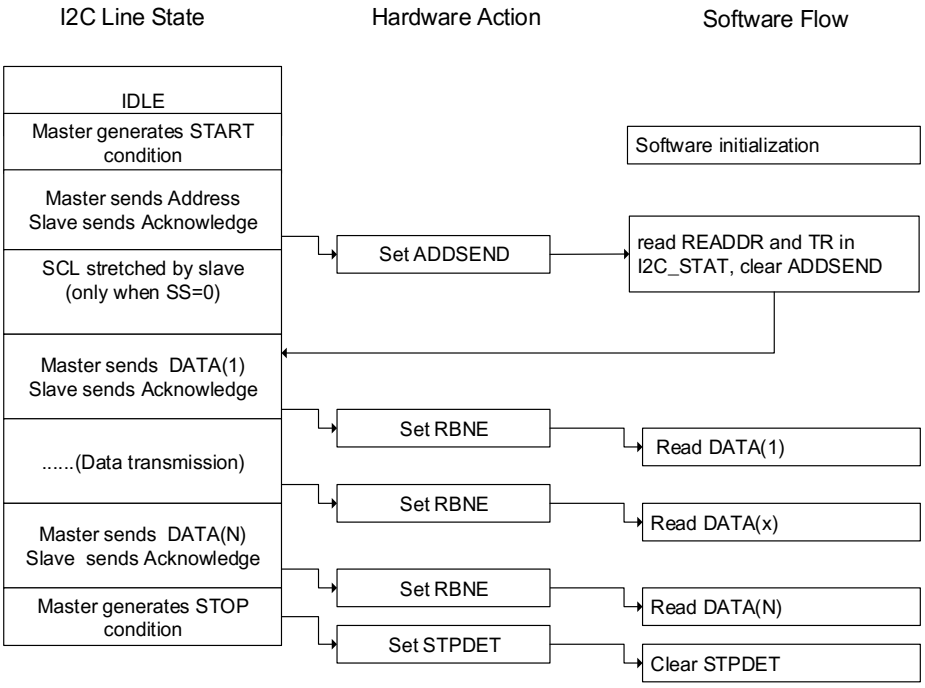
Figure 25-15. Programming model for slave transmitting when SS=1



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:

$$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\} \quad (25\text{-}1)$$

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}.

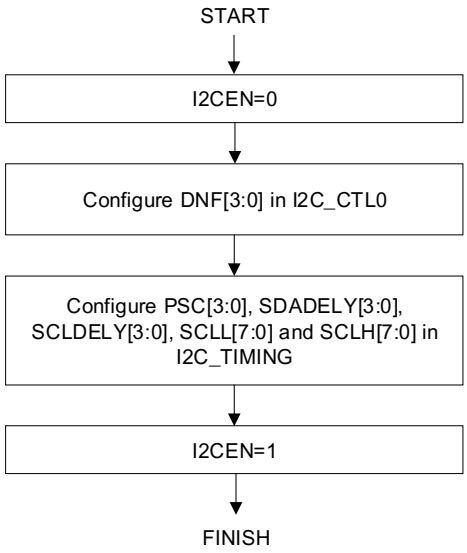
When works in master mode, the ADD10EN bit, SADDRESS [9:0] bits, TRDIR bit should be configured in I2C_CTL1 register. When the addressing mode is 10-bit in master receiving mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to 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)

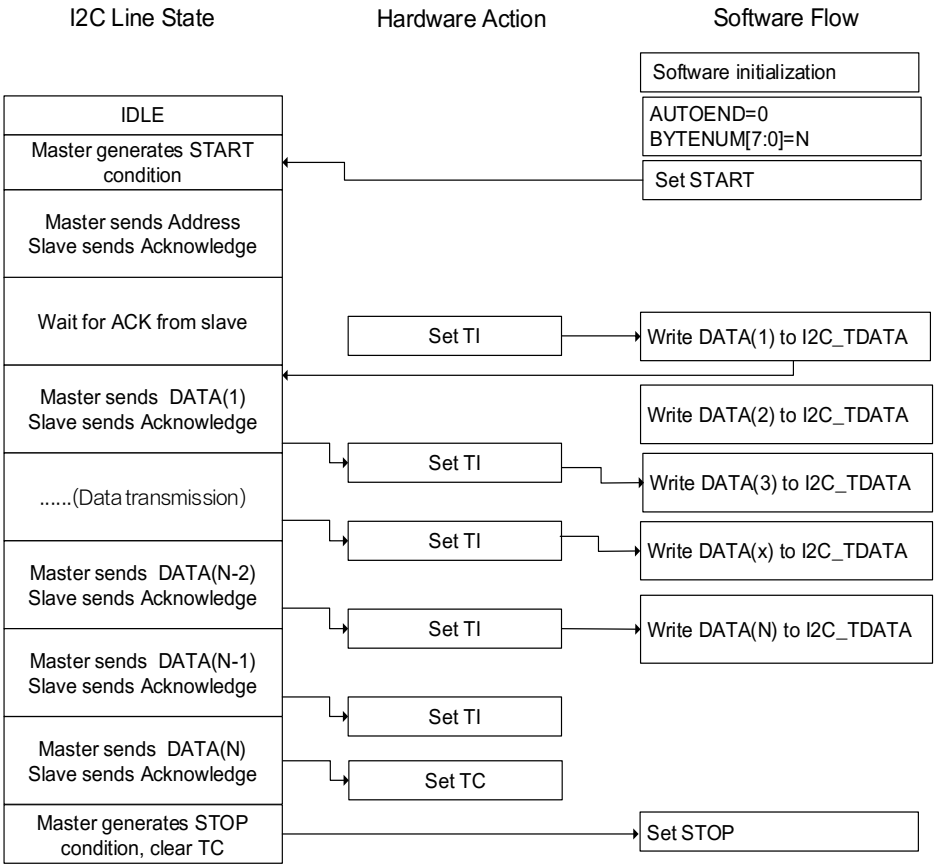
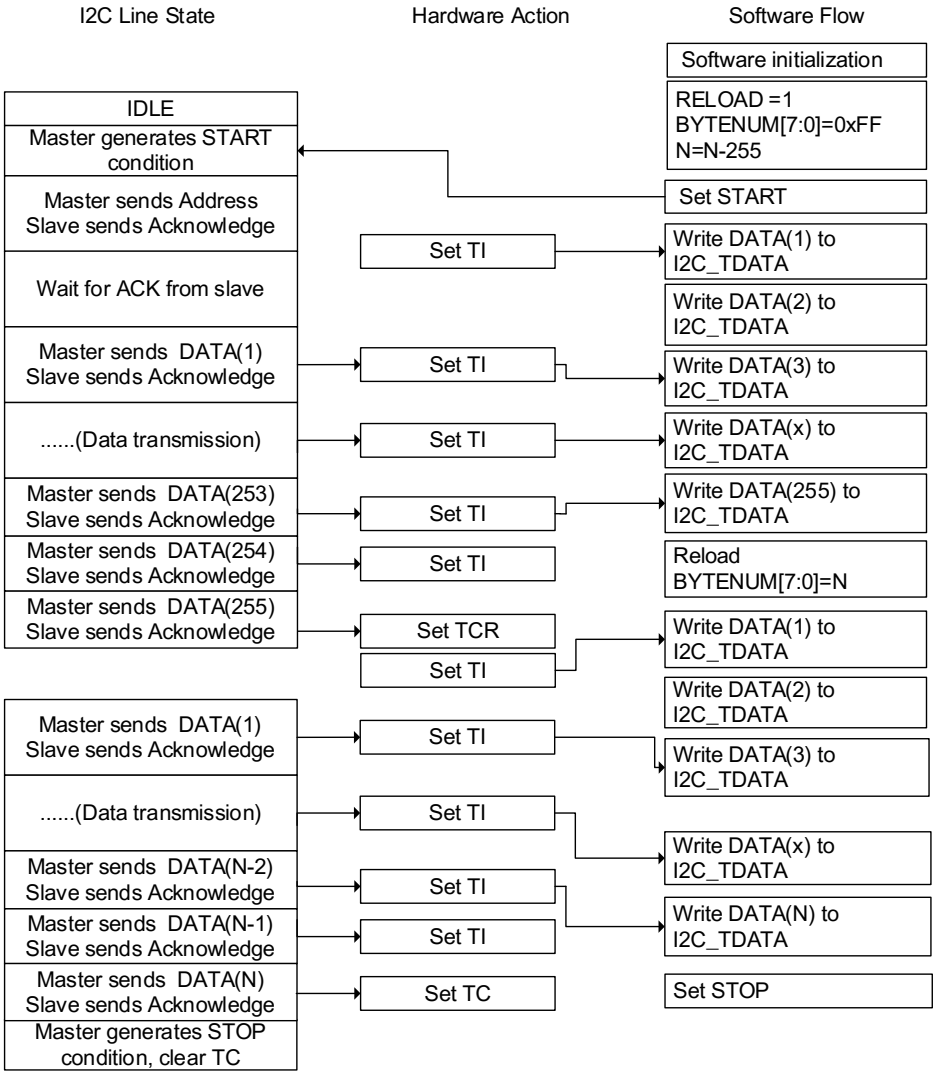


Figure 25-19. Programming model for master transmitting (N>255)



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)

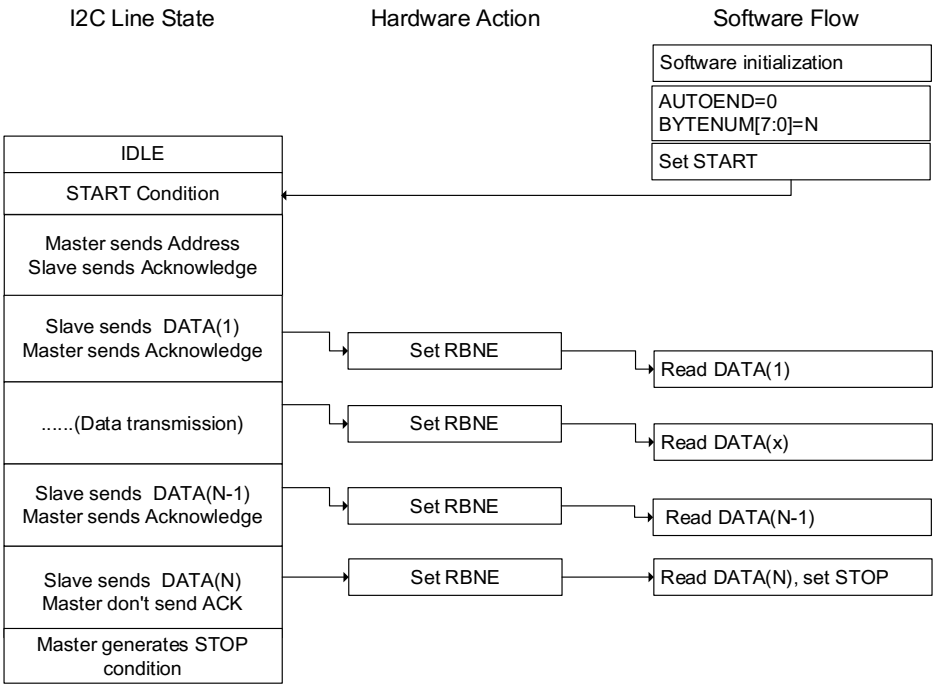
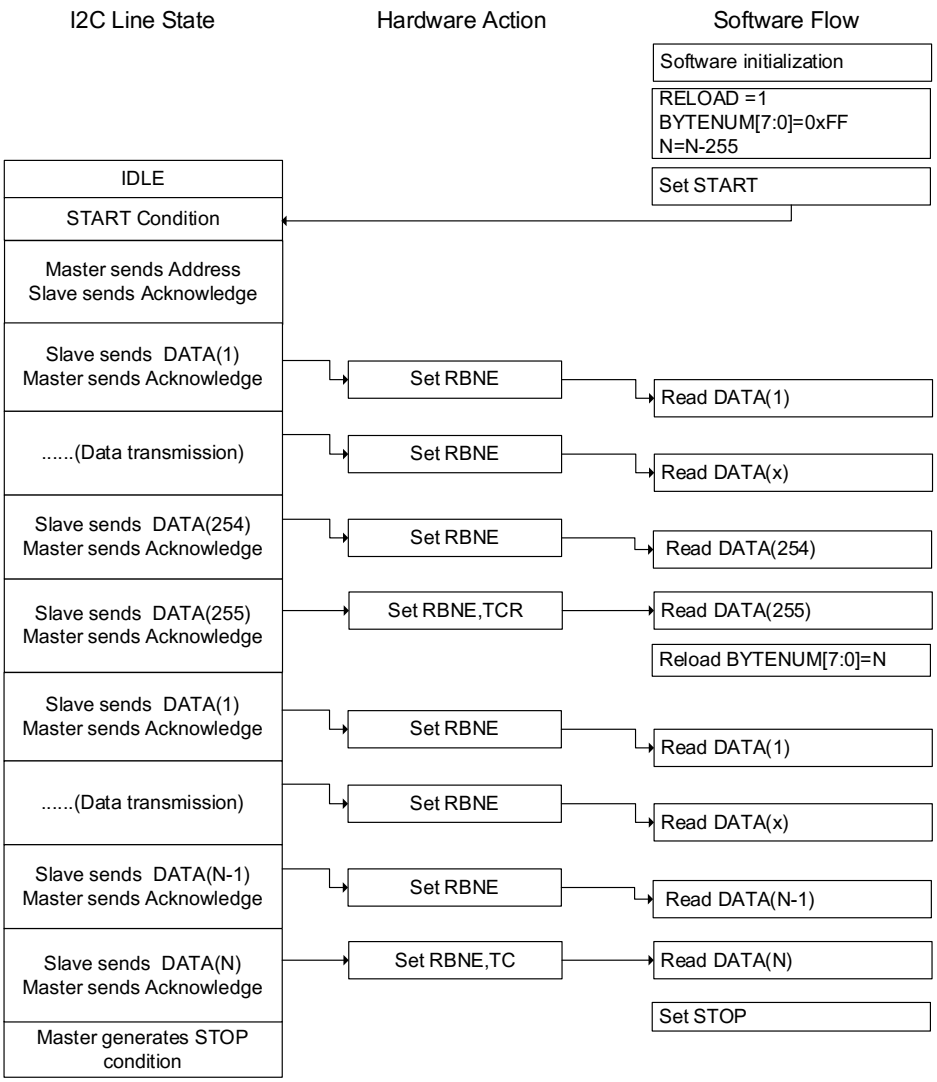


Figure 25-21. Programming model for master receiving (N>255)



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.

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

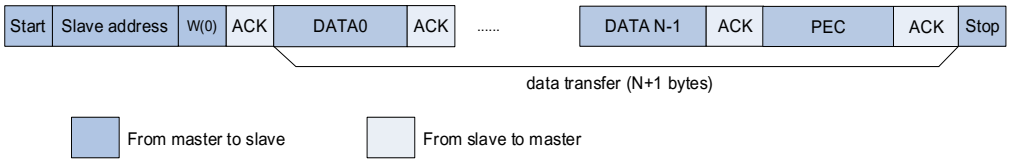
SMBus master transmitter and slave receiver

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 do not match, the NACK is automatically generated. If the PEC values match, 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.

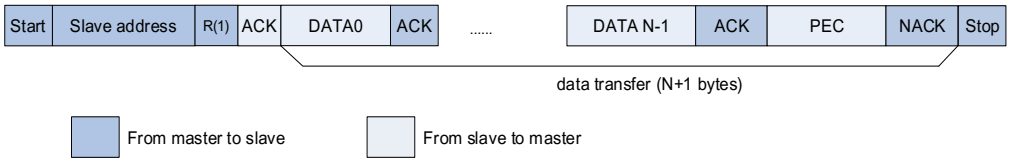
If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, the software mode (AUTOEND = 0) must be selected. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received 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. Wakeup from Deep-sleep mode

When the address of I2C matches correctly, it can wake up from MCU Deep-sleep mode. In order to wake up from Deep-sleep mode and Deep-sleep mode 1, WUEN bit must be set in the I2C_CTL0 register and the CK_IRC8M must be selected as the clock source for I2CCLK. During Deep-sleep mode and Deep-sleep mode 1, the IRC8M is switched off. The I2C

interface switches the CK_IRC8M on, and stretches SCL low until CK_IRC8M is woken up when a START is detected. Then the CK_IRC8M is used as the clock of I2C to receive the address. When address matching is detected, I2C stretches SCL during MCU wake-up. The SCL is released until the software clears the ADDSEND flag and the transmission proceeds normally. If the detected address does not match, CK_IRC8M will be closed again and the MCU will not be wake up.

Only an address match interrupt (ADDRIE=1) can wakeup the MCU. If the clock source of I2C is the system clock, or WUEN = 0, IRC8M will not switched on after receiving start signal. When wakeup from Deep-sleep mode and Deep-sleep mode 1 is enabled, the digital filter must be disabled and the SS bit in I2C_CTL0 must be cleared. Before entering Deep-sleep mode and Deep-sleep mode 1, the I2C peripheral must be disabled if wakeup from Deep-sleep mode and Deep-sleep mode1 is disabled (WUEN =0).

Note: Only address match of I2C2 can wakeup MCU from Deep-sleep mode and Deep-sleep mode 1.

25.3.12. 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.

25.3.13. I2C error and interrupts

The I2C error flags are listed in *Table 25-4. I2C error flags*.

Table 25-4. 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 25-5. I2C interrupt events*.

Table 25-5. 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	

25.3.14. 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

I2C2 base address: 0x4000 C000

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	WUEN	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	WUEN	Wakeup from Deep-sleep mode and Deep-sleep mode 1 enable (only support in I2C2) 0: Wakeup from power saving mode disable. 1: Wakeup from power saving mode enable. **Note:** WUEN can be set only when DNF[3:0] = 0000
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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESSEN	Reserved				ADDFORMAT	ADDRESS[9:8]		ADDRESS[7:1]							ADDRESS0
rw					rw	rw		rw							rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESSEN	I2C address enable 0: I2C address disable. 1: I2C address enable.
14:11	Reserved	Must be kept at reset value.
10	ADDFORMAT	Address mode for the I2C slave 0: 7-bit address 1: 10-bit address **Note:** When ADDRESSEN is set, this bit should not be written.
9:8	ADDRESS[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.

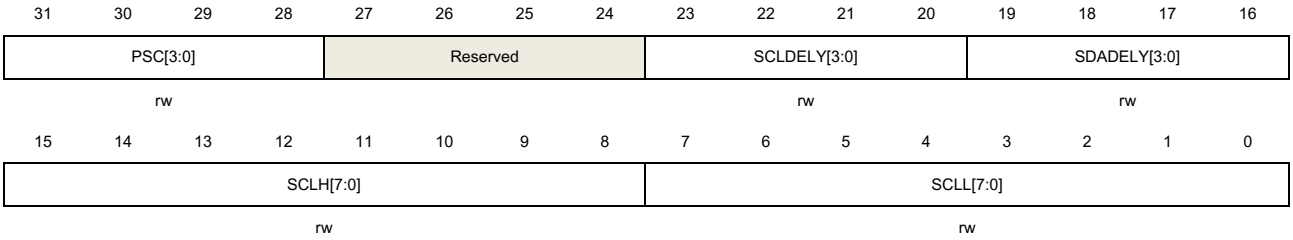
25.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

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

This register cannot be modified when the I2C is enabled (I2CEN=1).



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).

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	Reserved		STPDETC	NACKC	ADDSENDC	Reserved		
		w	w	w	w	w	w			w	w	w			

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit
10	OUERRC	Overrun/Underrun flag clear.

		Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit
9	LOSTARBC	Arbitration Lost flag clear. Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit
7:6	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).

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

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

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

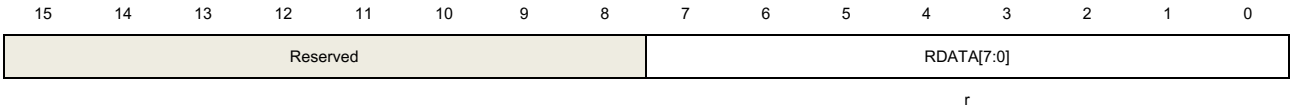
25.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

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

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



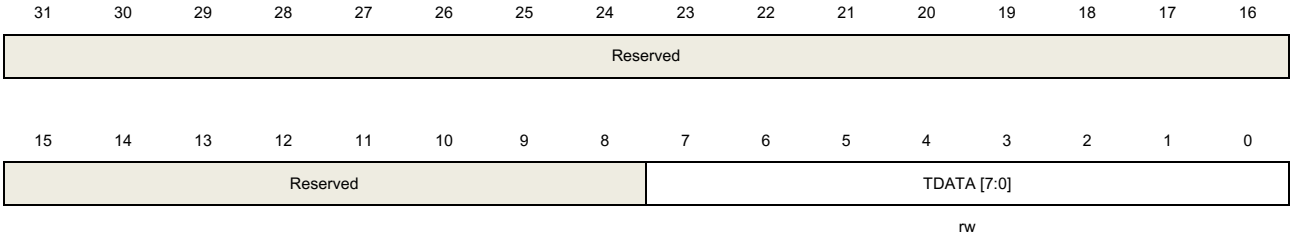
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

25.4.11. Transmit data register (I2C_TDATA)

Address offset: 0x28

Reset value: 0x0000 0000

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



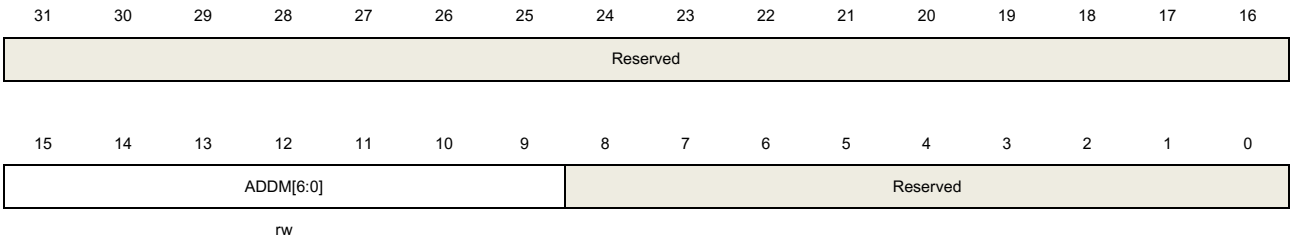
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

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. Serial peripheral interface/Inter-IC sound (SPI/I2S)

26.1. 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 and SPI3.

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. (By using two extra I2S modules called I2S1_ADD and I2S2_ADD, I2S full duplex mode is also supported in SPI1 and SPI2.)

26.2. Characteristics

26.2.1. 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 and SPI3).

26.2.2. I2S characteristics

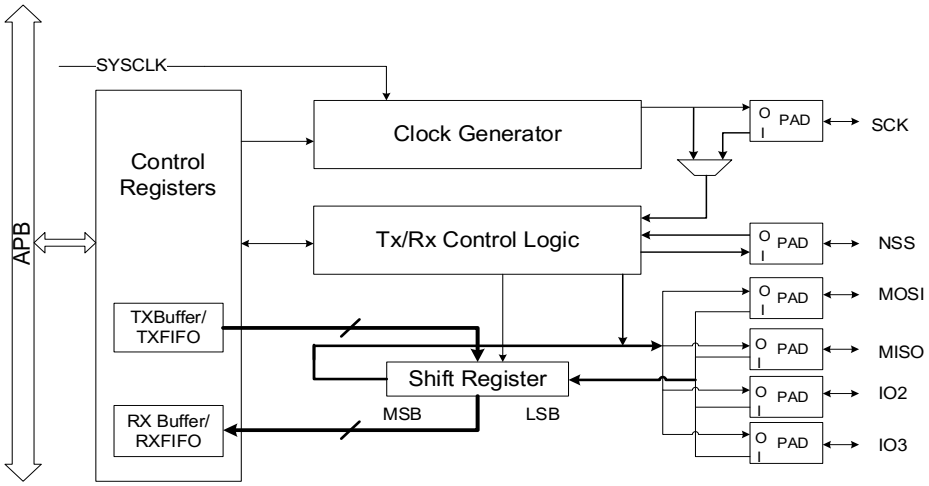
- Master or slave operation with transmission or reception mode.
- Master or slave operation with full-duplex mode(only in SPI1 and SPI2)
- Four I2S standards supported: Philips, MSB justified, LSB justified and PCM standard.
- Data length can be 16 bits, 24 bits or 32 bits.
- Channel length can be 16 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.

26.3. SPI function overview

26.3.1. SPI block diagram

Figure 26-1. Block diagram of SPI



26.3.2. SPI signal description

Normal configuration (Not Quad-SPI Mode)

Table 26-1. 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

Pin Name	Direction	Description
		Master in hardware NSS mode: when NSSDRV=1, it is NSS output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master application. Slave in Hardware NSS Mode: NSS input, as a chip select signal for slave.

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 and SPI3). Quad-SPI mode can only work at master mode.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 26-2. 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

26.3.3. 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 26-2. SPI timing diagram in normal mode

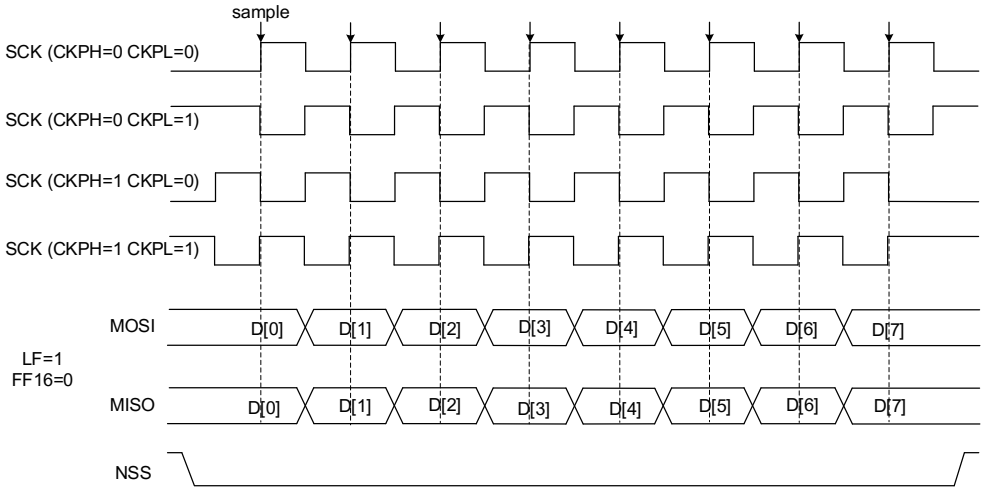
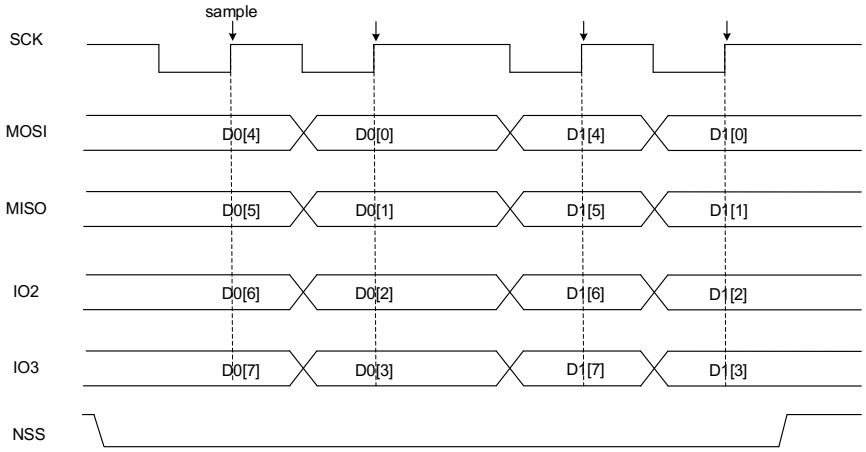


Figure 26-3. SPI timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



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.

26.3.4. 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 26-3. 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 26-4. 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	The slave can use hardware or software NSS mode.

Mode	Register configuration	Description
	SWNSS = 1 NSSDRV: Don't care	

26.3.5. SPI operation modes

Table 26-5. 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 unidirectional connection	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Not used MISO: Reception
MTB	Master Transmission with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: Transmission MISO: Not used
MRB	Master Reception with bidirectional connection	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Reception MISO: Not used
SFD	Slave Full-Duplex	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Transmission
STU	Slave Transmission with unidirectional connection	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Not used MISO: Transmission
SRU	Slave Reception with unidirectional connection	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: Don't care	MOSI: Reception MISO: Not used
STB	Slave Transmission with bidirectional connection	MSTMOD = 0 RO = 0	MOSI: Not used MISO: Transmission

Mode	Description	Register Configuration	Data Pin Usage
		BDEN = 1 BDOEN = 1	
SRB	Slave Reception with bidirectional connection	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI: Not used MISO: Reception

Figure 26-4. A typical Full-duplex connection

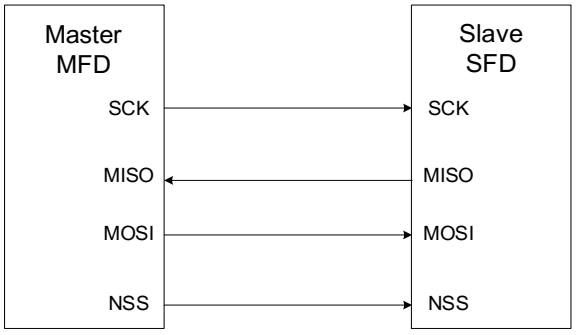


Figure 26-5. A typical simplex connection (Master: Receive, Slave: Transmit)

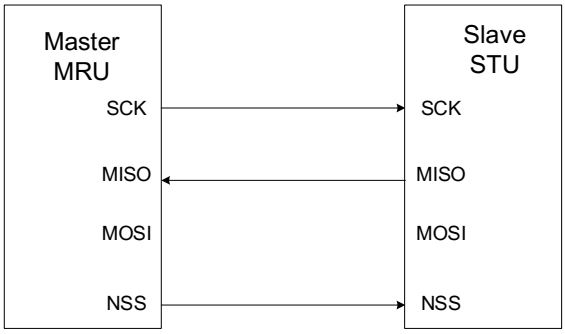


Figure 26-6. A typical simplex connection (Master: Transmit only, Slave: Receive)

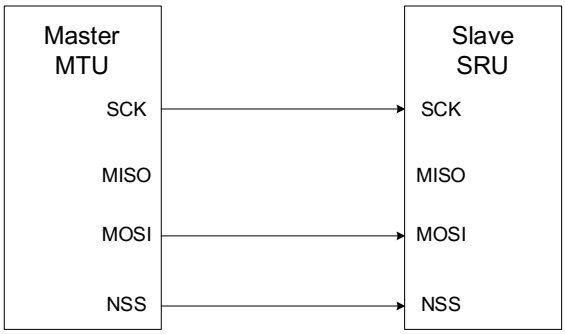
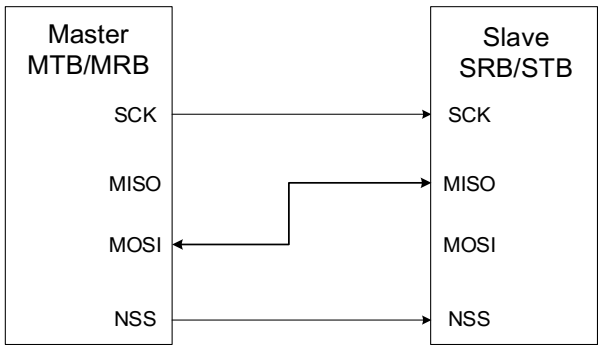


Figure 26-7. A typical bidirectional connection



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 26-8. Timing diagram of TI master mode with discontinuous transfer

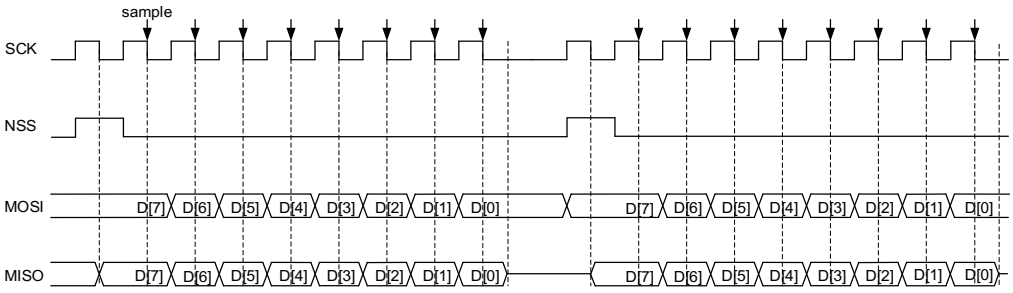
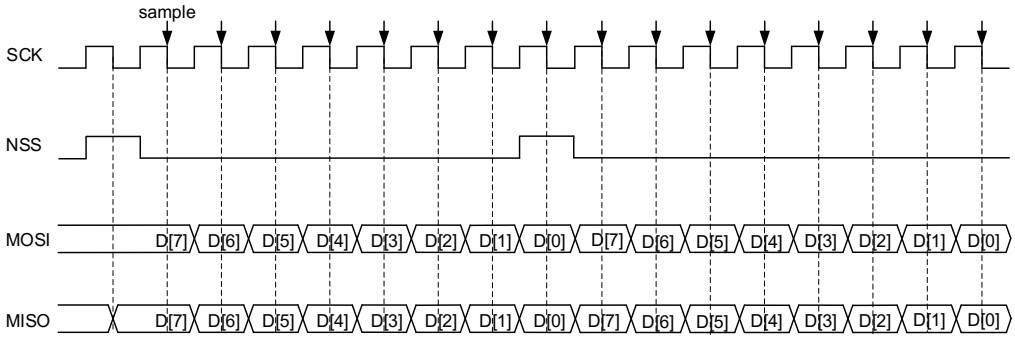
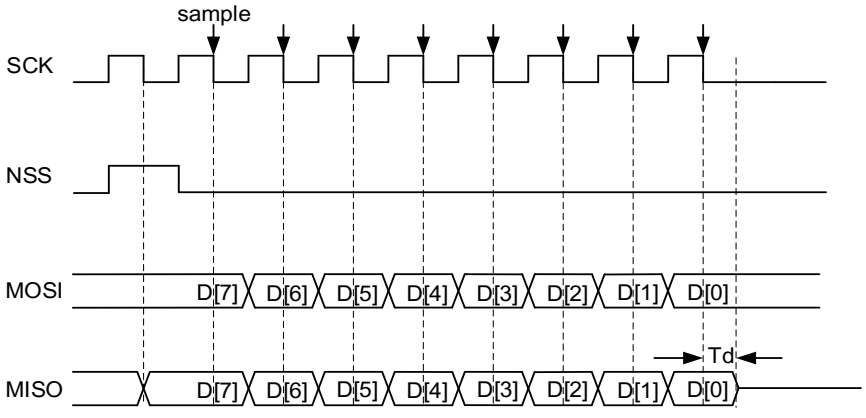


Figure 26-9. Timing diagram of TI master mode with continuous transfer



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 26-10. 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.

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (26\text{-}1)$$

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.

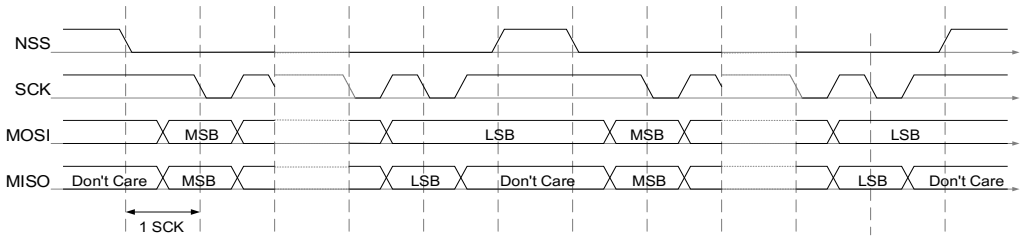
NSS pulse mode operation sequence

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 26-11. 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

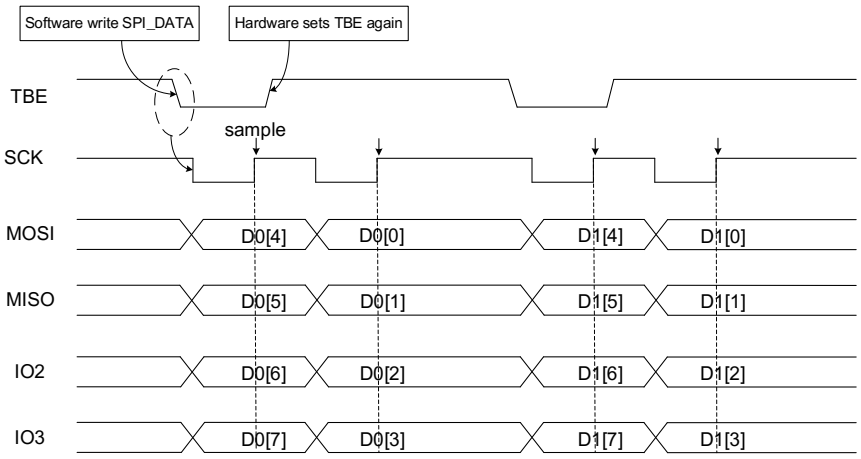
SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written

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 26-12. Timing diagram of quad write operation in Quad-SPI mode



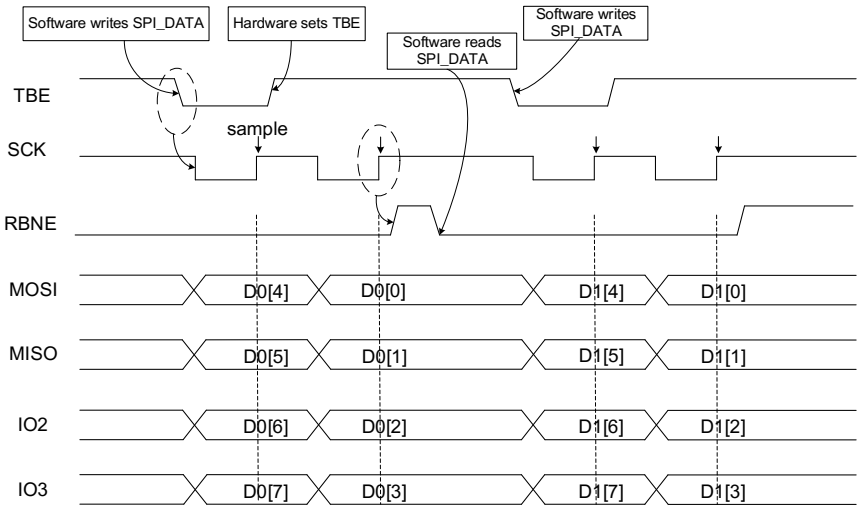
Quad read operation

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 26-13. 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.

26.3.6. 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.

26.3.7. 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).

26.3.8. 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.

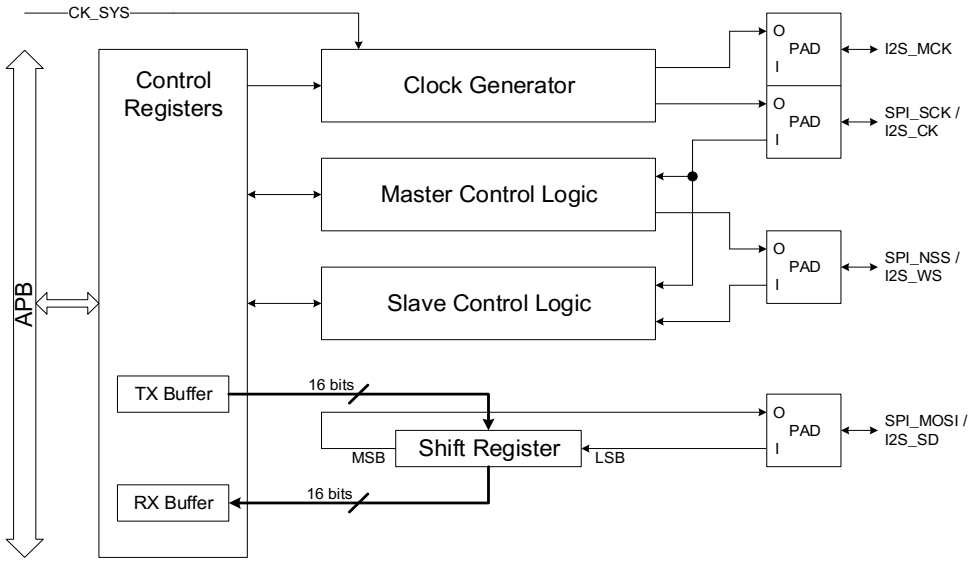
Table 26-6. SPI interrupt requests

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	

26.4. I2S function overview

26.4.1. I2S block diagram

Figure 26-14. 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).

26.4.2. 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.

26.4.3. 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 26-15. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

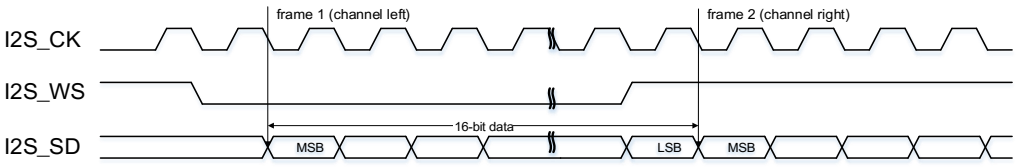
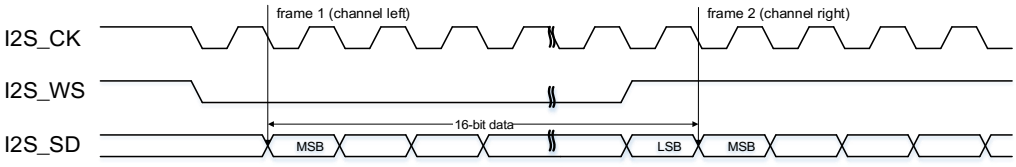


Figure 26-16. I2S Philips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)



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 26-17. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

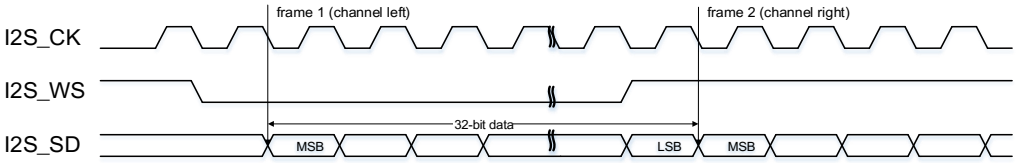
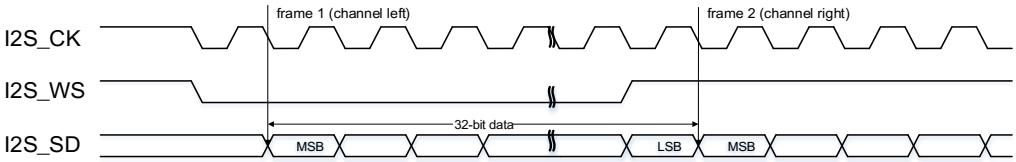


Figure 26-18. I2S Philips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)



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 26-19. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

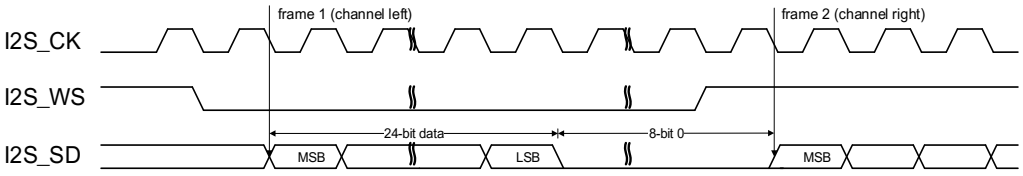
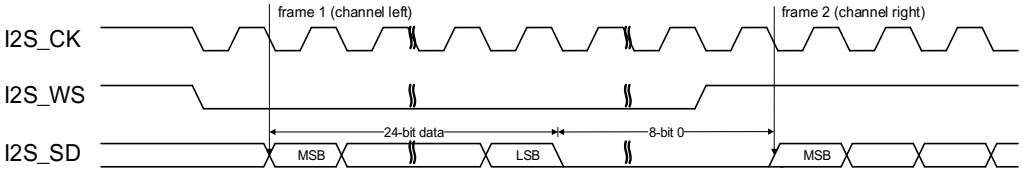


Figure 26-20. I2S Philips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



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 26-21. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

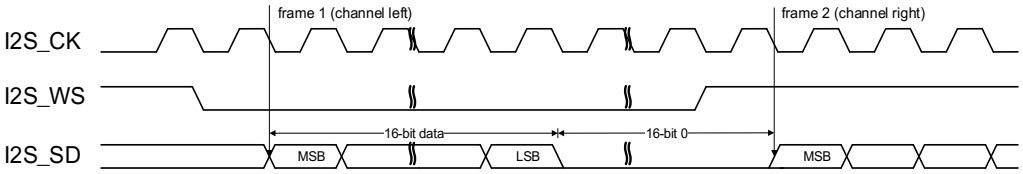
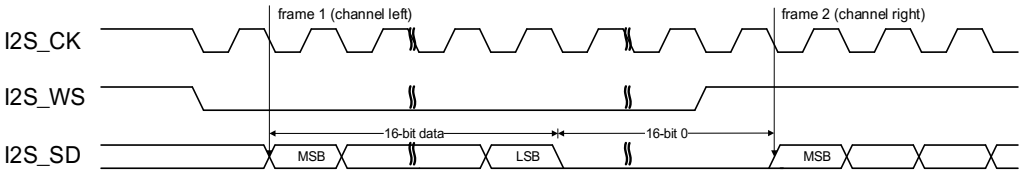


Figure 26-22. I2S Philips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 26-23. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

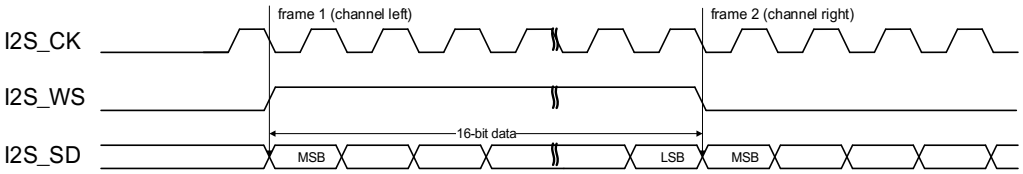


Figure 26-24. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

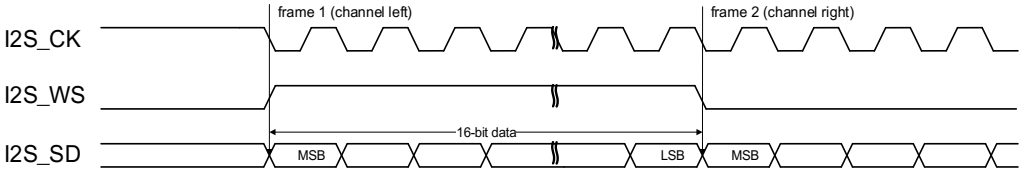


Figure 26-25. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

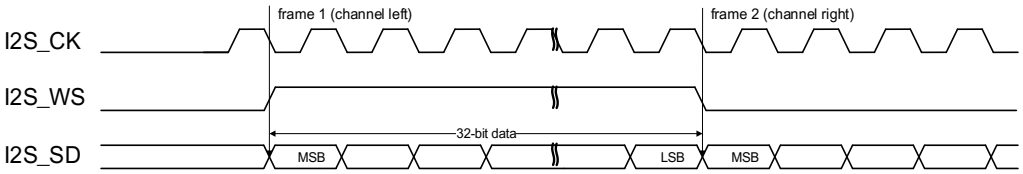


Figure 26-26. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

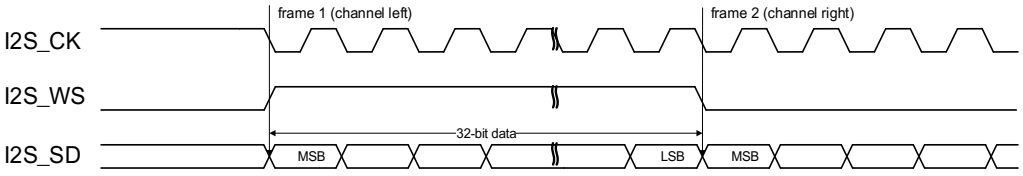


Figure 26-27. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

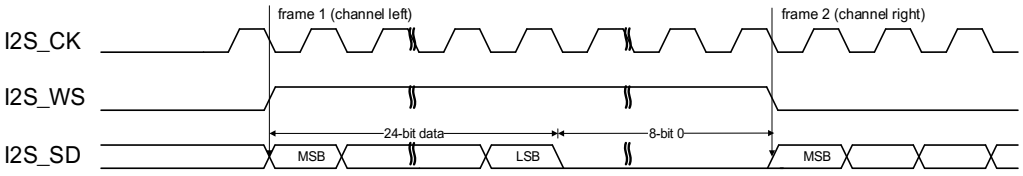


Figure 26-28. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

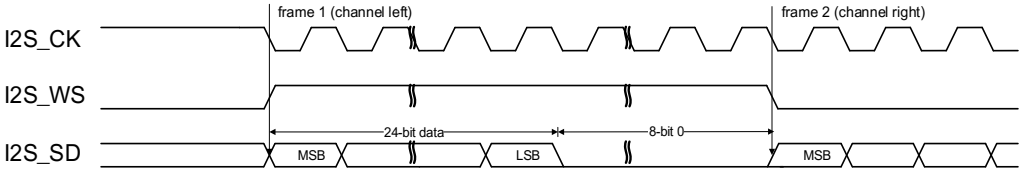


Figure 26-29. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

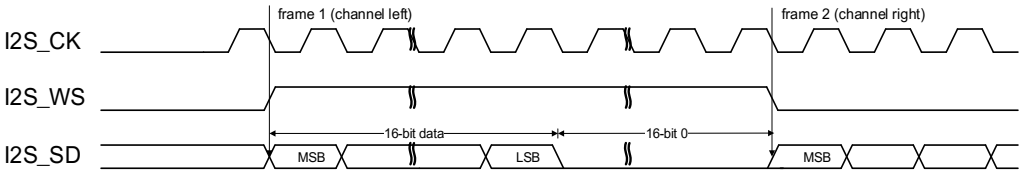
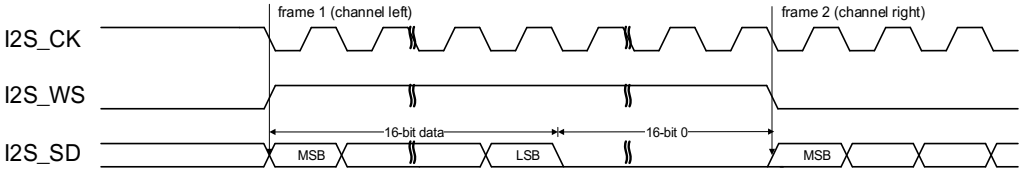


Figure 26-30. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 26-31. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

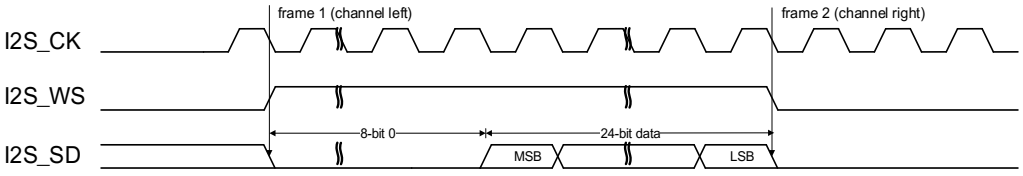
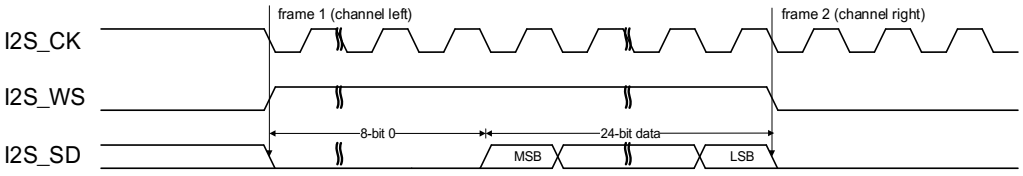


Figure 26-32. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



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 26-33. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

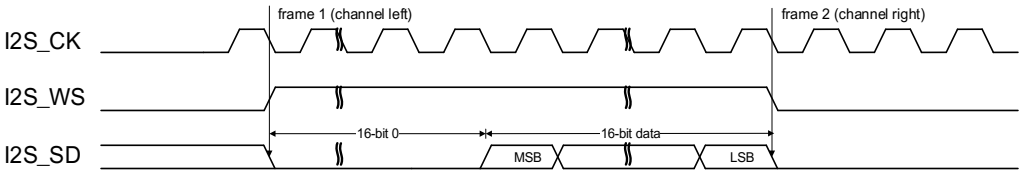
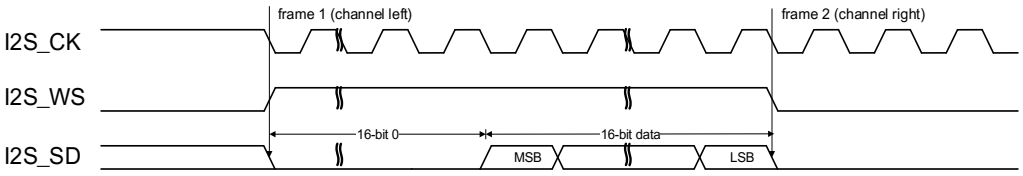


Figure 26-34. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 26-35. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

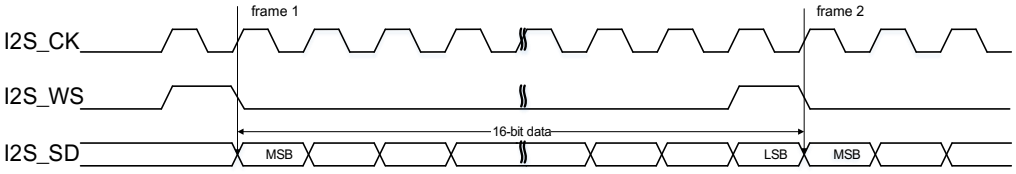


Figure 26-36. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

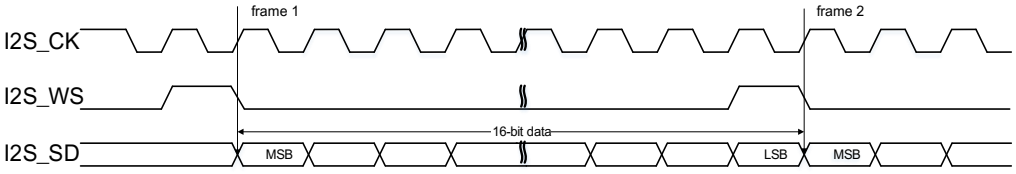


Figure 26-37. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

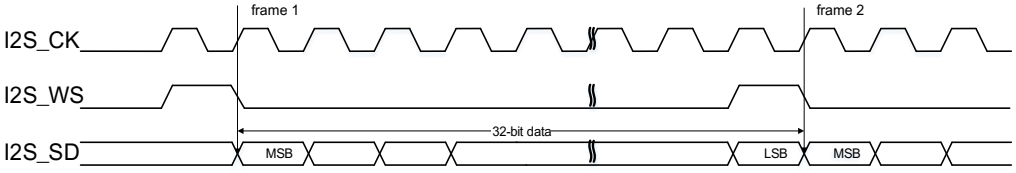


Figure 26-38. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

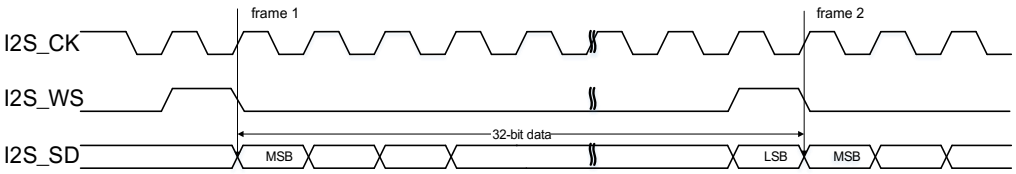


Figure 26-39. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

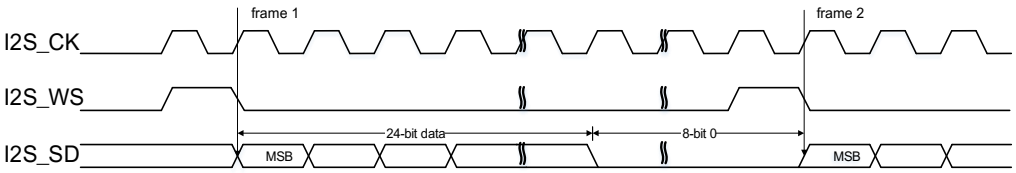


Figure26-40. PCM standard short frame synchronization mode timing diagram

(DTLEN=01, CHLEN=1, CKPL=1)

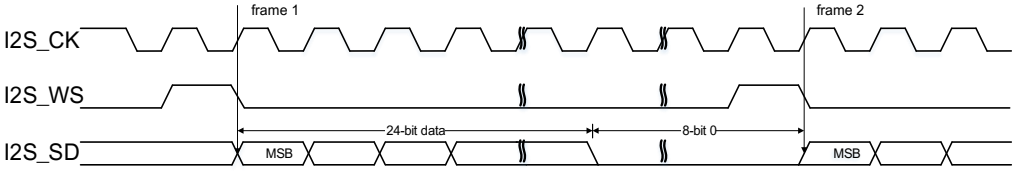


Figure 26-41. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

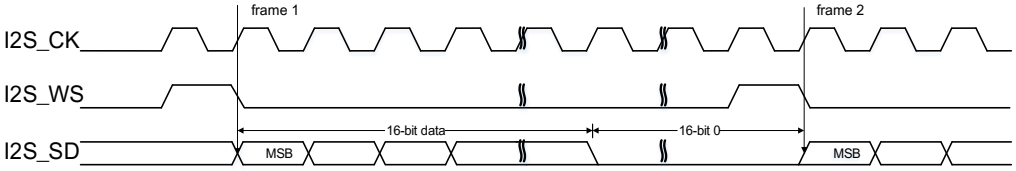
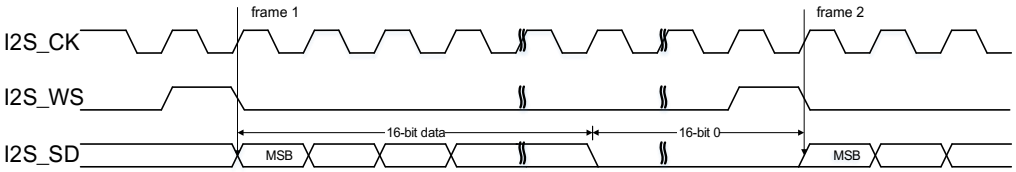


Figure 26-42. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 26-43. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

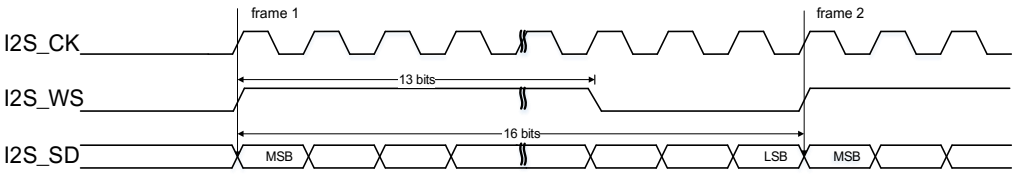


Figure26-44. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

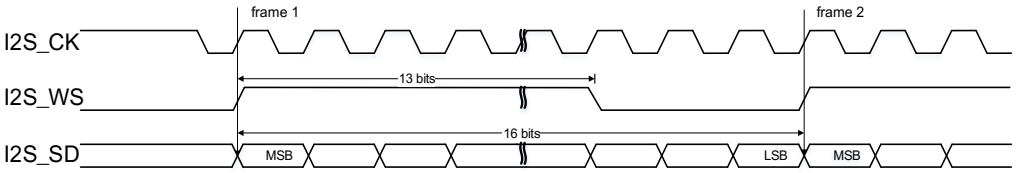


Figure 26-45. PCM standard long frame synchronization mode timing diagram

(DTLEN=10, CHLEN=1, CKPL=0)

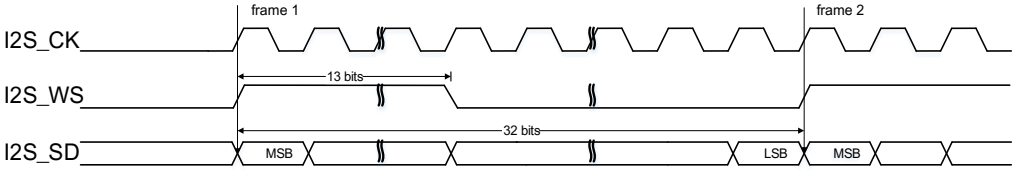


Figure 26-46. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

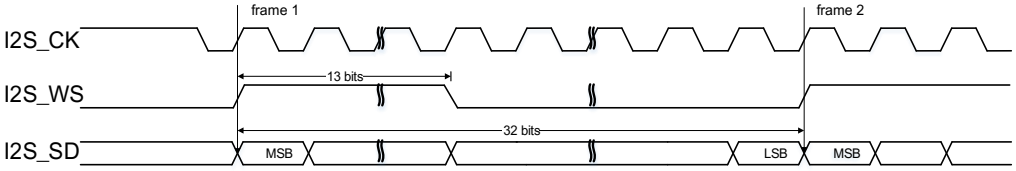


Figure 26-47. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

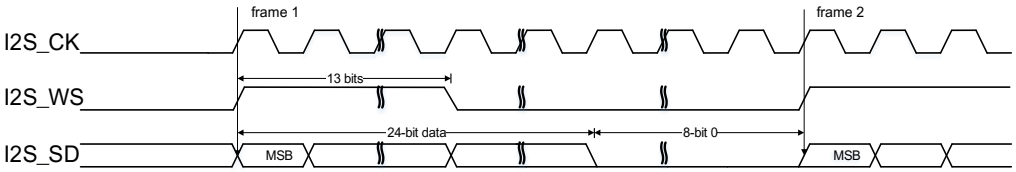


Figure 26-48. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

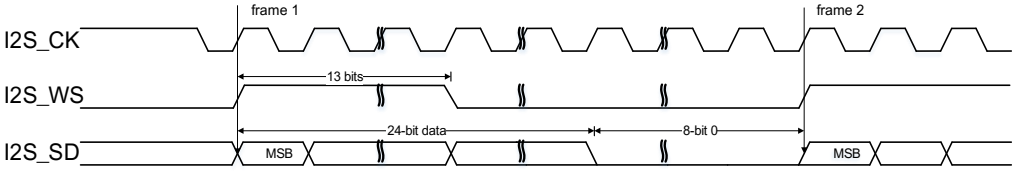


Figure 26-49. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

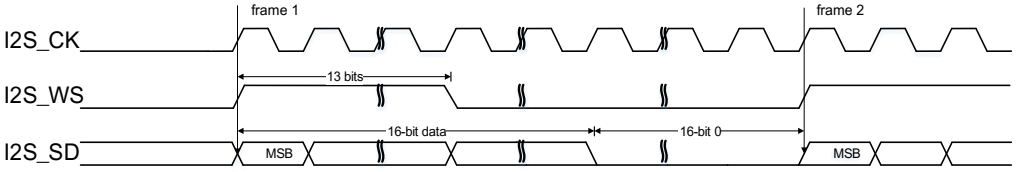
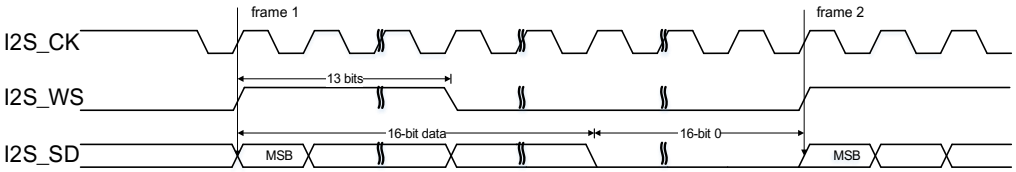
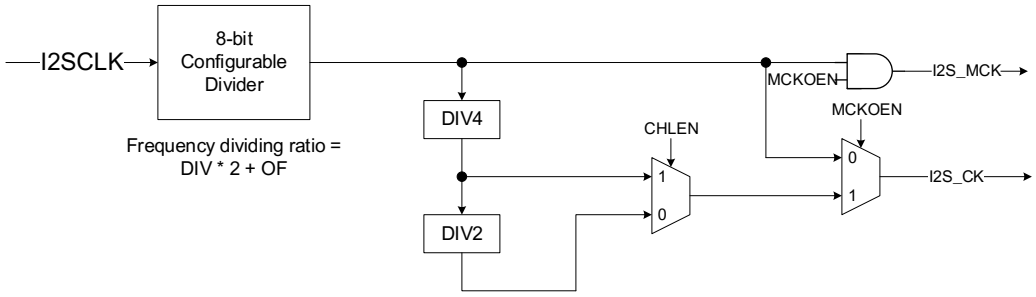


Figure 26-50. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



26.4.4. I2S clock

Figure 26-51. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as ***Figure 26-51. 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 26-7. I2S bitrate calculation formulas***.

Table 26-7. 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 26-8. Audio sampling frequency calculation formulas***.

Table 26-8. 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))

The source of I2S clock can be either from PLLI2S or an external I2S_CKIN pin, and this is programmable in RCU. Software should carefully calculate the factor of I2S and PLLI2S to

get the most accurate audio sampling frequency. If PLLI2S cannot meet the application's precision demand, an external precise I2S clock can be imported from I2S_CKIN pin.

26.4.5. 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 26-9. Direction of I2S interface signals for each operation mode*.

Table 26-9. Direction of I2S interface signals for each operation mode

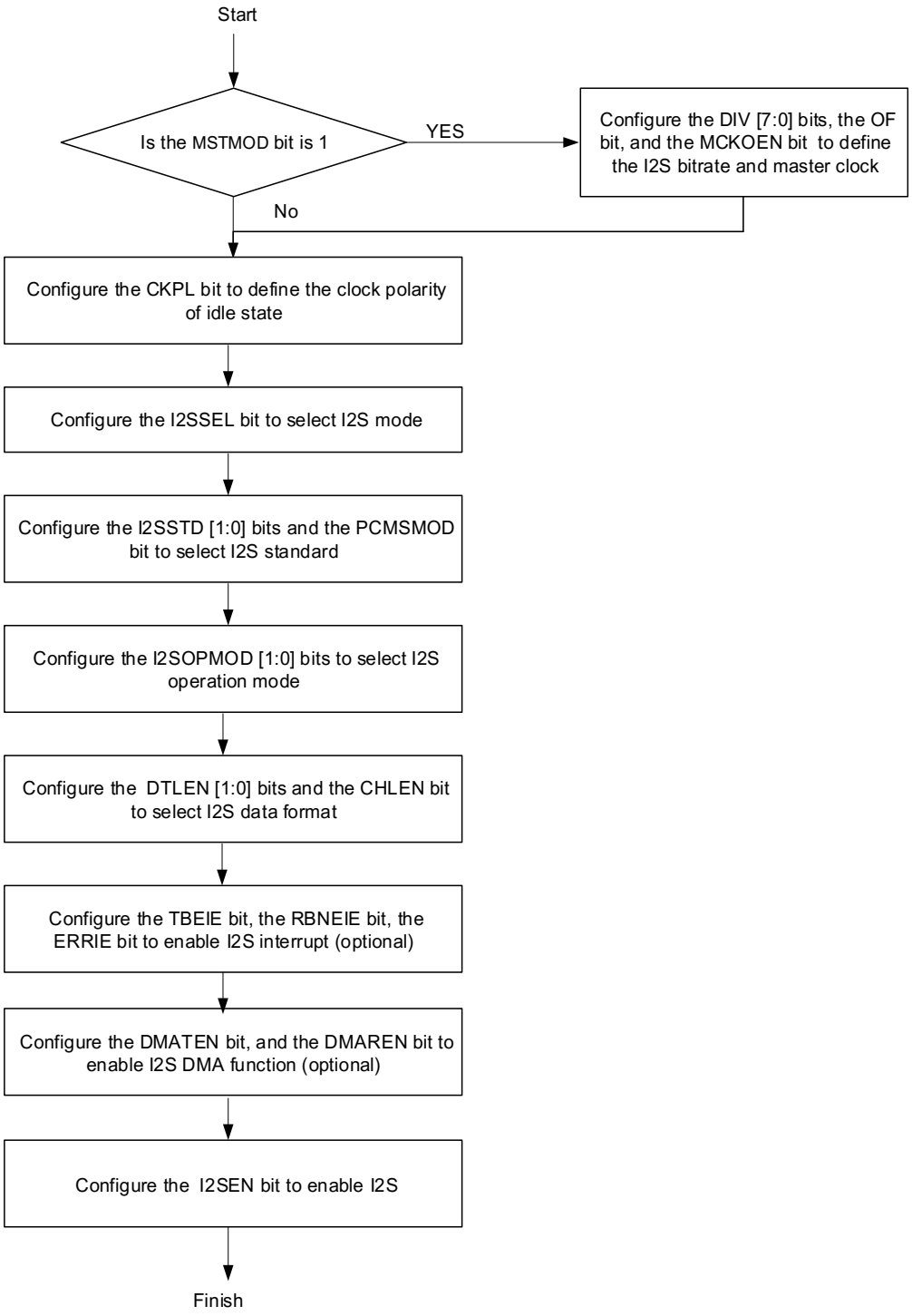
Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD	I2S_ADD_SD(2)
Master transmission	output or NU(1)	output	output	output	NU(1)
Master reception	output or NU(1)	output	output	input	NU(1)
Slave transmission	input or NU(1)	input	input	output	NU(1)
Slave reception	input or NU(1)	input	input	input	NU(1)
Full-duplex	output or NU(1)	output	output	output or input	output or input

1. NU means the pin is not used by I2S and can be used by other functions.
2. To support full-duplex operation in I2S1 and I2S2, there are two extra I2S modules in chip called I2S_ADD1 and I2S_ADD2. I2S_ADD_SD is the data pin of I2S_ADD module. Full-duplex will be described later in this chapter.

I2S initialization sequence

I2S initialization sequence is shown in below *Figure 26-52. I2S initialization sequence*.

Figure 26-52. 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.

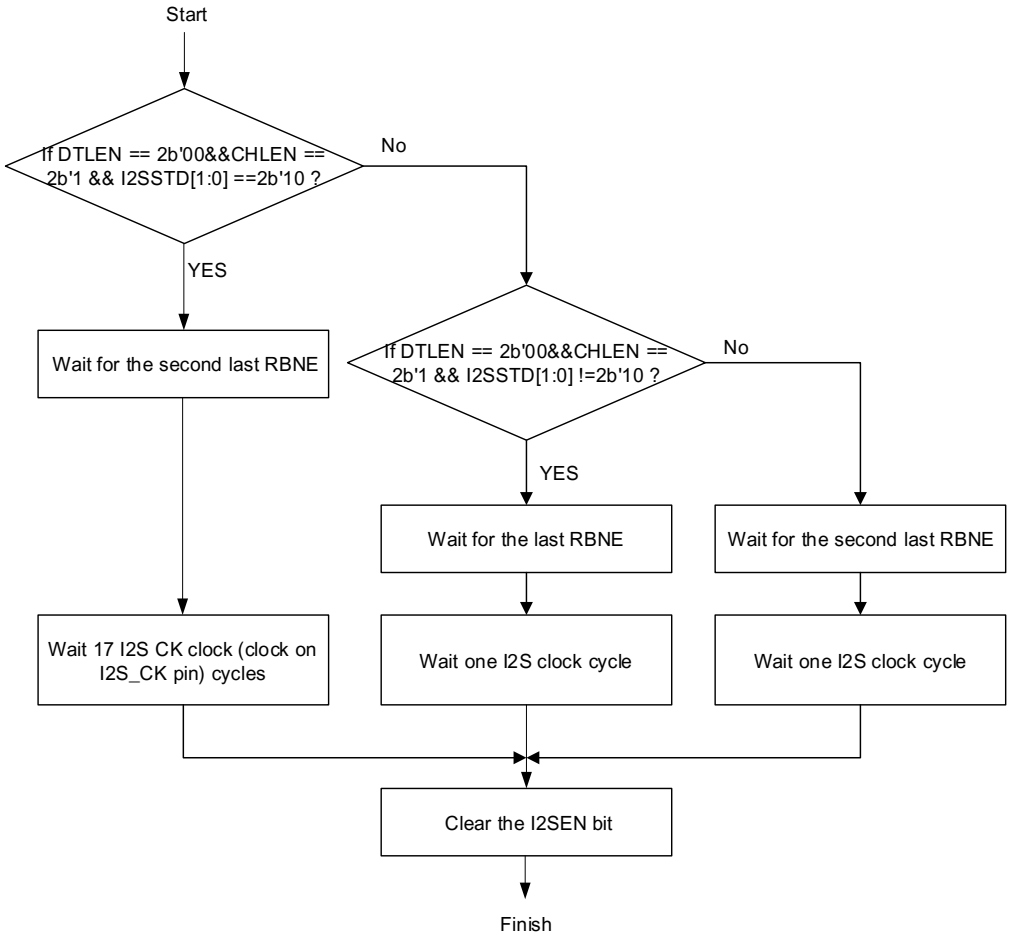
I2S master reception sequence

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 26-53. I2S master reception disabling sequence***.

Figure 26-53. 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.

I2S full-duplex mode

A single I2S only supports one-way transmission: transmit or receive mode. I2S full-duplex is supported by using an extra I2S module: I2S_ADD simultaneously with I2S. I2S_ADD module has the same function with I2S module, but can only work in slave mode. There are two I2S_ADD modules: I2S_ADD1 and I2S_ADD2, so only I2S1 and I2S2 support full-duplex mode. I2S_ADD's I2S_CK and I2S_WS are internally connected to its respective I2S's respective ports. I2S_ADD's I2S_SD pin is mapped to respective I2S's SPI_MISO pin.

In order to work in full-duplex mode, application should enable the I2S module as well as its corresponding I2S_ADD module. I2S supports two full-duplex modes: master mode and salve mode.

In master full-duplex mode, software should set I2S as a master, and I2S_ADD as a slave. Then I2S_ADD's WS and SCK signals come from the master I2S.

In slave full-duplex mode, software should set both I2S and I2S_ADD as slaves. Then, the WS and CK signals of both I2S_ADD and I2S come from external.

Application may configure I2S into either a transmitter or a receiver and thus, configure I2S_ADD into opposite data direction. During transmission, software should operate registers and handle interrupts for both I2S and I2S_ADD to make a full-duplex transmission.

26.4.6. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

26.4.7. 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 26-10. I2S interrupt***.

Table 26-10. 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	

Flag Name	Description	Clear Method	Interrupt Enable bit
		read SPI_STAT register.	
FERR	I2S Format Error	Read SPI_STAT register	

26.5. Register definition

SPI0 base address: 0x4001 3000

SPI1/I2S1 base address: 0x4000 3800

SPI2/I2S2 base address: 0x4000 3C00

SPI3 base address: 0x4001 8400

I2S1_ADD base address: 0x4000 3400

I2S2_ADD base address: 0x4000 4000

26.5.1. Control register 0 (SPI_CTL0)

Address offset: 0x00
Reset value: 0x0000 0000

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

This register has no meaning in I2S mode.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The 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 and SPI3 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

26.5.2. Control register 1 (SPI_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
								rw	rw	rw	rw	rw	rw	rw	rw

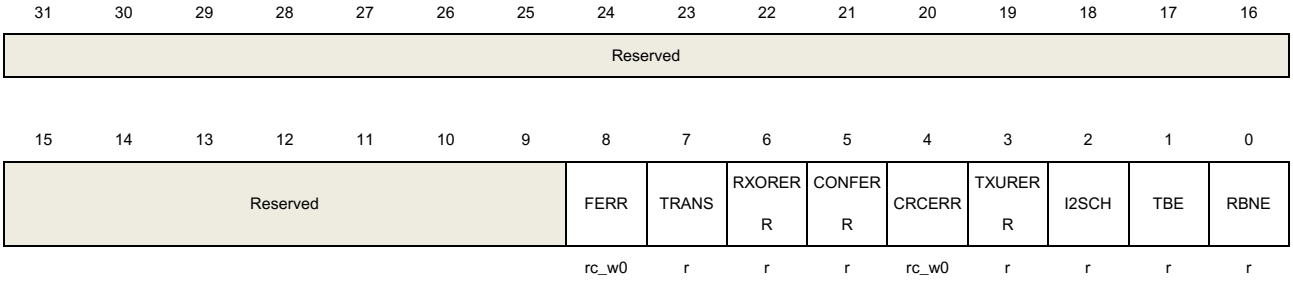
Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	TBEIE	Transmit buffer empty interrupt enable 0: TBE interrupt is disabled 1: TBE interrupt is enabled. An interrupt is generated when the TBE bit is set.
6	RBNEIE	Receive buffer not empty interrupt enable 0: RBNE interrupt is disabled 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.

26.5.3. Status register (SPI_STAT)

Address offset: 0x08
Reset value: 0x0002

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



Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	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.

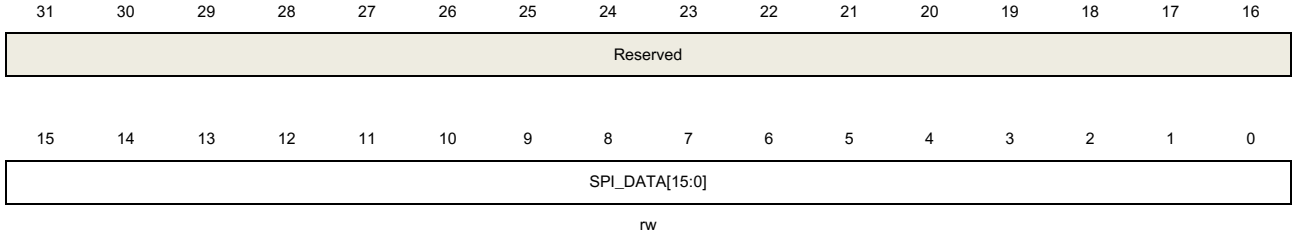
		1: SPI or I2S is currently transmitting and/or receiving a frame This bit is set and cleared by hardware.
6	RXORERR	Reception overrun error bit 0: No reception overrun error occurs. 1: Reception overrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_DATA register 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

26.5.4. Data register (SPI_DATA)

Address offset: 0x0C

Reset value: 0x0000 0000

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



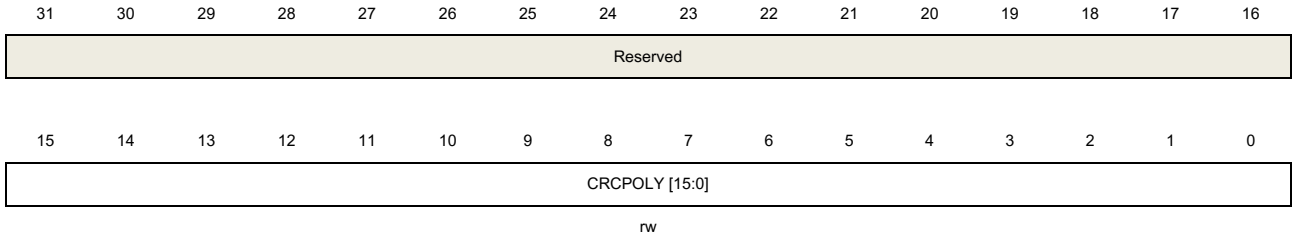
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	Data transfer register The hardware has two buffers, including transmit buffer and receive buffer. Write data to SPI_DATA will save the data to transmit buffer and read data from SPI_DATA will get the data from receive 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.

26.5.5. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

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



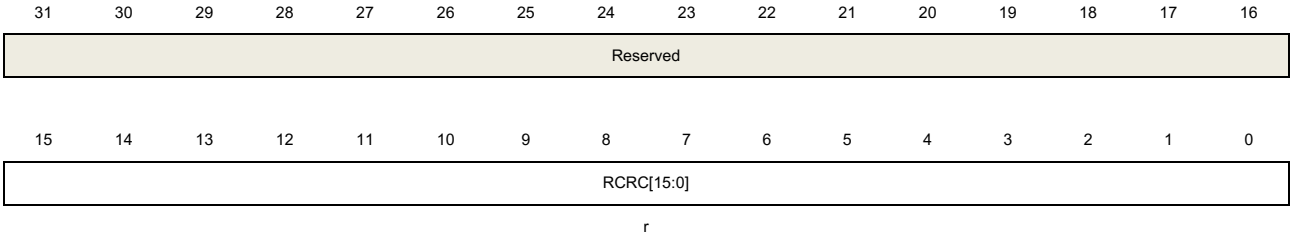
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CRCPOLY[15:0]	CRC polynomial register This register contains the CRC polynomial and it is used for CRC calculation. The default value is 0007h.

26.5.6. RX CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000 0000

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



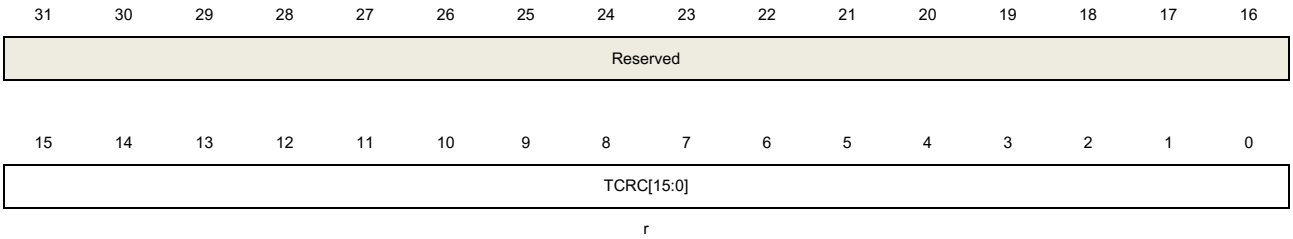
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCRC[15:0]	RX CRC register When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the received bytes and saves them in RCRC register. If the Data frame format is set to 8-bit data, CRC calculation is based 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.

26.5.7. TX CRC register (SPI_TCRC)

Address offset: 0x18

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:16	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.

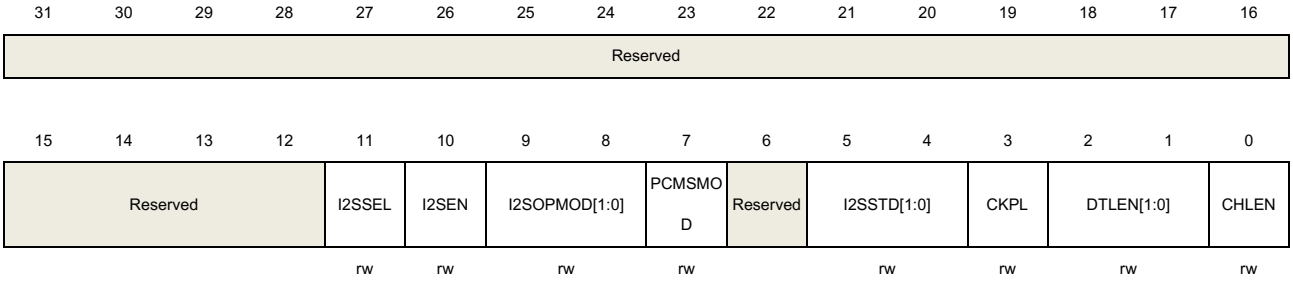
This register is reset when the CRCEN bit or the SPIEN bit in SPI_CTL0 register is cleared.

26.5.8. I2S control register (SPI_I2SCTL)

Address offset: 0x1C

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	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

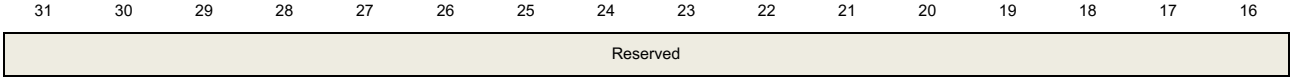
		0: Short frame synchronization 1: long frame synchronization This bit has a meaning only when PCM standard is used. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
6	Reserved	Must be kept at reset value.
5:4	I2SSTD[1:0]	I2S standard selection 00: I2S Philips 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.

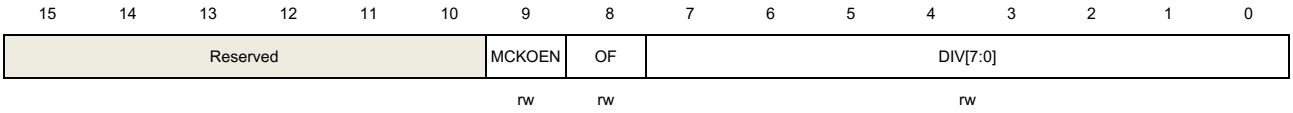
26.5.9. I2S clock prescaler register (SPI_I2SPSC)

Address offset: 0x20

Reset value: 0x0000 0002

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





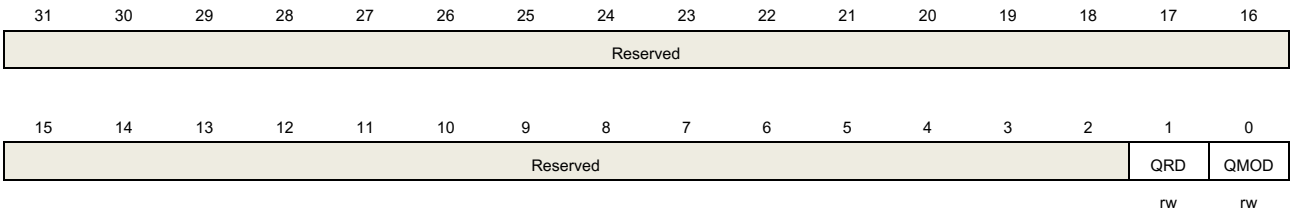
Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	MCKOEN	I2S_MCK output enable 0: I2S_MCK output is disabled 1: I2S_MCK output is enabled This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
8	OF	Odd factor for the prescaler 0: Real divider value 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.

26.5.10. Quad-SPI mode control register (SPI_QCTL)

Address offset: 0x80

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	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 and SPI3.

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 and SPI3.

27. Octal-SPI interface(OSPI)

27.1. Overview

The OSPI is a specialized interface that communicate with external memories such as serial PSRAMS, serial NAND, serial NOR Flash, HyperRAM™ and HypterFlash™ memories. The interface support single, dual, quad, dual CSN and octal SPI mode.

27.2. Characteristics

- Three functional modes:
  - indirect mode: all operations are performed depends on OSPI registers.
  - status polling mode: the values of status registers in external memory are periodically read and check.
  - memory-mapped mode: the external memory is mapped to the microcontroller address space(OSPI: 0xB000 0000 - 0xBFFF FFFF) and is accessed as an internal memory.
- Support read and write in memory-mapped mode.
- Support single, dual, quad, dual CSN and octal communication.
- Support data strobe support (DQS).
- Fully programmable command format for both indirect and memory-mapped mode.
- Support SDR(single data rate) and DTR(double transfer rate).
- Support two frame format:
  - the standard frame format.
  - HyperBus™ frame format.
- Support wrap for both standard frame format and HyperBus™ frame format.
- Integrated FIFO for transmission/reception.
- 8, 16 and 32-bits data access.
- DMA channel for indirect mode.
- Interrupt generation on FIFO threshold, status match, timeout, transfer complete and access error.

27.3. Function overview

27.3.1. OSPI block diagram

The pins of OSPI are described in *Table 27-1. OSPI signal description*.

Table 27-1. OSPI signal description

Pin name	Direction	Description
CSN0/1	O	chip select output(active low)
SCK	O	clock output
nSCK	O	inverted clock used only in the 1.8V HyperBus protocol.
DQS	I/O	data strobe signal
IO0/SO	I/O	single mode: data output. dual mode: data input or output. qual mode: data input or output. dual-qual mode: data input or output. octal mode: data input or output.
IO1/SI	I/O	single mode: data input. dual mode: data input or output. qual mode: data input or output. dual-qual mode: data input or output. octal mode: data input or output.
IO2	I/O	single mode: output mode and forced to 0, connect WP pin of external memories, control "write protect" function. dual mode: output mode and forced to 0, connect WP pin of external memories, control "write protect" function. quad mode: data input or output. dual-qual mode: data input or output. octal mode: data input or output.
IO3	I/O	single mode: output mode and forced to 1, connect HOLD pin of external memories, control "hold" function. dual mode: output mode and forced to 1, connect HOLD pin of external memories, control "hold" function. quad mode: data input or output. dual-qual mode: data input or output. octal mode: data input or output.
IO4	I/O	single mode: output mode and forced to 0. dual mode: output mode and forced to 0. quad mode: output mode and forced to 0. dual-qual mode: data input or output. octal mode: data input or output.
IO5	I/O	single mode: output mode and forced to 0. dual mode: output mode and forced to 0. quad mode: output mode and forced to 0. dual-qual mode: data input or output. octal mode: data input or output.
IO6	I/O	single mode: output mode and forced to 0.

Pin name	Direction	Description
		dual mode: output mode and forced to 0. quad mode: output mode and forced to 0. dual-qual mode: data input or output. octal mode: data input or output.
IO7	I/O	single mode: output mode and forced to 0. dual mode: output mode and forced to 0. quad mode: output mode and forced to 0. dual-qual mode: data input or output. octal mode: data input or output.

Figure 27-1 OSPI octal communication mode block diagram

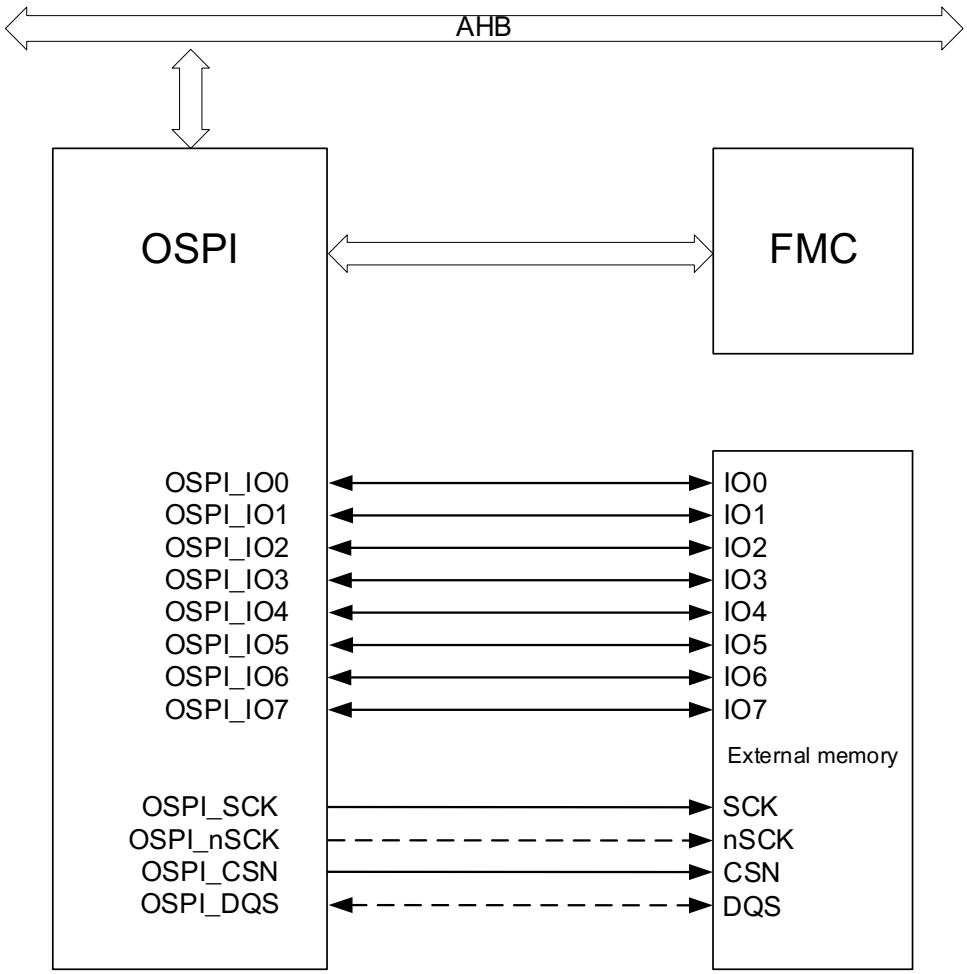


Figure 27-2 OSPI quad communication mode block diagram

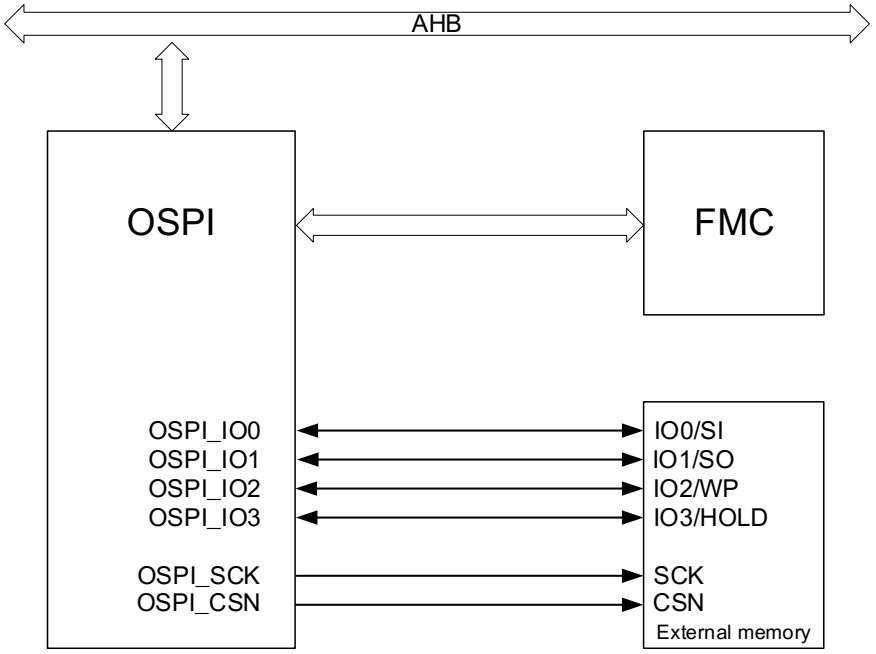
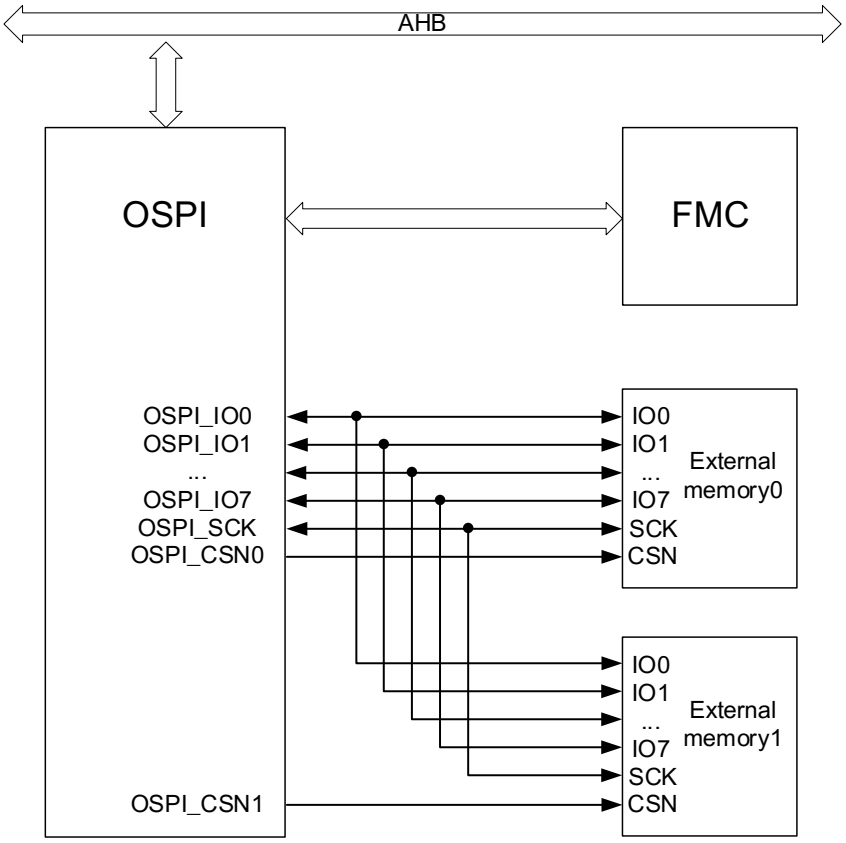


Figure 27-3 OSPI dual CSN communication mode block diagram



27.3.2. OSPI regular command format

In regular command, there are totally 5 phases which can be included or not: instruction, address, alternate-bytes, dummy and data. Any of these phases can be omitted or not, but at least one of the instructions phase, address phase, alternate byte phase and data phase must be present, this must be guaranteed by software, hardware is not designed to provide any protection methods. In addition, the most-signification-bit always occupies the highest IO line number.

Figure 27-4 OSPI command format in octal mode

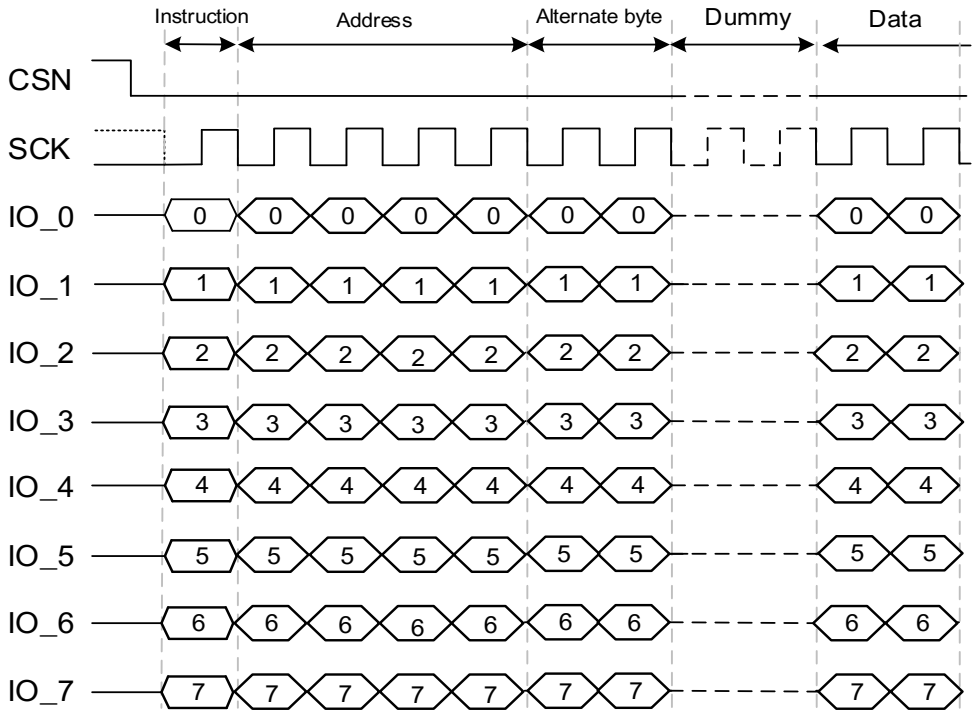
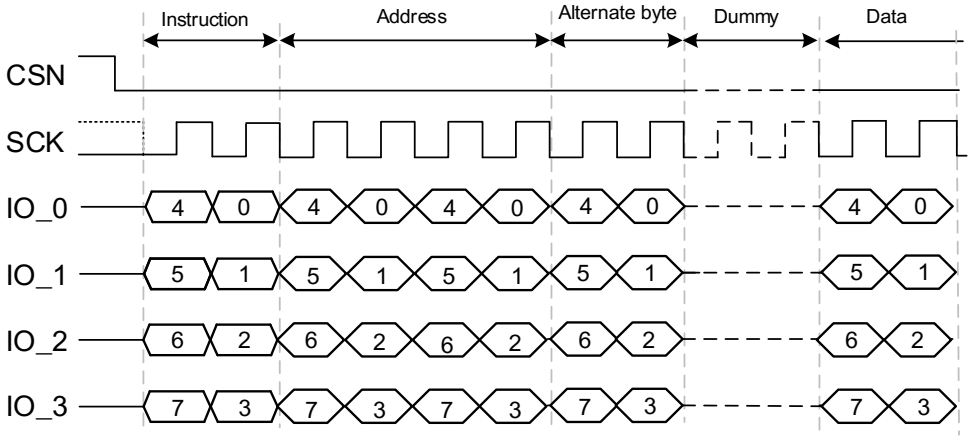


Figure 27-5 OSPI command format in quad mode



Instruction phase

In instruction phase, configure the command size to be sent in the INSSZ[1:0] bit field of the OSPI_TCFG register(8, 16, 24 or 32 bits), configure the instructions to be sent to the external memory in the OSPI_INS register. The IMOD[2:0] field of the OSPI_TCFG register defines the instruction phase mode (no instruction, 1-line, 2-lines, 4-lines or 8-lines). Set the INSDTR bit of the OSPI_TCFG register to enable the DTR mode, and the command is sent on each edge of the clock.

When FMOD[1:0] bit field of the OSPI_CTL register is 0b11, the OSPI works in memory-mapped mode. In memory-mapped mode, define the write operation instruction in the OSPI_WINS register, configure the format of the instruction in the OSPI_WTCFG register (size, mode and whether to enable DTR). Configure the read operation instruction and the format of the instruction in the OSPI_INS register and OSPI_TCFG register.

Address phase

In address phase, configure the address size to be sent in the ADDRSZ[1:0] bit field of the OSPI_TCFG register(8, 16, 24 or 32 bits). The ADDRMOD[2:0] field of the OSPI_TCFG register defines the address phase mode (no address, 1-line, 2-lines, 4-lines or 8-lines). Set the ADDRDTR bit of the OSPI_TCFG register to enable the DTR mode, and the address is sent on each edge of the clock. In indirect mode and status polling mode, define the address information to be sent in the OSPI_ADDR register.

In memory-mapped mode, the address to be sent is directly given by AHB. Configure the format of the write operation address in the OSPI_WTCFG register (size, mode and whether to enable DTR). Configure the format of read operation address in the OSPI_TCFG register.

Alternate-bytes phase

In alternate-bytes phase, configure the alternate bytes size to be sent in ALTESZ[1:0] bit field of the OSPI_TCFG register(8, 16, 24 or 32 bits), configure the alternate bytes to be sent to the external memory in the OSPI_ALTE register. The ALTEMOD[2:0] field of the OSPI_TCFG register defines the alternate-bytes phase mode (no alternate-bytes, 1-line, 2-lines, 4-lines or 8-lines). Set the ABDTR bit of the OSPI_TCFG register to enable the DTR mode, and the alternate bytes is sent on each edge of the clock.

In memory-mapped mode, define the write operation alternate bytes in the OSPI_WALTE register, configure the format of the alternate bytes in the OSPI_WTCFG register (size, mode and whether to enable DTR). Configure the read operation alternate bytes and the format of the alternate bytes in the OSPI_ALTE register and OSPI_TCFG register.

Dummy phase

In dummy phase, 0-31 cycles, as specified by DUMYC[4:0] bit field in OSPI_TIMCFG register, are given without any data being transferred for external memory, in order to wait flash prepare

data.

In memory-mapped mode, specify write operation dummy cycles in DUMYC[4:0] bit field of the OSPI_WTIMCFG register, and specify read operation dummy cycles in DUMYC[4:0] bit field of the OSPI_TIMCFG register.

Note:

1. At least 1 dummy cycle when OSPI is working in 2-lines, 4-lines or 8-lines mode to receive data from external memory.
2. At least 5 dummy cycles when using memory with DQS activated.

Data phase

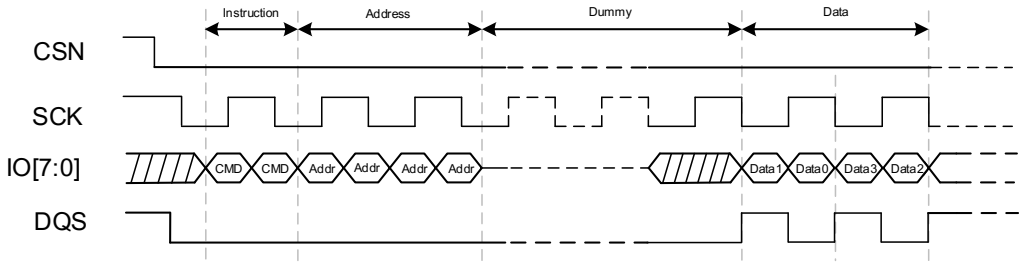
In data phase, any number of bytes can be transferred between the external memory and the OSPI interface. The DATAMOD[2:0] bit field of the OSPI_TCFG register defines the mode of the data phase(no data, 1-line, 2-lines, 4-lines or 8-lines, of which no data can only be used in indirect write mode). Set the DADTR bit of the OSPI_TCFG register to enable the DTR mode, and the data is sent on each edge of the clock. In indirect mode, the OSPI_DTLEN register defines the number of bytes to be sent or received. In write operation, data to be sent should be written to the OSPI_DATA register, while in read operation, received data is obtained by reading OSPI_DATA register. In status polling mode, OSPI_DTLEN register defines the number of bytes to be received, and the receive data is obtained from the OSPI_DATA register.

In memory-mapped mode, data to be sent is directly given by AHB. The number of bytes transmitted determines the access operation of the AHB bus, which can be read and written in 8-bit, 16-bit or 32-bit, and correspondingly transfer 1, 2, or 4 bytes. Configure the format of the data in the OSPI_WTCFG register(mode and whether to enable DTR) for sending. Configure the format of the data in the OSPI_TCFG register(mode and whether to enable DTR) for reading.

DQS signal

The DQS signal is typically used in high speed applications to indicate when the output data of memory is ready to be fetched by the MCU. When the DQS function is enabled, the toggling frequency is the same as the SCK frequency. For SDR Read operations, the data should only be latched on the rising edge of DQS signal. For DTR Read operations, the data should be latched on both rising edge and falling edge of the DQS signal.

Figure 27-6. DTR read timing using DQS in octal mode



Dual CSN mode

When dual CSN mode is enabled (DCSNEN = 1), refer to ***Figure 27-3 OSPI dual CSN communication mode block diagram***, the memory map mode requires the map size (MESZ) to be configured with a maximum value of 256M. The memory space is then divided into two parts, each occupying 128M. Specifically:

- CSN0 corresponds to the address range B000_0000 ~ B7FF_FFFF
- CSN1 corresponds to the address range B800_0000 ~ BFFF_FFFF

When the bus accesses the address range B000_0000 ~ B7FF_FFFF, CSN0 is pulled low; otherwise, CSN1 is pulled low. This design allows two memory devices to be connected simultaneously, sharing the same set of IOs, with the chip select signal (CSN) used to distinguish between them.

In indirect mode, enabling CSN1EN allows the selection of a fixed CSN output. Through register configuration, the settings for the two memory devices can be configured independently, enabling flexible and efficient operation.

Note:

1. When CSN1 is configured to operate in memory-mapped mode, to ensure proper access, the ADDRMOD bit field in the TCFG0 register associated with CSN0 must be configured to a non-zero value.

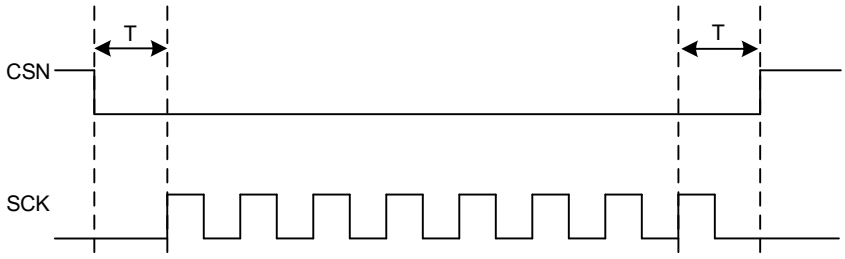
2. PE7/PE8/PE9/PE10/PE12/PE13/PE14/PE15 and PD9/PD10 cannot be used as pins corresponding to the memory-mapped mode of CSN1.

CSN and SCK behavior

The default value of CSN is high, and it falls before a command begins and rises soon as it finishes. SCK output signal is a gate signal from internal SCK, where the internal SCK is present all the time.

The SCK must stay low when CSN is high. CSN falls one SCK cycle before the first valid rising SCK edge, and rises on SCK cycle after the final valid rising SCK edge. Refer to ***Figure 27-7. CSN and SCK behavior***.

Figure 27-7. CSN and SCK behavior



27.3.3. HyperBusTM mode

OSPI can work in HyperBus™ mode by configuring the DTYSEL[2:0] bit field in the OSPI_DCFG0 register. In this mode, OSPI can communicate with external flash memory through the HyperBus™ protocol. The HyperBus™ frame consists of two phases: command/address phase and data phase. Pins definitions used in HyperBus™ mode refer to *Table 27-2. Signal line description in HyperBusTM mode*.

Table 27-2. Signal line description in HyperBus™ mode

Pin name	Direction	Description
CSN	O	chip select output(active low)
SCK, nSCK	O	differential clock, the nSCK is used in 1.8V HyperBus™ mode
RWDS	I	read and write data strobe and latency insertion
IO[7:0]	I/O	data input/output

Instruction/address phase

In instruction/address phase, OSPI sends 48 bits data to specify a series of operations such as address space, burst type, read or write, refer to *Table 27-3 Instruction/address phase description in HyperBusTM mode*.

Table 27-3 Instruction/address phase description in HyperBus™ mode

CA bit	Bit name	Description
47	R/W#	Indicates the access as a read or write operation, R/W# = 1 indicates read operation, R/W# = 0 indicates write operation.
46	Address space(AS)	Indicates whether the read or write transaction accesses the memory or register space. AS=0 indicates memory space, AS=1 indicates the register space.
45	Burst type	Indicates where the burst while be linear or wrapped. Burst type=0 indicates wrapped burst, Burst type=1 indicates linear burst.
44:16	Row and upper	Select the row and the upper column address

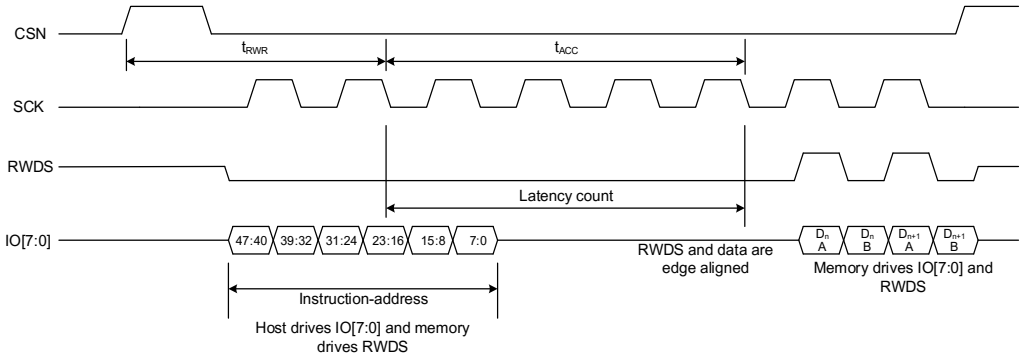
CA bit	Bit name	Description
	column address	
15:3	Reserved	-
2:0	Lower column address	Indicates lower column component of the target address: the starting word within a half-page.

The type of target space is configured by DTYSEL[2:0] bit field in the OSPI_DCFG0 register, and the total size of the device is configured in the MESZ[4:0] bit field in the OSPI_DCFG0 register.

Read operation

When OSPI read data from HyperBus™ device, OSPI send 48 bits CA to device firstly, and then start reading data after several cycles latency. When the data starts to be transmitted, the host reads the data and samples it on the edge of the RWDS signal. Refer to ***Figure 27-8. HyperBusTM read operation with initial latency***.

Figure 27-8. HyperBus™ read operation with initial latency



The t_{RWR} is the minimal recovery time for device, which is defined by the RWRTM[7:0] bit field of the OSPI_HBLCFG register. The t_{ACC} is the access time for the device, which is defined by ACCTM[7:0] bit field of the OSPI_HBLCFG register.

If external HyperBus™ device requires an additional latency, OSPI will add an additional t_{ACC} time, refer to ***Figure 27-9. HyperBusTM read operation with additional latency***. For accesses that cross page, an additional latency needs to be added when the page is crossed, refer to ***Figure 27-10. Read operation page crossing with latency***.

Figure 27-9. HyperBus™ read operation with additional latency

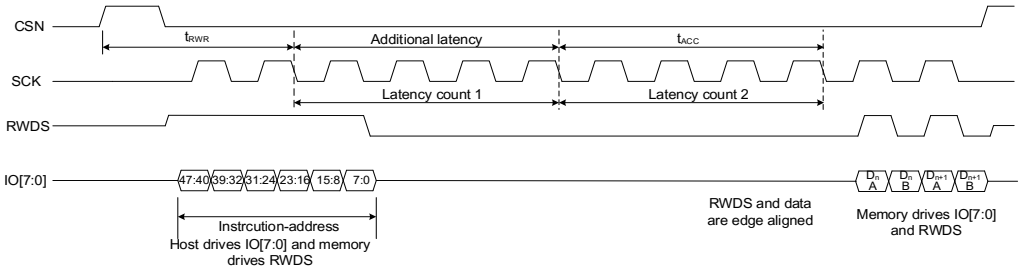
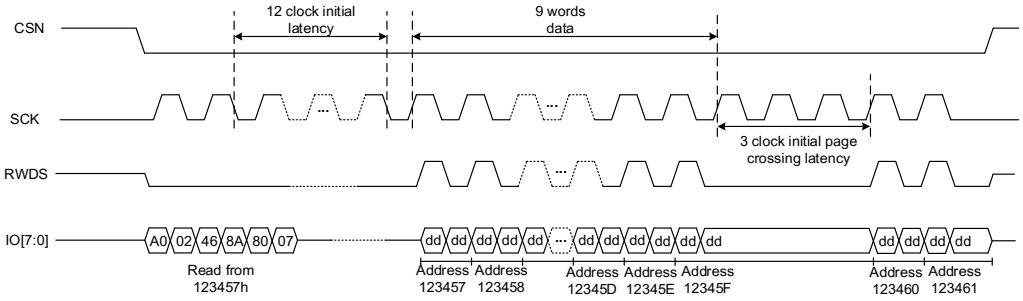


Figure 27-10. Read operation page crossing with latency



Write operation

When the host write data to HyperBus™ device, RWDS request an additional latency during the instruction/address phase and for write masking during the data phase, write operation with initial latency refer to ***Figure 27-11. HyperBusTM write operation with initial latency***.

If external HyperBus™ device requires an additional latency, OSPI will add an additional t_{ACC}, refer to ***Figure 27-12. HyperBusTM write operation with additional latency***. For some devices that require zero write latency, the latency can be set to zero by configuring the WZLAT bit in the OSPI_HBLCFG register, refer to ***Figure 27-13. HyperBusTM write operation with no latency***.

For some devices that require fixed latency, the latency can be set to $2xt_{ACC}$ by configuring the LMOD bit in the OSPI_HBLCFG register. In fixed latency mode, the RWDS is not taken into account for OSPI.

Note: When there is additional delay in Hyper, to avoid interference caused by routing delay during data reception, the SSAMPLE in OSPI_TIMCFG needs to be configured.

Figure 27-11. HyperBus™ write operation with initial latency

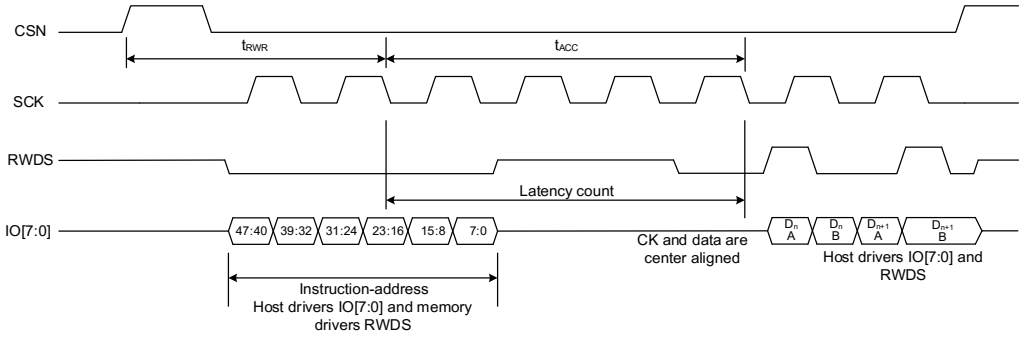


Figure 27-12. HyperBus™ write operation with additional latency

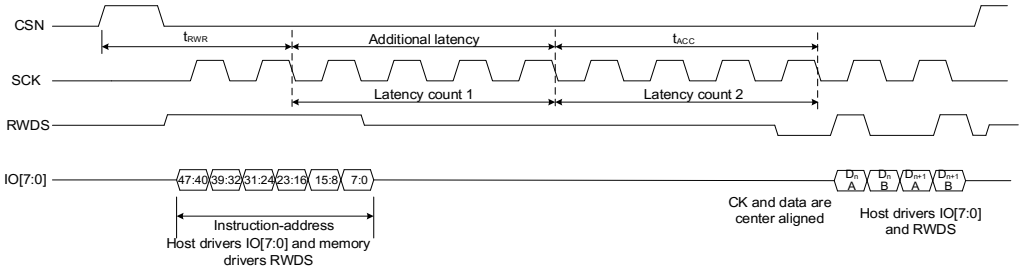
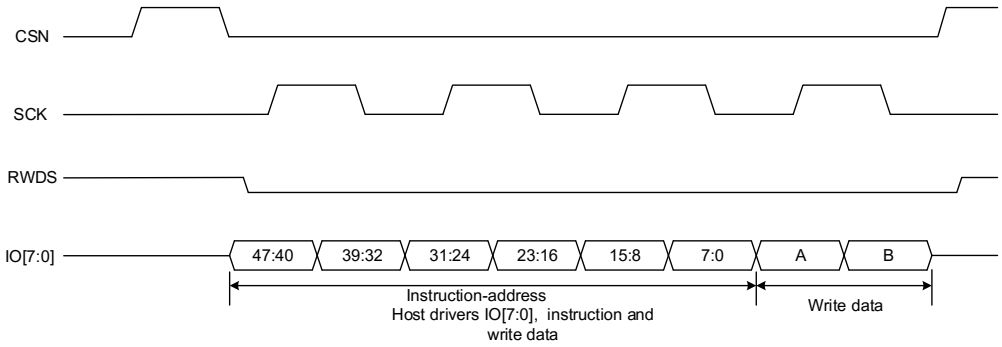


Figure 27-13. HyperBus™ write operation with no latency



27.3.4. Wrap support

Wrap is supported only in read mode when cache is enabled and when data size is 4 bytes for AHB. To enable cache, the OSPI_WPTCFGx, OSPI_WPTIMCFGx, OSPI_WPINSx and OSPI_WPALTEx must be programmed according to the wrapped read access characteristics.

27.3.5. Chip select boundary

The Chip select (CS) boundary function limits transactions to addresses aligned within a specific boundary range. This boundary size is defined by the CS boundary field CSBND[4:0] in the OSPI_DCFG2 register, where the field value represents 2^{CSBND}. When the least significant address reaches 0xF, the nCS signal will deactivate and transition to the next data

address for subsequent transactions. If CSBND[4:0] = 0, the function is disabled, and it is advised to set the minimum value to 3.

Note: Chip select boundary function does not apply to Flash memory's write mode. Write operations demand additional commands to complete programming for the next page.

27.3.6. Refresh support

Standard frame format and HyperBus™ frame format both support the refresh feature, which can limit the duration of transactions to the value defined in the REFRATE[31:0] field of the OSPI_DCFG3 register. The duration is expressed in the number of cycles, allowing external RAM devices to perform internal refresh operations regularly. The refresh value must be greater than the minimal transaction size in terms of number of cycles including the instruction/address/alternate/dummy phases.

27.4. Operating modes

27.4.1. Indirect mode

In indirect write mode, data to be transmitted are written in OSPI_DATA register. While in indirect read mode, data to be received are read from OSPI_DATA register.

OSPI_DTLEN register defines the number of byte to be transferred. If DTLEN = 0xFFFF FFFF, the number of data is considered undefined, the transmission continues until the memory size boundary is reach as specified by MESZ[4:0]. If both DTLEN = 0xFFFF FFFF and MESZ[4:0] = 0x1F, then the transmission continues indefinitely until the OSPI is disabled.

Transfer complete flag TC is set when the number of byte programed in DTLEN is reached, in case of undefined transfer length, TC is set when the transmit/received byte number equals to external memory size. An interrupt is generated if TCIE and TC are both set, and it is cleared by setting TCC to 1.

Trigger a command sequence in regular command mode

The command sequence starts immediately after the last information is provided by software according to communication requirement. After the command starts, the BUSY bit is set to 1.

When neither address nor data are required, the sequence start immediately after OSPI_INS has be accessed.

When address is required and no data is required, the sequence starts after OSPI_ADDR has been accessed.

When both address and data are required in indirect write mode, the command sequence starts after OSPI_DATA has been accessed.

Trigger a command sequence in HyperBus™ mode

The command sequence starts immediately after the last information is provided by software according to communication requirement. After the command starts, the BUSY bit is set to 1.

In indirect read mode, the command sequence starts immediately after OSPI_ADDR has been accessed.

In indirect write mode, the command sequence starts immediately after OSPI_DATA has been accessed.

FIFO and flag control

A 32-byte FIFO is used for data transmission. In indirect write mode, 32-bit write access add 4-bytes to FIFO, 16-bit write access add 2-bytes to FIFO, and 8-bit add 1-byte.

FIFO threshold is defined by FTL, in indirect read mode, when the amount of bytes in the FIFO is equal or above the defined threshold, FIFO threshold flag FT is set. FT is also set after data phase is complete if FIFO is not empty. In indirect write mode, when the amount of the empty bytes in the FIFO is above the threshold, FT is set.

An interrupt is generated if both FTIE and FT is set. If DMA is enabled, a DMA request is generated by FT, until this flag is cleared.

In indirect read mode, when the FIFO becomes full, the OSPI temporarily stop SCK clock to avoid overrun. The reading sequence is not resumed until more than 4 bytes are available in FIFO.

When the threshold exceeds 20 bytes, it is necessary to read at least 16 bytes; otherwise, the OSPI will not continue fetching data from the external device, and the FT flag will not be set.

27.4.2. Status polling mode

In status polling mode, the OSPI periodically starts a read command with up to 4 bytes of data per frame. If the data length defined in the OSPI_DTLEN register is greater than 4, ignore the extra length and only read 4 bytes. The received data can be bit-wise masked and compared with a defined data content, if a match happens, then an interrupt is generated when SMIE is set.

Status polling access starts the same as indirect read sequence. BUSY stays high even between periodic intervals.

Polling match mode SPMOD controls the comparison match mode, if SPMOD = 0, the AND mode is selected. In this mode, status match flag SM is set only when there is a match on all the unmasked bits. While if SPMOD = 1, the OR mode is selected. In this mode, SM is set if there is a match on any of the unmasked bit.

If status-polling-mode-stop SPS is set, status polling sequence stop when a match is detected,

and the BUSY flag is cleared at the end of data phase.

In status polling mode, FIFO is bypassed, the read status bytes are stored in OSPI_DATA register, and the stored status bytes are not affected by the MASK control field. OSPI_DATA content is renewed at the beginning of data phase.

FT is set at the end of data phase, where the external flash memory status bytes are considered read, and it is cleared when OSPI_DATA register is read.

In status polling mode, the external memory must be configured in fixed latency mode for HyperBus.

27.4.3. Memory-mapped mode

In memory-mapped mode, the external flash memory is considered as internal memory, no more than 256MB can be addressed even if the external memory is larger. The Memory map mode also doesn't allow an address outside what defined by MESZ[4:0] but still within 256MB range.

If any of above condition happens, AHB will generate an error .The effect of the error depends on the AHB master. If the master is CPU, a hardfault is generate, if it is DMA, transfer error is generated, and the corresponding DMA channel is disabled.

In this mode, read operations support byte, half-word, word, or burst access, while write operations support word, double-word, or burst access.

Execute in place (XIP) is also supported, where OSPI continue to load bytes to the address after completing the most recent access. If the subsequent access is indeed made at a continuous address, the access with be completed faster since the value is already prefetched. Otherwise, the read sequence is restarted, polling CSN low before the read sequence starts.

After the FIFO is empty, the OSPI enters hold state, in which no SCK is sent, CSN is maintained low during this period. If timeout counter is enabled, CSN will be pulled high when hold state contains number of SCK clock cycles equals to TMOUT control field.

At the beginning of a transfer, BUSY goes high before CSN falls, and is cleared when a timeout occurred or abort/disable is issued.

27.5. OSPI configuration

27.5.1. OSPI system configuration

The details of the OSPI system configuration as follows:

1. Configure the functional mode of OSPI with FMOD[1:0] bit field.
2. If OSPI is in status polling mode, then need to configure the SPMOD and SPS bits to select the polling matching mode and the stop mode of the automatic polling mode.

3. Configure the threshold of FIFO with FTL[4:0] bit field.
4. If need to use DMA, set the DMAEN bit to 1. Must be disable DMA during OSPI configuration, otherwise unexpected requests may be generated.
5. If OSPI is in memory-mapped mode, configure the TMOUTEN bit to select whether to enable the timeout counter, and set the TIMEOUT[15:0] bit field of the OSPI_LPTOUT register to configure the timeout value.
6. If need to use interrupt, the respective interrupt enable bit can be set.

27.5.2. OSPI device configuration

Configure parameters of OSPI and external device by OSPI_DCFG0 and OSPI_DCFG1 register, the details as follows:

1. Configure the external flash memory type by set the value of DTYSEL[2:0] bit field.
2. Configure the size of external flash memory by set the value of MESZ[4:0].
3. Configure chip-select minimum high time by set the value of CSHC[5:0].
4. Configure prescaler by set the value of PSC[7:0].

27.5.3. OSPI regular command configuration

Indirect mode

The details of the OSPI system configuration as follows:

1. Configure the data length by setting OSPI_DTLEN register.
2. Configure the frame timing by setting OSPI_TIMCFG register.
3. Configure the frame format by setting OSPI_TCFG register.
4. Specify the instruction to be sent to the external flash memory by setting OSPI_INS register.
5. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
6. Specify the address to be sent to the external flash memory by setting OSPI_ADDR register.
7. Read/write the data through OSPI_DATA register.

Status polling mode

When OSPI is in status polling mode, the details as follows:

1. Specify to mask the received status byte by setting OSPI_STATMK register.
2. Specify the value to be compared with the OSPI_STATMK register by setting OSPI_STATMATCH.
3. Specify the number of clock cycles between read operations by setting OSPI_INTERVAL register.
4. Configure the data length by setting OSPI_DTLEN register.

5. Configure the frame timing by setting OSPI_TIMCFG register.
6. Configure the frame format by setting OSPI_TCFG register.
7. Specify the instruction to be sent to the external flash memory by set OSPI_INS register.
8. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
9. Specify the address to be sent to the external flash memory by set OSPI_ADDR register.

Memory map mode

In memory map mode, external flash memory is accessed as internal memory. In this mode, the configuration of OSPI needs to be completed before accessing the memory for the first time. The specific configuration is as follows:

1. Configure the read operation frame timing by setting OSPI_TIMCFG register.
2. Configure the read frame format by setting OSPI_TCFG register.
3. Specify the instruction to be sent to the external flash memory by set OSPI_INS register.
4. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
5. Configure the write operation frame timing by setting OSPI_WTIMCFG register.
6. Configure the write frame format by setting OSPI_WTCFG register.
7. Specify the instruction to be sent to the external flash memory by set OSPI_WINS register.
8. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_WALTE register.

27.5.4. OSPI HyperBusTM mode configuration

Indirect mode

When OSPI is in indirect mode, The specific configuration is as follows:

1. Configure the data length by setting OSPI_DTLEN register.
2. Specify the address to be sent to the external flash memory by set OSPI_ADDR register.
3. Read/write the data through OSPI_DATA register.

Status polling mode

When OSPI is in status polling mode, the specific configuration is as follows:

1. Specify to mask the received status byte by setting OSPI_STATMK register.
2. Specify the value to be compared with the OSPI_STATMK register by setting OSPI_STATMATCH.
3. Specify the number of clock cycles between read operations by setting OSPI_INTERVAL register.
4. Configure the data length by setting OSPI_DTLEN register.
5. Specify the address to be sent to the external flash memory by set OSPI_ADDR register.

Memory map mode

In memory mapped mode, external flash memory is accessed as internal memory. In this mode, the configuration of OSPI needs to be completed before accessing the memory for the first time. The specific configuration is as follows:

1. Configure the read operation frame timing by setting OSPI_TIMCFG register.
2. Configure the read frame format by setting OSPI_TCFG register.
3. Specify the instruction to be sent to the external flash memory by set OSPI_INS register.
4. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_ALTE register.
5. Configure the write operation frame timing by setting OSPI_WTIMCFG register.
6. Configure the write frame format by setting OSPI_WTCFG register.
7. Specify the instruction to be sent to the external flash memory by set OSPI_WINS register.
8. Specify the alternate bytes to be sent to the external flash memory immediately after the address is sent by setting OSPI_WALTE register.

27.6. Data sampling shift

The OSPI samples data after a reset, driven by the external memory for 1/2 clock cycles. When sampling with the internal clock instead of the external device's DQS, the SSAMPLE bit in the OSPI_TIMCFG register can be configured to shift the data sampling by 1/2 clock cycle. Alternatively, combined with RXSFT[2:0] in OSPI_DCFG0 register, the sampling can be shifted by up to 7.5 cycles. When DQSEN = 1, the DQS signal can be offset to compensate for data propagation delays. This functionality can be executed and controlled by internally. By setting the RSCKDEN bit in OSPI DCFG0 via software, the delay module can be bypassed.

27.7. OSPI send instruction only once

Sending instruction only once is set by SIOO, this function is valid for memory-mapped mode only, if SIOO bit is set, the instruction is sent only once after OSPI_TCFG has be accessed. Subsequent command sequence skip instruction phase, until OSPI_TCFG is accessed again. SIOO has no affect when IMOD = 00.

27.8. Busy

BUSY bit is set once the OSPI start to operate the external flash memory.

In indirect mode, BUSY is reset if the command phase is end and the FIFO is empty. In status polling mode, when a match occurs or due to an abort, the BUSY bit will be reset. After the first access in memory-mapped mode, the BUSY bit will be reset only an abort occurs.

Setting the ABORT bit to 1 allows any operation to be aborted. Once the abort action is

completed, both BUSY and ABORT are reset, and the FIFO is flushed. Before setting ABORT to 1, the software must ensure that all current transactions are completed using synchronization barriers. If a write operation to a status register is aborted, some devices may exhibit abnormal behavior.

27.9. Error management

An error can be generated in the following case.

- In indirect or status polling, TERR is generated immediately when a wrong address has been programed in OSPI_ADDR register according to MESZ[4:0].
- In indirect mode, if the address (ADDR) plus data length (DTLEN) is greater than external memory size, TERR will be set once the OSPI is triggered.
- In memory-mapped mode, when an out of range access is done by AHB master will generate an AHB error.
- When an AHB master is accessing the memory-mapped space while the memory mapped mode is not enabled, will generate an AHB error.

The OSPI generate an AHB BUS error in following case:

- When an AHB read request while the memory-mapped mode is not enabled, will generate an AHB error.
- Read or write address exceeds the size of external memory.
- During a read or write, an abort occurs.
- During a read or write, OSPI is disabled.
- Write wrap burst received.

27.10. OSPI interrupt

Table 27-4. OSPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
FT	FIFO threshold	By hardware	FTIE
TC	Transfer complete	Set TCC bit in OSPI_STATC register	TCIE
TERR	Transfer error	Set TERRC bit in OSPI_STATC register	TERRIE
TMOUT	Timeout	Set TMOUTC bit in OSPI_STATC register	TMOUTIE
SM	Status match	Set SMC bit in OSPI_STATC register	SMIE

27.11. Register definition

OSPI base address: 0x4002 5000

27.11.1. Control register (OSPI_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1 (Except for bit0 and bit1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCSNEN	CSN1EN	FMOD[1:0]		RCKDV[3:0]				SPMOD	SPS	Reserved	TMOUTIE	SMIE	FTIE	TCIE	TERRIE
rw	rw	rw		rw				rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				FTL[4:0]				DEVS	Reserved	DQSOEN	Reserved	TMOUTEN	DMAEN	ABORT	OSPIEN
				rw				rw		rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31	DCSNEN	Enable dual CSN mode 0: Disable dual CSN mode 1: Enable dual CSN mode
30	CSN1EN	Select to fix a specific CSN output (when not in memory-mapped mode) 0: Fix CSN0 output 1: Fix CSN1 output
29:28	FMOD[1:0]	Functional mode 00: Indirect write mode 01: Indirect read mode 10: Status polling mode 11: Memory mapped mode If DMAEN = 1, then the DMA controller for the corresponding channel must be disabled before changing the FMOD[1:0] value. If FMODE[1:0] and FTHRES[4:0] are wrongly updated while DMAEN = 1, the DMA request signal automatically goes to inactive state.
27:24	RCKDV[3:0]	Receive SCK delay value 0000: Delay 1 cell 0001: Delay 2 cell …… 1111: Delay 16 cell
23	SPMOD	Status polling match mode

		This bit indicates which method must be used to determine a match during the status polling mode. 0: AND match mode. SM is set if all the unmasked bits received from the flash memory match the corresponding bits in the match register. 1: OR match mode. SM is set if any one of the unmasked bits received from the flash memory matches its corresponding bit in the match register.
22	SPS	Status polling mode stop This bit determines if status polling is stopped after a match. 0: Status polling mode stops by abort or OSPI is disabled. 1: Status polling mode stops as soon as there is a match.
21	Reserved	Must be kept at reset value.
20	TMOUTIE	Timeout interrupt enable 0: Disable timeout interrupt. 1: Enable timeout interrupt.
19	SMIE	Status match interrupt enable 0: Disable status match interrupt. 1: Enable status match interrupt.
18	FTIE	FIFO threshold interrupt enable 0: Disable FIFO threshold interrupt. 1: Enable FIFO threshold interrupt.
17	TCIE	Transfer complete interrupt enable 0: Disable transfer complete interrupt. 1: Enable transfer complete interrupt.
16	TERRIE	Transfer error interrupt enable 0: Disable transfer error interrupt. 1: Enable transfer error interrupt.
15:13	Reserved	Must be kept at reset value.
12:8	FTL[4:0]	FIFO threshold level This bit are useful in indirect mode, the threshold number of bytes in the FIFO that will cause the FIFO threshold flag to be set. In indirect write mode (FMOD = 00): 0: FT is set if there are 1 or more free bytes available to be written to the FIFO. 1: FT is set if there are 2 or more free bytes available to be written to the FIFO. ... 31: FT is set if there are 32 free bytes available to be written to the FIFO. In indirect read mode (FMOD = 01): 0: FT is set if there are 1 or more free bytes available to be read from the FIFO. 1: FT is set if there are 2 or more free bytes available to be read from the FIFO. ...

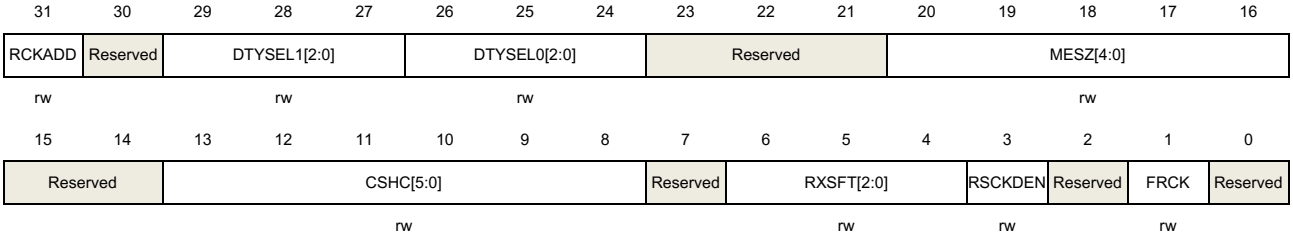
		31: FT is set if there are 32 free bytes available to be read from the FIFO. If DMAEN = 1, then the DMA controller for the corresponding channel must be disabled before changing the FTL value.
7	DEVS	Device select This bit selects the device to be addressed in single/dual/quad mode in single memory mode 0: Select device 0, transfer data through IO[3:0]. 1: Select device 1, transfer data through IO[7:4]. If OSPI is in octal mode, this bit is ignored.
6	Reserved	Must be kept at reset value.
5	DQSOEN	DQS output enable Setting this bit to 1 will output the DQS signal as SCK during write operations. 0: Disable DQS output 1: Enable DQS output
4	Reserved	Must be kept at reset value.
3	TMOUTEN	Timeout counter enable This bit is valid only in memory-mapped mode. If there is no access to external flash, activating this bit will release CSN after the time defined by TIMEOUT[15:0]. 0: Timeout counter is disabled, and thus the chip select (CSN) remains active indefinitely after an access in memory-mapped mode. 1: Timeout counter is enabled, and thus the chip select is released in memory - mapped mode after TIMEOUT[15:0] cycles of flash memory inactivity.
2	DMAEN	DMA enable In indirect mode, DMA can be used to transfer data through OSPI_DATA register. DMA transfers are initiated when FT is set. 0: Disable DMA. 1: Enable DMA. **Note:** Resetting the DMAEN bit while a DMA transfer is ongoing, breaks the handshake with the DMA. Do not write this bit during DMA operation.
1	ABORT	Abort request This bit aborts the on-going command sequence. It is automatically reset once the abort is completed. This bit stops the current transfer. 0: No abort requested 1: Abort requested **Note:** This bit is always read as 0.
0	OSPIEN	Enable OSPI 0: Disable OSPI. 1: Enable OSPI. **Note:** The DMA request can be aborted without having received the ACK in case

this OSPIEN bit is cleared during the operation.
In case this bit is set to 0 during a DMA transfer, the REQ signal to DMA returns to inactive state without waiting for the ACK signal from DMA to be active.

27.11.2. Device configuration register 0 (OSPI_DCFG0)

Address offset: 0x08
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31	RCKADD	An additional clock is sent during reception, but the received data remains unchanged. To prevent DQS from missing an edge, configuration is required when choosing DQS for reception. 0: No additional clock 1: One additional clock
30:27	Reserved	Must be kept at reset value.
29:27	DTYSEL1[2:0]	When dual CSN mode is enabled, the Memory type corresponding to CSN1 same as DTYSEL
26:24	DTYSEL0[2:0]	Select device type 000: Micron mode, D0/D1 ordering in DTR 8 bit data mode. Regular SPI protocol in single/dual/quad/octal line mode. **Note:** This is the default value and care must be taken to change DTYSEL0[2:0] for memories different from Micron. 001: Macronix mode, D1/D0 ordering in DTR 8 bit data mode. Regular SPI protocol in single/dual/quad/octal line mode. 010: Standard mode. 011: Macronix RAM mode, D1/D0 ordering in DTR 8 bit data mode. Regular SPI protocol in single/dual/quad/octal line mode with dedicated address mapping. 100: HyperBus memory mode, the protocol follows the HyperBus specification. 8-data-bit DTR mode must be selected. 101: HyperBus register mode, addressing register space. The memory-mapped accesses in this mode must be non-cacheable, or Indirect read/write modes must be used.

		Others: Reserved
23:21	Reserved	Must be kept at reset value.
20:16	MESZ[4:0]	Memory size This field defines the size of external memory using the following formula: Number of bytes in memory = $2^{[MESZ[4:0]+1]}$ MESZ[4:0]+1 is effectively the number of address bits in the memory. The memory capacity can be up to 4GB in indirect mode, while it is limited to 256MB in memory mapped mode.
15:14	Reserved	Must be kept at reset value.
13:8	CSHC[5:0]	Chip select high cycle CSHC+1 defines the minimum number of SCK cycle which the CSN must stay high between two command sequences. 0: CSN stays high for at least 1 cycle between memory commands. 1: CSN stays high for at least 2 cycles between memory commands. ... 63: CSN stays high for at least 64 cycles between memory commands.
7	Reserved	Must be kept at reset value.
6:4	RXSFT[2:0]	Shift receive step if receive data delay more than 1/2 cycle. To configure this bit fields, ensure that the CSHC configuration time is at least 3 cycles longer than the bit fields. 000: No shift 001: 1 cycle 010: 2 cycles 011: 3 cycles 100: 4 cycles 101: 5 cycles 110: 6 cycles 111: 7 cycles
3	RSCKDEN	Sampling delay enable 0: Disable sampling delay 1: Enable sampling delay
2	Reserved	Must be kept at reset value.
1	FRCK	This bit indicates the SCK is free running or not when OSPI is free. 0: Disable SCK free running. 1: Enable SCK free running.
0	Reserved	Must be kept at reset value.

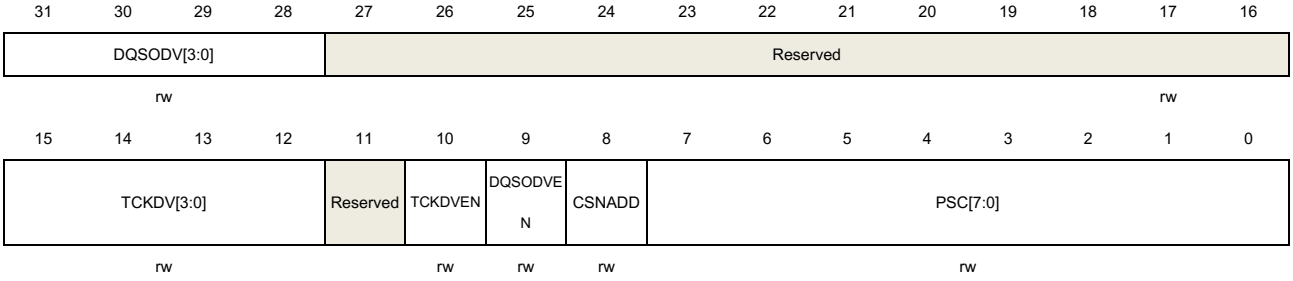
27.11.3. Device configuration register 1 (OSPI_DCFG1)

Address offset: 0x0C

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:28	DQSODV[3:0]	DQS out transfer SCK delay chain value 0000: Enable 1 delay cell 0001: Enable 2 delay cell …… 1111: Enable 16 delay cell
27:16	Reserved	Must be kept at reset value.
15:12	TCKDV[3:0]	SCK transfer delay chain value 0000: Enable 1 delay cell 0001: Enable 2 delay cell …… 1111: Enable 16 delay cell
11	Reserved	Must be kept at reset value.
10	TCKDVEN	SCK transfer delay enable 0: Disable sck transfer delay 1: Enable sck transfer delay
9	DQSODVEN	DQS out transfer SCK delay enable 0: Disable DQS transfer delay 1: Enable DQS transfer delay
8	CSNADD	CSN add 1/2 cycle 0: No add 1: Add 1/2 cycle
7:0	PSC[7:0]	This field defines the scaler factor for generating SCK based on the kernel clock (value+1) 0: $F_{CLK} = F_{KERNEL}$. In this case, DTR mode is not supported.

1: $F_{CLK} = F_{KERNEL} / 2$

2: $F_{CLK} = F_{KERNEL} / 3$

…

255: $F_{CLK} = F_{KERNEL} / 256$

For odd clock division factors, SCK's duty cycle is not 50%. The clock signal remains high one cycle longer than it stays low.

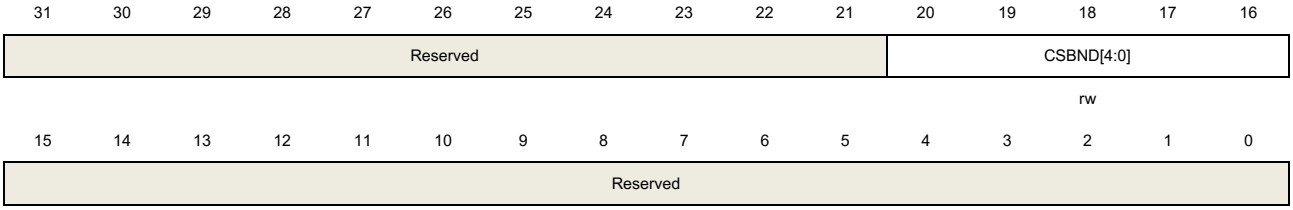
27.11.4. Device configuration register 2 (OSPI_DCFG2)

Address offset: 0x10

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	CSBND[4:0]	Chip select boundary This field is used to enable the transaction boundary feature. When enabled, it is recommended to set the minimum value to 3. At the boundary of every 2^{CSBND} bytes, the nCS signal will be released. 0: Disable the CS boundary feature Other values: Set the CS boundary to 2^{CSBND} bytes
15:0	Reserved	Must be kept at reset value.

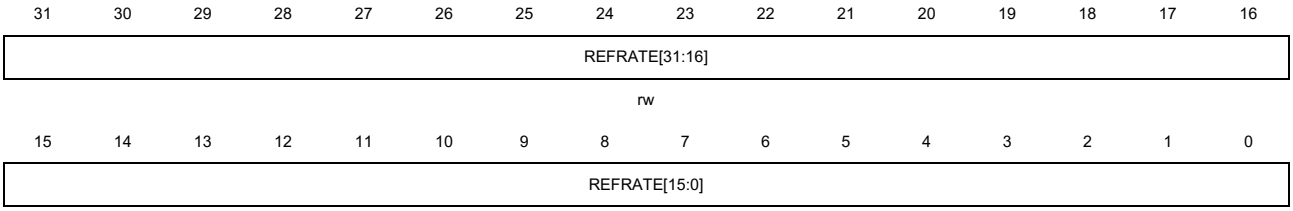
27.11.5. Device configuration register 3 (OSPI_DCFG3)

Address offset: 0x14

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



rw

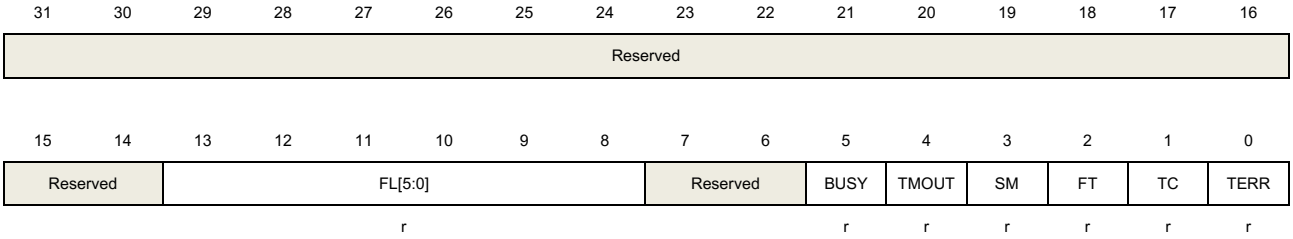
Bits	Fields	Descriptions
31:0	REFRATE[31:0]	Refresh rate Only for 8 lines read in memory-mapped mode 0: Disable refresh others: The maximum communication length is REFRATE+1 clock cycles.

27.11.6. Status register (OSPI_STAT)

Address offset: 0x20

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.
13:8	FL[5:0]	FIFO level This field gives the number of valid bytes which are being held in the FIFO. In status polling mode, FL is zero.
7:6	Reserved	Must be kept at reset value.
5	BUSY	Busy This bit is set when a command is transferring. This bit is cleared once the operation with the flash memory is finished and the FIFO is empty.
4	TMOUT	Timeout flag This bit is set when timeout occurs. It is cleared by writing 1 to TMOUTC. When this bit is set, read operations in memory-mapped mode no longer guarantee data correctness. This bit must be cleared before the next transfer.
3	SM	Status match flag This bit is set in status polling mode when the unmasked received data matches the expected value. It is cleared by writing 1 to SMC.
2	FT	FIFO threshold flag In indirect mode, this bit is set when the FIFO threshold has been reached, or if the FIFO is not empty after the last read operation from the flash memory.

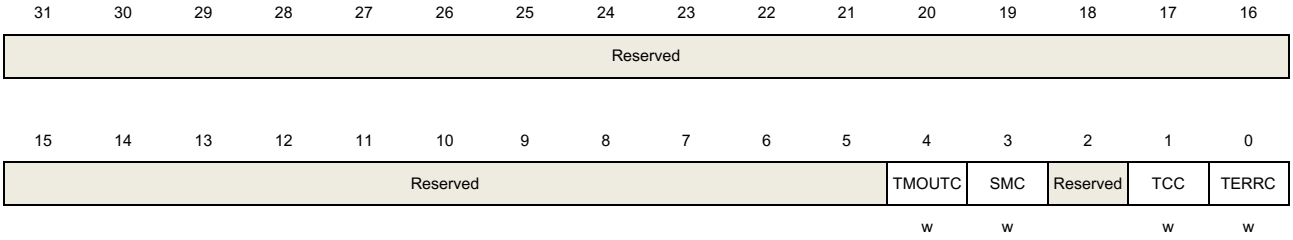
		In status polling mode, this bit is set when the status register is read from the external flash, and it is cleared when OSPI_DATA is read.
1	TC	Transfer complete flag This bit is set in indirect mode when the programmed number of data has been transmitted. It is cleared by writing 1 TCC.
0	TERR	Transfer error flag This bit is set when an invalid address is being accessed in indirect mode. It is cleared by writing 1 to TERRC.

27.11.7. Status clear register (OSPI_STATC)

Address offset: 0x24

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TMOUTC	Clear timeout flag Writing 1 clears the TMOUT flag in OSPI_STAT register.
3	SMC	Clear status match flag Writing 1 clears the SM flag in the OSPI_STAT register.
2	Reserved	Must be kept at reset value.
1	TCC	Clear transfer complete flag Writing 1 clears the TC flag in the OSPI_STAT register.
0	TERRC	Clear transfer error flag Writing 1 clears the TERR flag in the OSPI_STAT register.

27.11.8. Data length register (OSPI_DTLEN)

Address offset: 0x40

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

Bits	Fields	Descriptions
31:0	DTLEN[31:0]	Data length The data length is DTLEN+1 in indirect and status poling mode. The value of DTLEN no greater than 3 should be used for status polling mode. All 1's in indirect mode means undefined length, where OSPI will continue until the end of memory, as defined by MESZ[4:0]. 0x0000 0000: 1 byte will be transferred 0x0000 0001: 2 bytes will be transferred 0x0000 0002: 3 bytes will be transferred 0x0000 0003: 4 bytes will be transferred ... 0xFFFF FFFD: 4,294,967,294 (4G-2) bytes will be transferred 0xFFFF FFFE: 4,294,967,295 (4G-1) bytes will be transferred 0xFFFF_FFFF: undefined length - all bytes until the end of flash memory (as defined by MESZ[4:0]) are to be transferred. Continue reading indefinitely if MESZ[4:0] = 0x1F. This field has no effect when in memory-mapped mode. **Note:** DTR mode does not support odd byte numbers, configuring odd bytes may cause unexpected errors.

27.11.9. Address register(OSPI_ADDR)

Address offset: 0x48

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

Bits	Fields	Descriptions
31:0	ADDR[31:0]	Address

Address to be send to the external flash memory.

In HyperBus™ mode, the value written to this register must be an even number. This bit field can only be written when the BUSY bit is 0 and the memory-mapped mode is not configured.

27.11.10. Data register (OSPI_DATA)

Address offset: 0x50

Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data Data to be transferred through the flash memory In indirect write mode, data written to this register is stored on the FIFO before sent to the flash memory. If the FIFO is full, a write operation is stalled until the FIFO has enough space. In indirect read mode, reading this register gives the data received from the flash memory. If the FIFO does not have as many bytes as requested by the read command and if BUSY=1, the read operation is stalled until enough data is present or until the transfer is complete. In status polling mode, this register contains the last data read from the flash memory.

27.11.11. Status mask register (OSPI_STATMK)

Address offset: 0x80

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	MASK[31:0]	Status mask Mask to be applied to the status bytes received from the flash memory. For bit n: 0: Bit n of the data received is masked and its value is not considered in the matching logic. 1: Bit n of the data received is unmasked and its considered in the matching logic.

27.11.12. Status match register(OSPI_STATMATCH)

Address offset: 0x88

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

Bits	Fields	Descriptions
31:0	MATCH[31:0]	Status match Expected value to be compared with the masked status register to get a match.

27.11.13. Interval register (OSPI_INTERVAL)

Address offset: 0x90

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

15:0	INTERVAL[15:0]	Interval cycle Number of SCK cycles between two read commands in status polling mode.

27.11.14. Transfer configuration0 register (OSPI_TCFG0)

Address offset: 0x100

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOO	Reserved	DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
rw		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31	SIOO	Send instruction only once mode This bit has no effect when IMOD = 00. 0: Send instruction on every command sequence. 1: Send instruction only for the first command sequence.
30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS. 1: Enable DQS.
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase 1: Enable DTR mode in data phase
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. Others: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size

		This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. Others: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase 1: Enable DTR mode in address phase
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. Others: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction.

		10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. Others: Reserved.

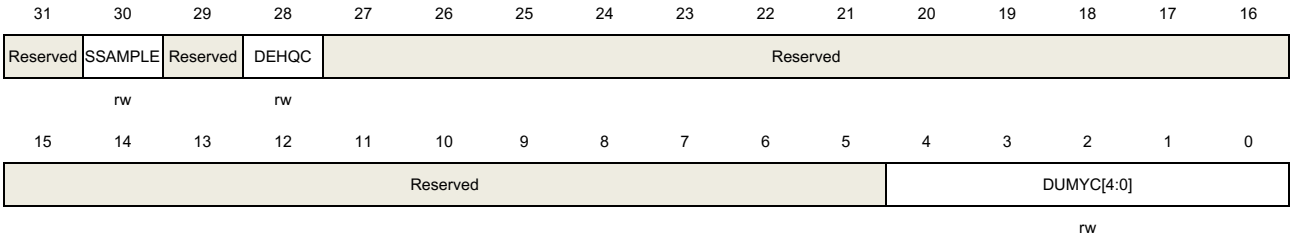
27.11.15. Timing configuration0 register (OSPI_TIMCFG0)

Address offset: 0x108

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	SSAMPLE	Sample shift By default, the OSPI samples data 1/2 of a SCK cycle after the data is driven by the flash memory. This bit allows the data is to be sampled later in order to account for external signal delays. 0: No shift. 1: 1/2 cycle shift.
29	Reserved	Must be kept at reset value.
28	DEHQC	Delay hold 1/4 cycle 0: No delay hold. 1: delay hold 1/4 cycle. **Note:** to ensure correct data sampling, this bit must be set to 1.

27:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 6 dummy cycles are required when DQS is used.

27.11.16. Instruction0 register (OSPI_INS0)

Address offset: 0x110
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

27.11.17. Alternate bytes0 register (OSPI_ALTE0)

Address offset: 0x120
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

27.11.18. Low power timeout register (OSPI_LPOUT)

Address offset: 0x130
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	TIMEOUT[15:0]	Timeout cycle After access when OSPI works in memory mapped mode, this field indicates how many SCK cycles the OSPI waits after the clock becomes inactive and until raises the CSN, and the external memory is set to a low power consumption state.

27.11.19. Wrap transfer configuration0 register (OSPI_WPTCFG0)

Address offset: 0x140
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS 1: Enable DQS
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate

		0: Disable DTR mode in data phase. 1: Enable DTR mode in data phase.
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. Others: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. Others: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase 1: Enable DTR mode in address phase
10:8	ADDRMOD[2:0]	Address mode

		This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. Others: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. Others: Reserved.

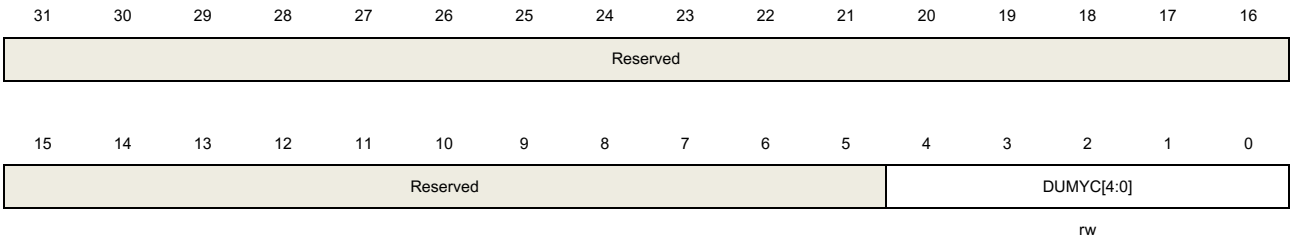
27.11.20. Wrap timing configuration0 register (OSPI_WPTIMCFG0)

Address offset: 0x148

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



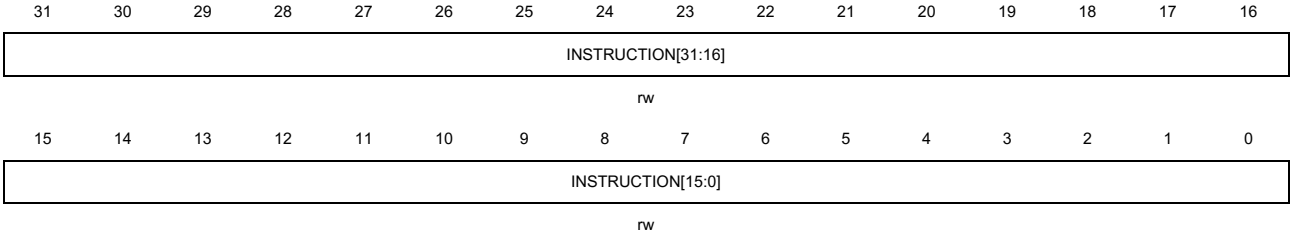
Bits	Fields	Descriptions

31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

27.11.21. Wrap instruction0 register (OSPI_WPINS0)

Address offset: 0x150
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

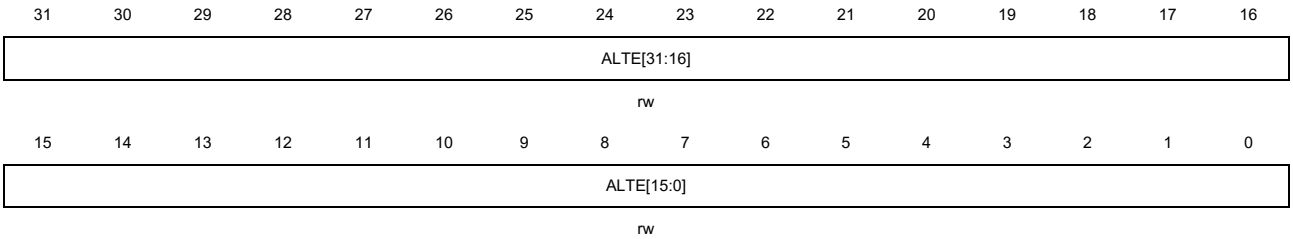


Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

27.11.22. Wrap alternate byte0 register (OSPI_WPALTE0)

Address offset: 0x160
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

27.11.23. Write transfer configuration0 register (OSPI_WTCFG0)

Address offset: 0x180

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS 1: Enable DQS
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase. 1: Enable DTR mode in data phase.
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. Others: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase

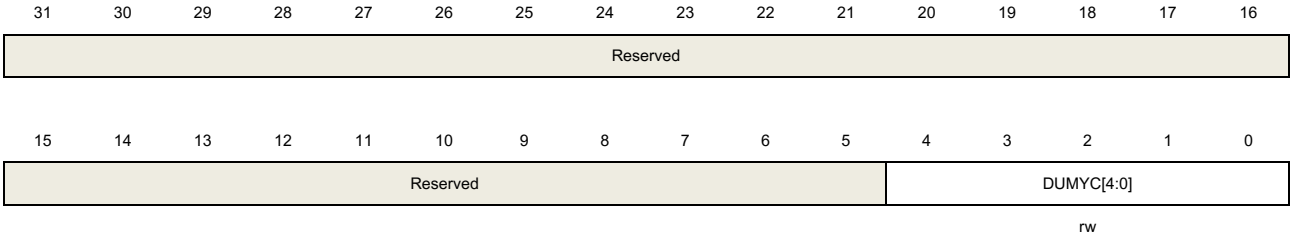
		1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. Others: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase. 1: Enable DTR mode in address phase.
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. Others: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation.

000: No instruction.
001: Instruction on a single line.
010: Instruction on two lines.
011: Instruction on four lines.
100: Instruction on eight lines.
Others: Reserved.

27.11.24. Write timing configuration0 register (OSPI_WTIMCFG0)

Address offset: 0x188
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

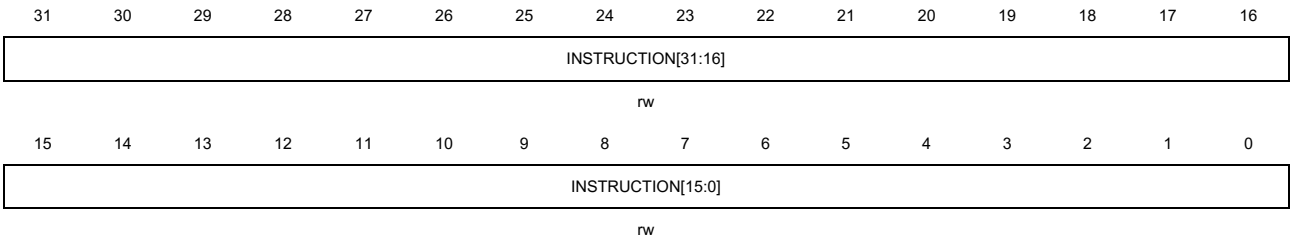


Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

27.11.25. Write instruction0 register (OSPI_WINS0)

Address offset: 0x190
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction

Instruction to be send to the external flash memory.

27.11.26. Write alternate byte0 register (OSPI_WALTE0)

Address offset: 0x1A0

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

27.11.27. HyperBusTM latency configuration register (OSPI_HBLCFG)

Address offset: 0x200

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:16	RWRTM[7:0]	Read write recovery time The number of SCK cycles indicates the read and write recovery time.
15:8	ACCTM[7:0]	Access time The number of SCK cycles indicates the access time.
7:2	Reserved	Must be kept at reset value.
1	WZLAT	Write zero latency

		0: Disable write operation zero latency. 1: Enable write operation zero latency.
0	LMOD	Latency mode 0: Variable initial latency. 1: Fixed latency.

27.11.28. Transfer configuration1 register (OSPI_TCFG1)

Address offset: 0x300
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOO	Reserved	DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[2:0]		
rw		rw		rw	rw					rw		rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31	SIOO	Send instruction only once mode This bit has no effect when IMOD = 00. 0: Send instruction on every command sequence. 1: Send instruction only for the first command sequence.
30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS. 1: Enable DQS.
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase 1: Enable DTR mode in data phase
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines.

		Others: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. Others: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase 1: Enable DTR mode in address phase
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. Others: Reserved.
7:6	Reserved	Must be kept at reset value.

5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. Others: Reserved.

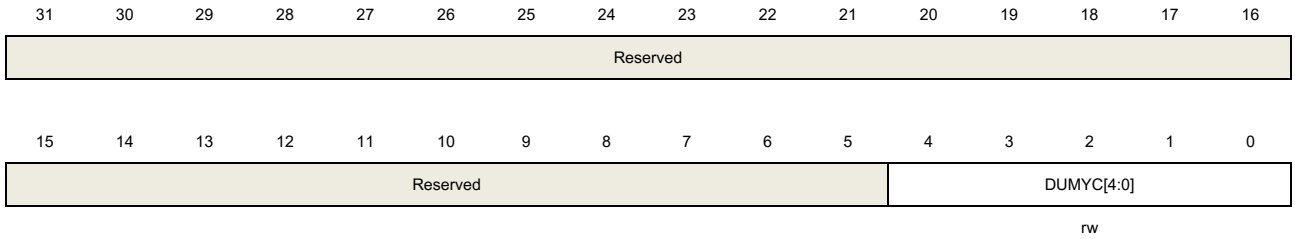
27.11.29. Timing configuration1 register (OSPI_TIMCFG1)

Address offset: 0x304

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

27.11.30. Instruction1 register (OSPI_INS1)

Address offset: 0x308

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

27.11.31. Alternate1 bytes register (OSPI_ALTE1)

Address offset: 0x30C

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

27.11.32. Wrap transfer configuration1 register (OSPI_WPTCFG1)

Address offset: 0x310

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS 1: Enable DQS
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase. 1: Enable DTR mode in data phase.
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. Others: Reserved.
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. Others: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size

		This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase 1: Enable DTR mode in address phase
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. Others: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction. 10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. Others: Reserved.

27.11.33. Wrap timing configuration1 register (OSPI_WPTIMCFG1)

Address offset: 0x314
Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

27.11.34. Wrap instruction1 register (OSPI_WPINS1)

Address offset: 0x318
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

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

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

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

27.11.35. Wrap alternate1 byte register (OSPI_WPALTE1)

Address offset: 0x31C
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).
This register cannot be modified when the BUSY bit is 1.

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

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

ALTE[15:0]
rw

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

27.11.36. Write transfer configuration1 register (OSPI_WTCFG1)

Address offset: 0x320

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DQSEN	Reserved	DADTR	DATAMOD[2:0]			Reserved		ALTESZ[1:0]		ABDTR	ALTEMOD[1:0]		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ADDRSZ[1:0]		ADDRDTR	ADDRMOD[2:0]			Reserved		INSSZ[1:0]		INSDTR	IMOD[2:0]		
		rw		rw	rw					rw		rw	rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	DQSEN	DQS enable 0: Disable DQS 1: Enable DQS
28	Reserved	Must be kept at reset value.
27	DADTR	Data double transfer rate 0: Disable DTR mode in data phase. 1: Enable DTR mode in data phase.
26:24	DATAMOD[2:0]	Data mode This bit defines the operation mode of the data phase. 000: No data. 001: Data on a single line. 010: Data on two lines. 011: Data on four lines. 100: Data on eight lines. Others: Reserved
23:22	Reserved	Must be kept at reset value.
21:20	ALTESZ[1:0]	Alternate bytes size

		This bit defines alternate bytes size. 00: 8-bit alternate byte 01: 16-bit alternate bytes 10: 24-bit alternate bytes 11: 32-bit alternate bytes
19	ABDTR	Alternate bytes double transfer rate 0: Disable DTR mode in alternate bytes phase 1: Enable DTR mode in alternate bytes phase
18:16	ALTEMOD[2:0]	Alternate bytes mode This field defines the alternate bytes phase mode of operation. 000: No alternate bytes. 001: Alternate bytes on a single line. 010: Alternate bytes on two lines. 011: Alternate bytes on four lines. 100: Alternate bytes on eight lines. Others: Reserved.
15:14	Reserved	Must be kept at reset value.
13:12	ADDRSZ[1:0]	Address size This bit defines address size. 00: 8-bit address. 01: 16-bit address. 10: 24-bit address. 11: 32-bit address.
11	ADDRDTR	Address double transfer rate 0: Disable DTR mode in address phase. 1: Enable DTR mode in address phase.
10:8	ADDRMOD[2:0]	Address mode This field defines the address phase mode of operation. 000: No address. 001: Address on a single line. 010: Address on two lines. 011: Address on four lines. 100: Address on eight lines. Others: Reserved.
7:6	Reserved	Must be kept at reset value.
5:4	INSSZ[1:0]	Instruction size This field defines instruction size. 00: 8-bit instruction. 01: 16-bit instruction.

		10: 24-bit instruction. 11: 32-bit instruction.
3	INSDTR	Instruction double transfer rate 0: Disable DTR mode in instruction phase. 1: Enable DTR mode in instruction phase.
2:0	IMOD[2:0]	Instruction mode This field defines the instruction phase mode of operation. 000: No instruction. 001: Instruction on a single line. 010: Instruction on two lines. 011: Instruction on four lines. 100: Instruction on eight lines. Others: Reserved.

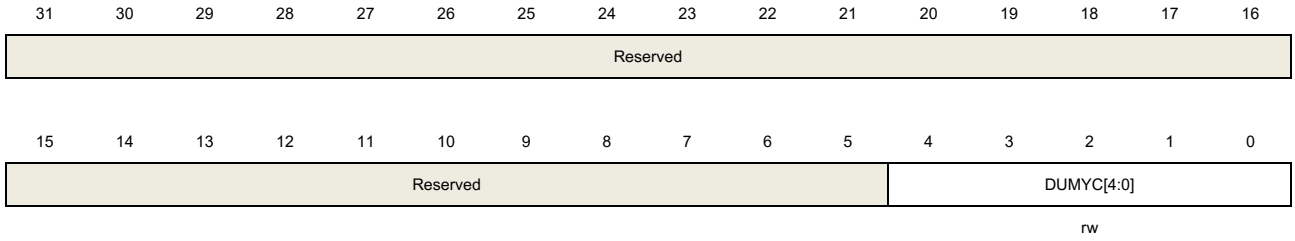
27.11.37. Write timing configuration1 register (OSPI_WTIMCFG1)

Address offset: 0x324

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4:0	DUMYC[4:0]	Number of dummy cycles This bit field defines the duration of the dummy instruction phase, at least 5 dummy cycles are required when DQS is used.

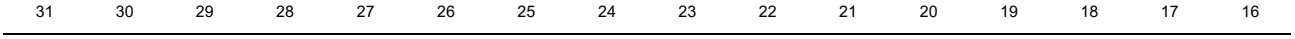
27.11.38. Write instruction1 register (OSPI_WINS1)

Address offset: 0x328

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.



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

Bits	Fields	Descriptions
31:0	INSTRUCTION[31:0]	Instruction Instruction to be send to the external flash memory.

27.11.39. Write alternate1 byte register (OSPI_WALTE1)

Address offset: 0x32C

Reset value: 0x0000 0000

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

This register cannot be modified when the BUSY bit is 1.

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

Bits	Fields	Descriptions
31:0	ALTE[31:0]	Alternate bytes Alternate bytes to be send to the external flash memory.

28. External memory controller (EXMC)

28.1. 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. The 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.

28.2. Characteristics

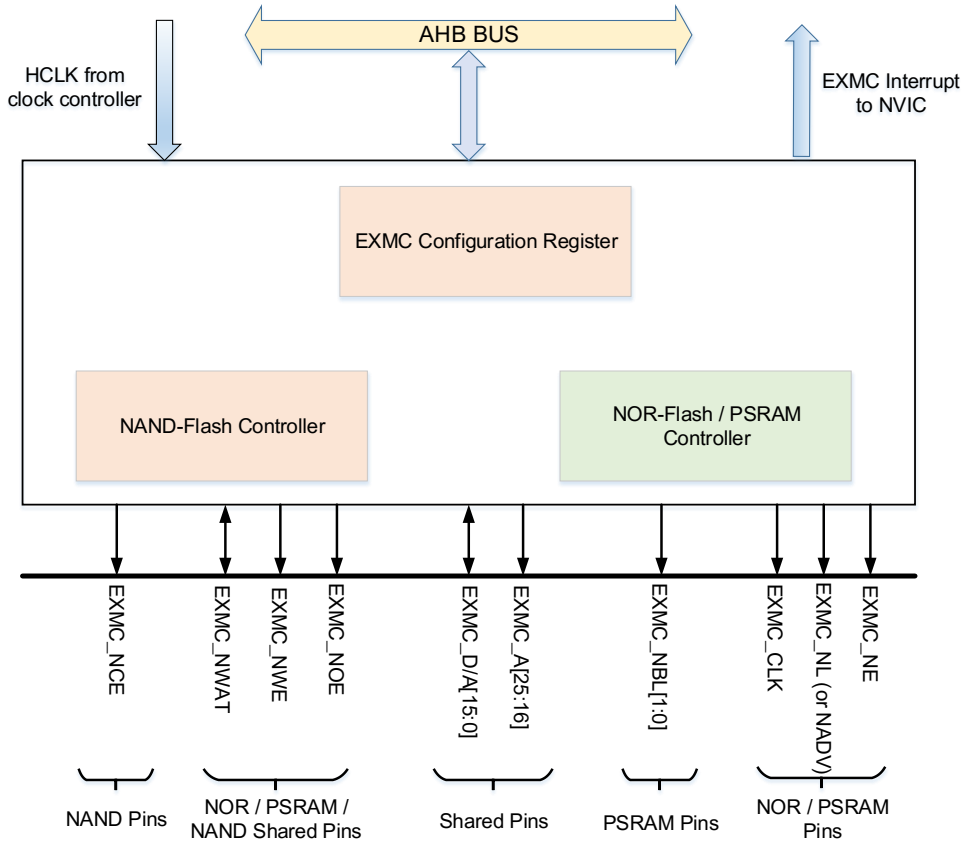
- Supported external memory:
  - PSRAM
  - NOR Flash
  - 8-bit or 16-bit NAND Flash
- Protocol translation between the AMBA and the multitude of external memory protocol.
- Offering a variety of programmable timing parameters to meet user's specific needs.
- Each bank has its own chip-select signal which can be configured independently.
- Embedded ECC hardware for NAND Flash access.
- 8 or 16 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.

28.3. Function overview

28.3.1. 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. The AHB clock (HCLK) is the reference clock.

Figure 28-1. The EXMC block diagram



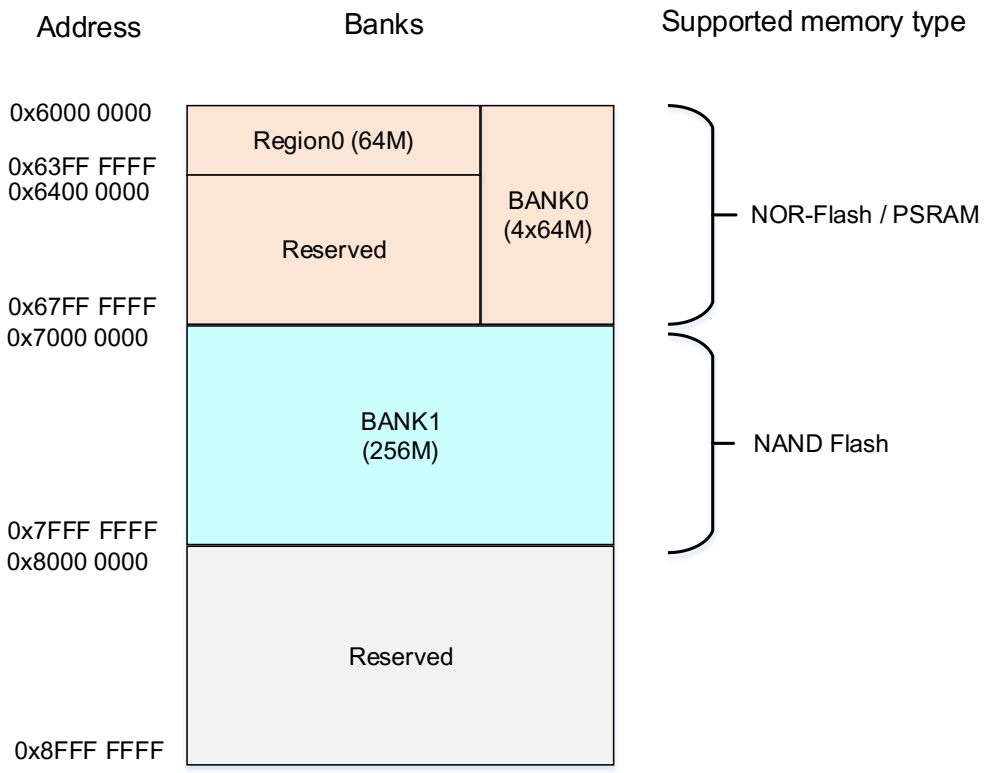
28.3.2. Basic regulation of EXMC access

EXMC is the conversion interface between AHB bus and external device protocol. 32-bit 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 the 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 SRAM, ROM. 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.

28.3.3. External device address mapping

Figure 28-2. 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.

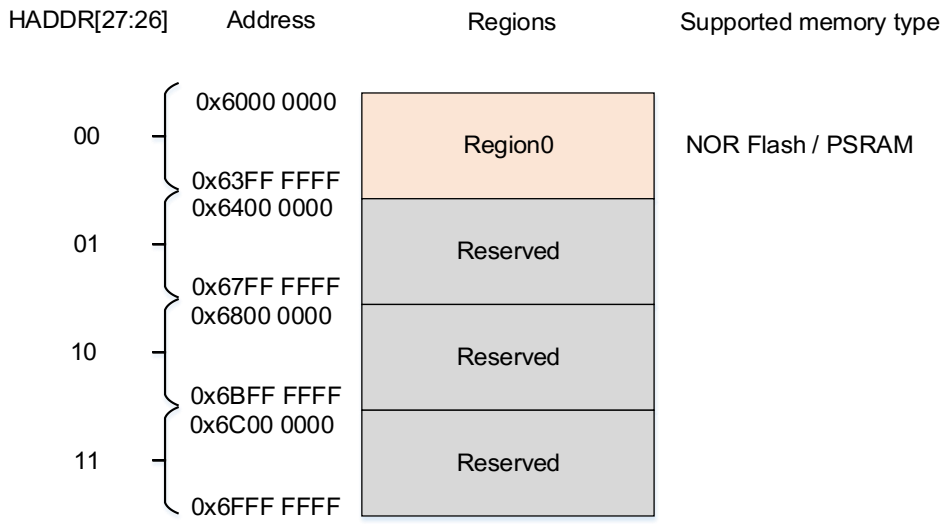
Bank0 is used for NOR-Flash and PSRAM device access.

Bank1 is used to access NAND Flash exclusively.

NOR/PSRAM address mapping

Figure 28-3. Four regions of bank0 address mapping reflects the address mapping of the two regions of bank0. Internal AHB address lines HADDR[27:26] bit are used to select the two regions.

Figure 28-3. Four regions of bank0 address mapping



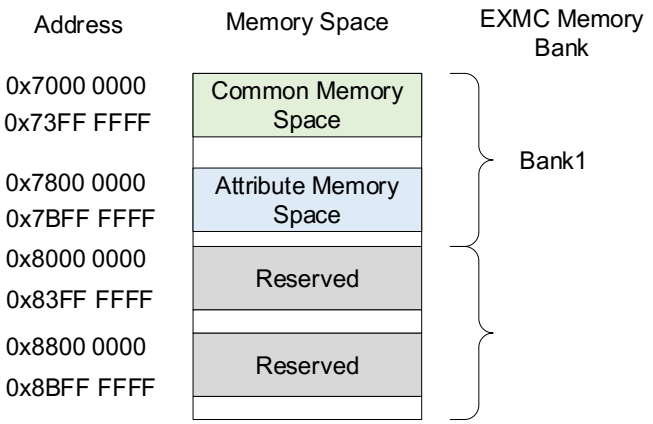
HADDR[25:0] is the byte address whereas the external memory may not be byte accessed, this will lead to address inconsistency. EXMC can adjust HADDR to accommodate the data width of the external memory according to the following rules.

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

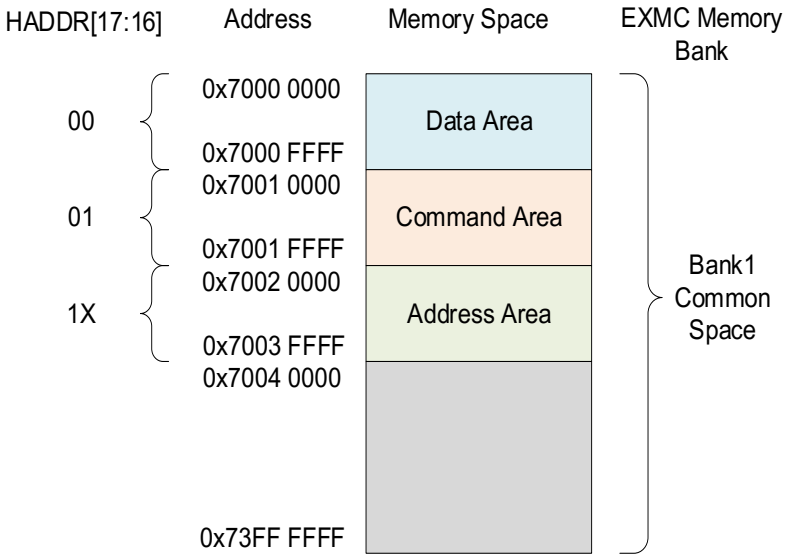
Bank1 is designed to access NAND Flash. Each bank is further divided into several memory spaces as shown in ***Figure 28-4. NAND address mapping***.

Figure 28-4. 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 28-5. Diagram of bank1 common space.***

Figure 28-5. 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. 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.

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. 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].

28.3.4. NOR/PSRAM controller

NOR/PSRAM memory controller controls bank0, which is designed to support NOR Flash, PSRAM. EXMC has 2 independent chip-select signals for each of the 2 regions within bank0, named EXMC_NE[x] (x = 0,1). Other signals for NOR/PSRAM access are shared. Each 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 28-1. 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[x]	Output	Async/Sync	Chip selection x=0/1
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 28-2. PSRAM non-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[x]	Output	Async/Sync	Chip selection x=0/1
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 below shows an example of the supported devices type, access modes and transactions when the memory data bus is 16-bit for NOR and PSRAM.

Table 28-3. 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	
	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 SRAM, PSRAM, NOR Flash and other external static memory.

Table 28-4. 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 28-5. EXMC_timing models

Timing model		Mode description	Write timing parameter	Read timing parameter
Async	Mode AM	NOR Flash address/data mux	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
Sync	Mode SM	NOR Flash (read) / PSRAM(CRAM) address/data mux	DLAT CKDIV	DLAT CKDIV

As shown in ***Table 28-5. EXMC_timing models***, EXMC NOR Flash / PSRAM controller provides a variety of timing model, users can modify those parameters listed in ***Table 28-4. 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 28-6. Multiplex mode read access

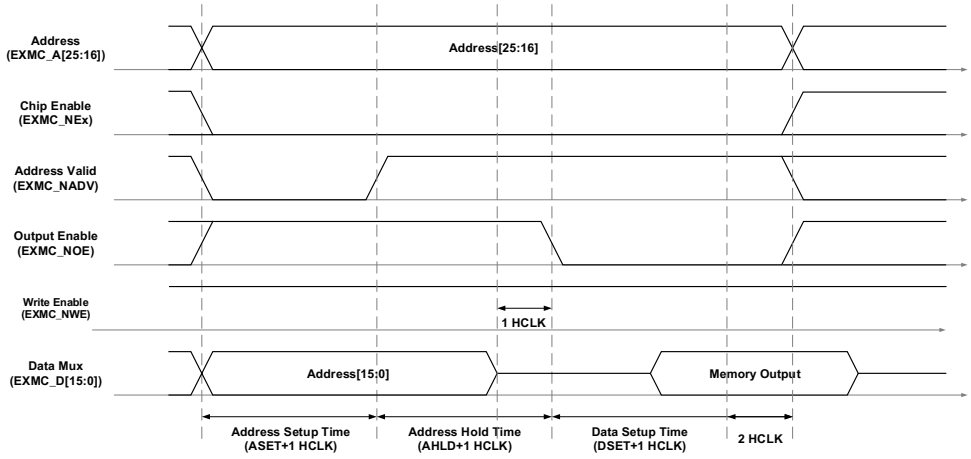


Figure 28-7. Multiplex mode write access

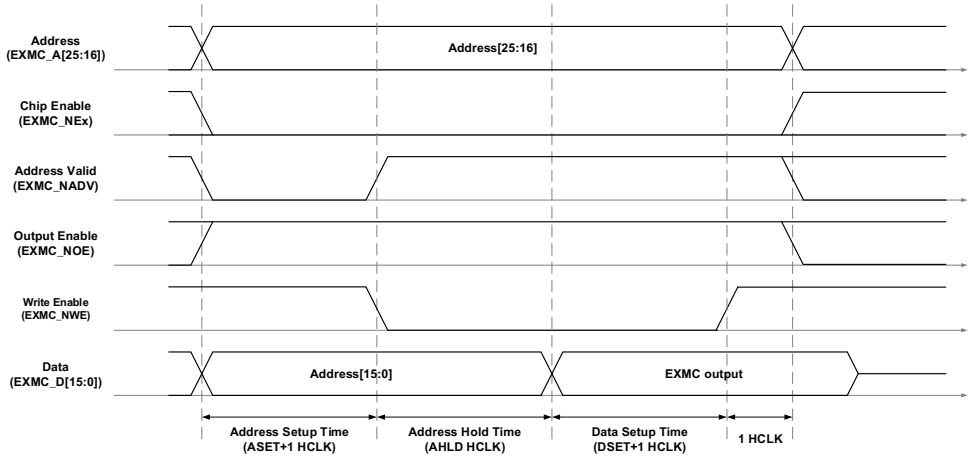


Table 28-6. Multiplex mode related registers configuration

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTLx		
31-20	Reserved	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	Depends on memory
14	Reserved	0x0
13	NRWTEN	0x0
12	WREN	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

Bit Position	Bit Name	Reference Setting Value
6	NREN	0x1
5-4	NRW	Depends on memory
3-2	NRTP	0x2:NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_PNTCFGx		
31-28	Reserved	0x0
27-24	DLAT	No effect
23-20	CKDIV	No effect
19-16	BUSLAT	Minimum time between EXMC_NE[x] rising edge to EXMC_NE[x] 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_PNCTLx. 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:

$$T_{DATA_SETUP} >= maxT_{WAIT_ASSERTION} + 4HCLK \quad (28\text{-}1)$$

If memory wait signal is aligned to EXMC_NE:

If

$$maxT_{WAIT_ASSERTION} >= T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (28\text{-}2)$$

be

$$T_{DATA_SETUP} >= (maxT_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (28\text{-}3)$$

Otherwise

$$T_{DATA_SETUP} >= 4HCLK \quad (28\text{-}4)$$

Figure 28-8. Read access timing diagram under async-wait signal assertion

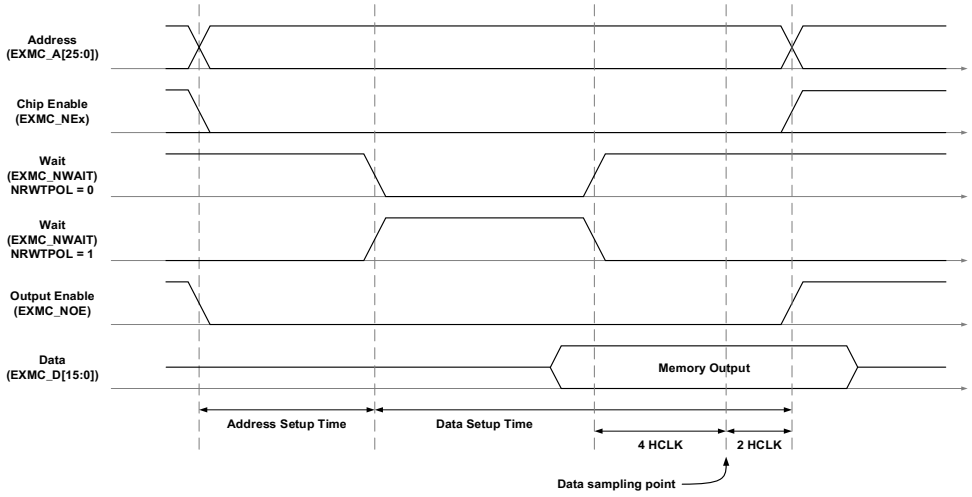
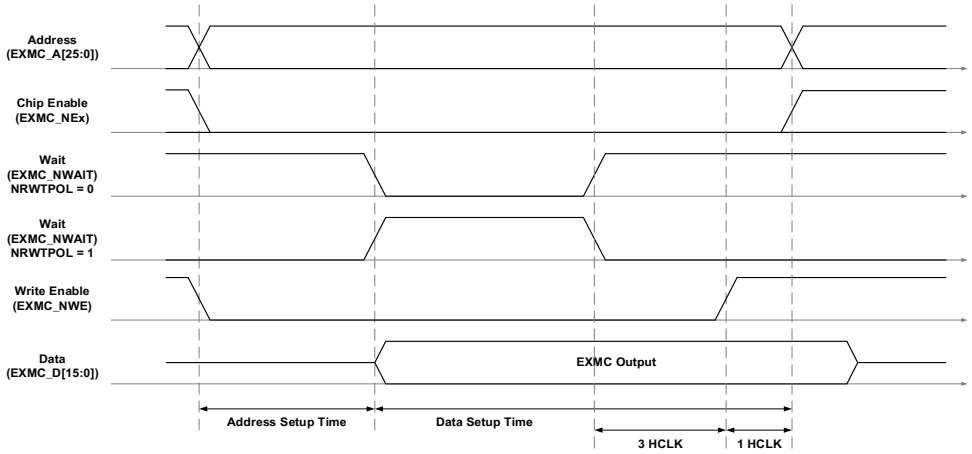


Figure 28-9. Write access timing diagram under async-wait signal assertion



Synchronous access timing diagram

The relations between memory clock (EXMC_CLK) and system clock (HCLK) clock are as follows:

$$EXMC_CLK = \frac{HCLK}{CKDIV+1} \quad (28\text{-}5)$$

CKDIV is the synchronous clock divider ratio, it is configured through the CKDIV control field in the EXMC_PNTCFGx 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 (28-6)

For NOR Flash's specification including the EXMC_NADV cycle, their relationship should be:

NOR Flash latency=DLAT+3 (28-7)

2. Data wait

Users should guarantee that EXMC_NWAIT signal matches that of the external device. This signal is configured through the EXMC_PNCTLx 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_PNCTLx 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_PNCTLx 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 28-3. EXMC bank 0 supports all transactions***.

Synchronous mux burst read timing - NOR, PSRAM (CRAM)

Figure 28-10. Read timing of synchronous multiplexed burst mode

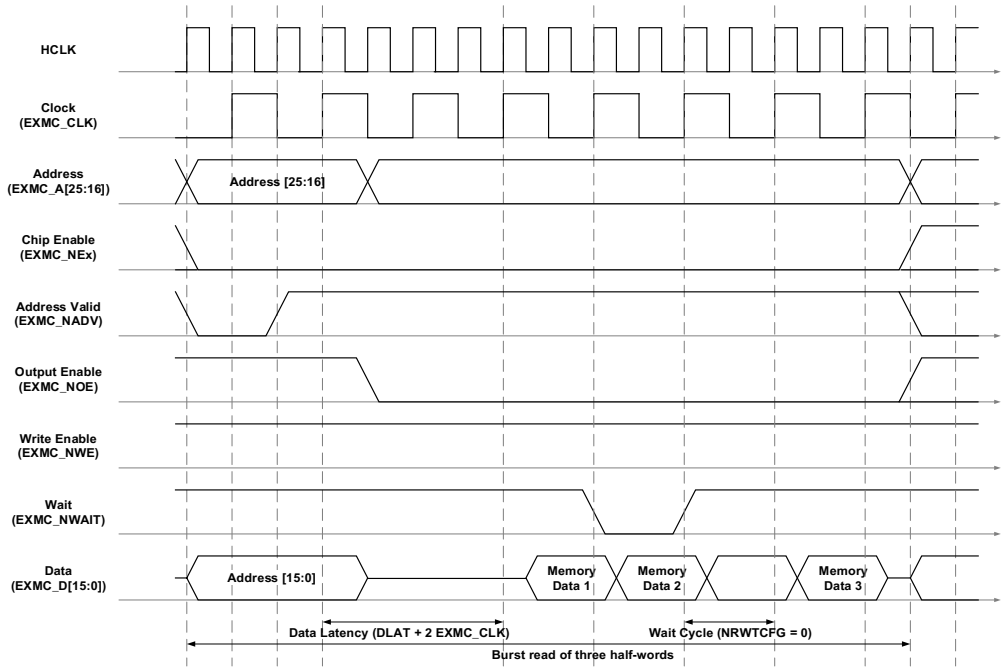


Table 28-7. Timing configurations of synchronous multiplexed read mode

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTLx		
31-20	Reserved	0x000
19	SYNCWR	No effect
18-16	CPS	0x0
15	ASYNCWAIT	0x0
14	Reserved	0x0
13	NRWTEN	Depends on memory
12	WREN	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_PNTCFGx(Read)		
31-28	Reserved	0x0
27-24	DLAT	Data latency
23-20	CKDIV	The figure above: 0x1, EXMC_CLK=2HCLK

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTLx		
19-16	BUSLAT	Time between EXMC_NE[x] rising edge to EXMC_NE[x] 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 28-11. Write timing of synchronous multiplexed burst mode

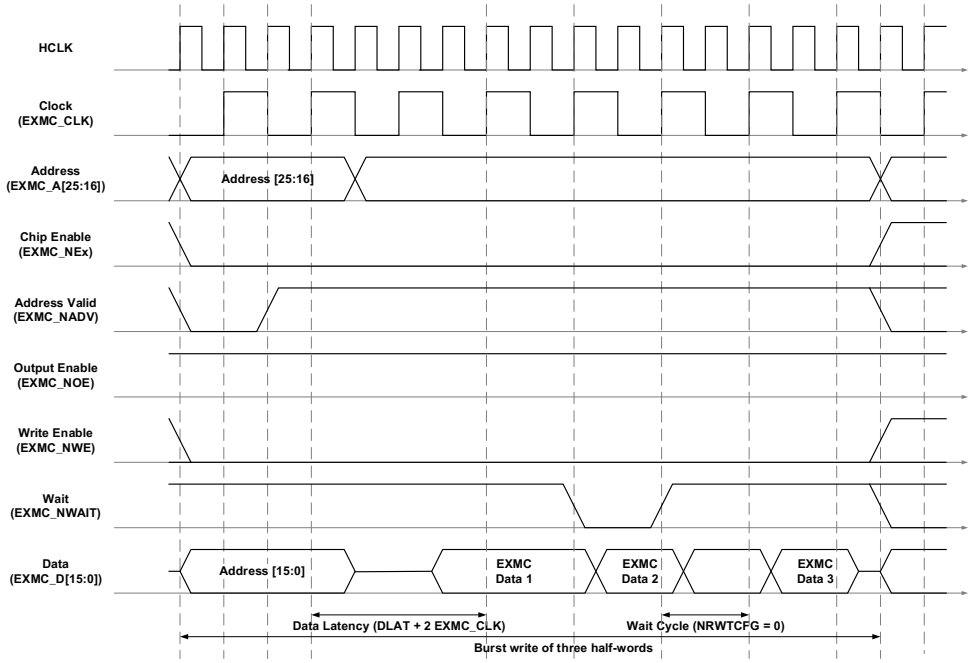


Table 28-8. Timing configurations of synchronous multiplexed write mode

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTLx		
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
9	NTWTPOL	Depends on memory
8	SBRSTEN	No effect
7	Reserved	0x1

Bit Position	Bit Name	Reference Setting Value
EXMC_PNCTLx		
6	NREN	Depends on memory
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1, Depends on users
0	NRBKEN	0x1
EXMC_PNTCFGx(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[x] rising edge to EXMC_NE[x] falling edge
15-8	DSET	No effect
7-4	AHLD	No effect
3-0	ASET	No effect

28.3.5. NAND Flash controller

EXMC has partitioned Bank1 as NAND Flash access field, Bank2 and Bank3 are reserved. Each bank has its own set of control register for access timing configuration. 8-bit and 16-bit NAND Flash are supported. An ECC hardware is provided for the NAND Flash controller to ensure the robustness of data transfer and storage.

NAND Flash interface function

Table 28-9. 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 28-10. 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	

Memory	Mode	R/W	AHB transaction size	Comments
	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
		W	32	

NAND Flash controller timing

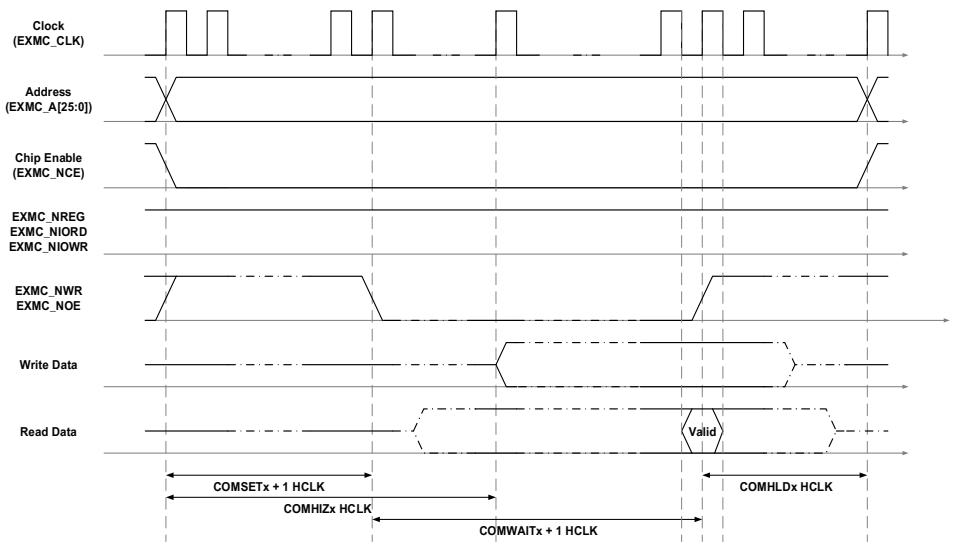
EXMC can generate the appropriate signal timing for NAND Flash device. Each bank has a corresponding register to manage and control the external memory, such as EXMC_NCTL, EXMC_NSTAT, EXMC_NCTCFG, EXMC_NATCFG and EXMC_NECC. Among these registers, EXMC_NCTCFG, EXMC_NATCFG registers contain four timing parameters individually which are configured according to user specification and features of the external memory.

Table 28-11. 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 28-12. 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 28-12. 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:

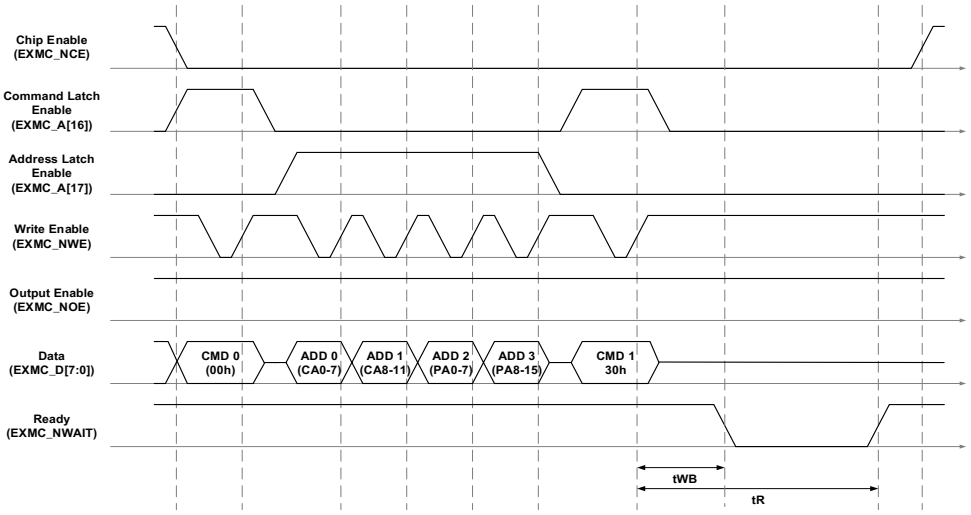
1. Configure EXMC_NCTL and EXMC_NCTCFG register. When pre-waiting is needed, EXMC_NATCFG has to be configured.
2. Send the command of NAND Flash read operation to the common space. Namely, during the valid period of EXMC_NCE and EXMC_NWE, when EXMC_CLE (EXMC_A[16]) becomes valid (high level), data on 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 28-13. 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 t_{WB} (t_{WB} is defined as the time from EXMC_NWE high to EXMC_NWAIT low).

For EXMC_NCE-sensitive NAND Flash, after the first command byte following address bytes has been entered, EXMC_NCE must remain low until EXMC_INTx goes from low to high. The ATTHLD value of attribute space can be set in EXMC_NATCFG register to meet the timing requirements of t_{WB}. 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. 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 the EXMC_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.

28.4. Registers definition

EXMC base address: 0xA000 0000

28.4.1. NOR/PSRAM controller register

PSRAM/NOR Flash control register (EXMC_PNCTLx) (x=0, 1)

Address offset: 0x00 + 8 * x, (x = 0, 1)
Reset value: 0x0000 30DX

This register has 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	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 NWAIT 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) Others: 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 register (EXMC_PNTCFGx) (x=0, 1)

Address offset: 0x04 + 8 * x, (x = 0, 1)

Reset value: 0x0FFF 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				DLAT[3:0]				CKDIV[3:0]				BUSLAT[3:0]			
				rw				rw				rw			

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

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 multiplexed read 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 mode D and 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

28.4.2. NAND Flash controller register

NAND Flash control register (EXMC_NCTL)

Address offset: 0x60

Reset value: 0x0000 0018

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ECCSZ[2:0]			ATR[3]
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATR[2:0]			CTR[3:0]				Reserved		ECCEN	NDW[1:0]		NDTP	NDBKEN	NDWTEN	Reserved
	rw			rw					rw	rw		rw	rw	rw	

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

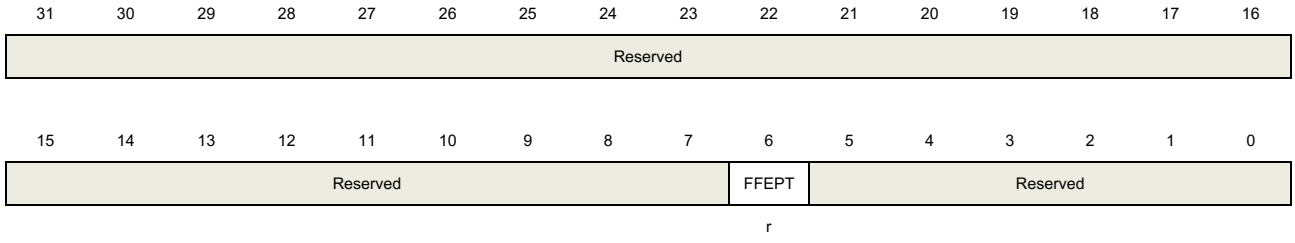
		0x1: CLE to RE delay = 2 * HCLK …… 0xF: CLE to RE delay = 16 * HCLK
8:7	Reserved	Must be kept at reset value.
6	ECCEN	ECC enable 0: Disable ECC, and reset EXMC_NECC register. 1: Enable ECC
5:4	NDW[1:0]	NAND bank memory data bus width 00: 8 bits 01: 16 bits Others: Reserved
3	NDTP	NAND bank memory type 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 register (EXMC_NSTAT)

Address offset: 0x64
Reset value: 0x0000 0042 (for bank1)

In addition to interrupt controlling bits, this register also 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.

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



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 register (EXMC_NCTCFG)

Address offset: 0x68

Reset value: 0xFCFC FCFC

These operations applicable to common memory space for NAND Flash.

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

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 register (EXMC_NATCFG)

Address offset: 0x6C
Reset value: 0xFCFC FCFC

These parameters are applicable when the timing of the write operation at the last address of the NAND flash differs from other operations. (refer to ***NAND Flash pre-wait functionality***).

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

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: 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 register (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(byte)	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]

29. Controller area network (CAN)

29.1. Overview

CAN bus (Controller Area Network) is a bus standard designed to allow microcontrollers and devices to communicate with each other without a host computer.

The CAN network interfaces of this series supports the CAN protocol 2.0A and B, ISO11898-1:2015 and BOSCH CAN-FD specifications.The CAN interface automatically handles the transmission and the reception of CAN frames. The CAN0 and CAN1 provides 28 scalable/configurable identifier filter banks.The CAN2 independently provides 14 scalable/configurable identifier filter banks. 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.

29.2. Characteristics

- Supports CAN protocols version 2.0A, B.
- Classical frames: baud rates up to 1 Mbit/s.
- Supports transmitter delay compensation.
- Supports the time-triggered communication.
- Interrupt enable and clear.
- Supports CAN FD Frame with up to 64 data bytes (ISO11898-1 and Bosch CAN FD specification V1.0).
- FD frames: baud rates up to 8 Mbit/s.
- Supports transmitter delay compensation

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 in CAN0 and CAN1.
- 14 independent and scalable/configurable identifier filter banks in CAN2.
- FIFO lock.

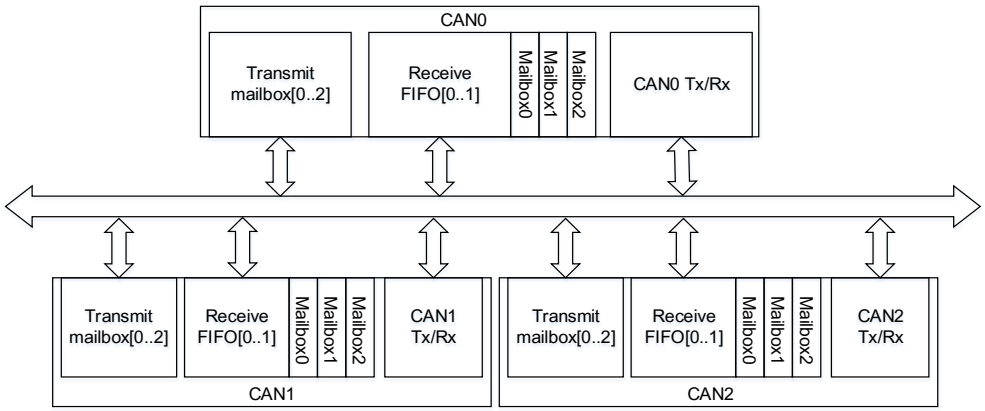
Time-triggered communication

- Disable retransmission automatically in time-triggered communication mode.
- 16-bit free timer.
- Time stamp on SOF reception.
- Time stamp sent in last two data bytes.

29.3. Function overview

Figure 29-1. CAN module block diagram shows the CAN block diagram.

Figure 29-1. CAN module block diagram



29.3.1. 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.

29.3.2. 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 logical one.

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

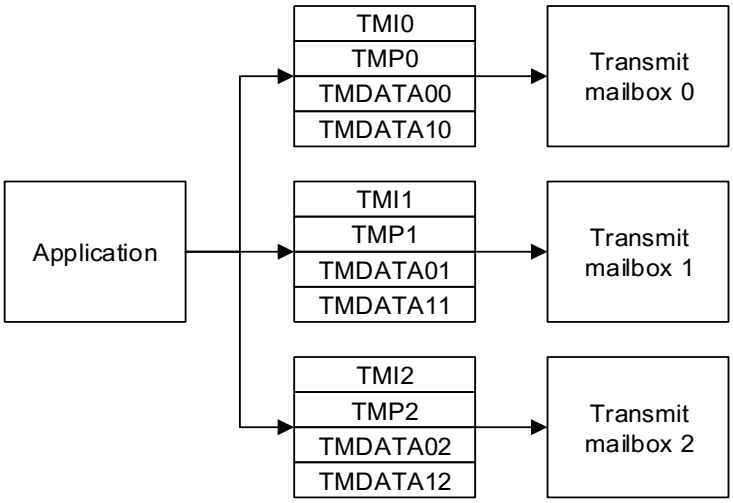
Normal communication mode is the default communication mode when the LCMOD and SCMOD bits in CAN_BT register are cleared.

29.3.3. 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 29-2. Transmission register*.

Figure 29-2. Transmission register

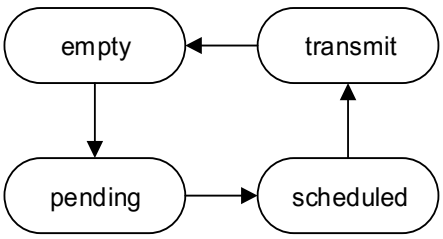


If FD frame would be transmitted, always write TMDATA00 registers when mailbox 0 is used, TMDATA01 register when mailbox 1 is used and TMDATA02 register when mailbox 2 is used until the end. For example, if application wants to transmit 64 bytes data using mailbox0, it needs to write the 64 bytes data 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 becomes idle and can be handed over to the application for reuse. the mailbox is free (**empty state**). As is shown in *Figure 29-3. State of transmit mailbox*.

Figure 29-3. State of transmit mailbox



Transmit status and error

The CAN_TSTAT register includes the transmit status and error bits: MTFx, MTFNERRx, MALx, MTEx.

- MTFx: mailbox transmit finished. Typically, MTFx is set when the frame in the transmit mailbox has been sent.

- MTFNERRx: mailbox transmit finished with no error. MTFNERRx is set when the frame in the transmit mailbox has been sent without any error.
- MALx: mailbox arbitration lost. MALx is set when the frame transmission is failed due to the arbitration lost.
- MTEx: mailbox transmit error. MTEx 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, MTFx and MTFNERRx are set if transmission is successful.

Transmission options

Abort

MSTx 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 MTFNERRx and MTFx 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.

29.3.4. Data reception

Reception register

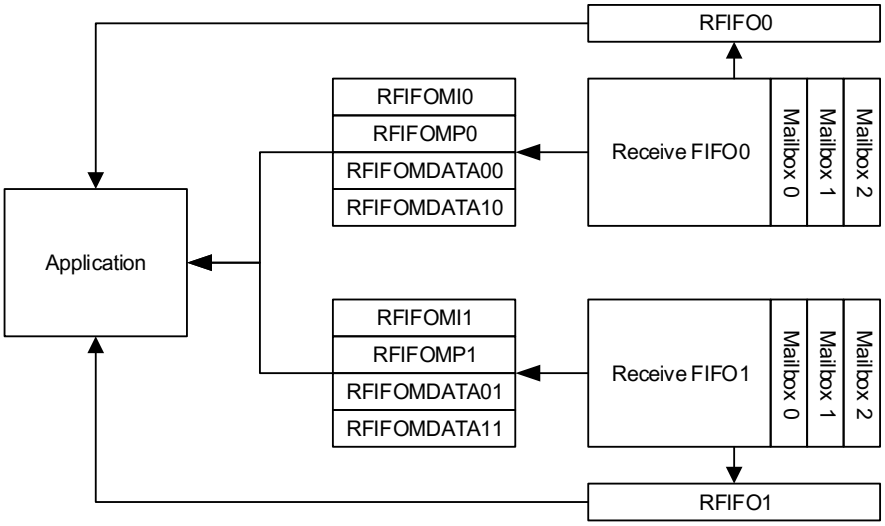
The application receives data from the CAN network through two Fifos with a depth of 3.

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 29-4. Reception register*.

Figure 29-4. 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 CAN-FD frame has been received, the data will be stored in the internal dedicated SRAM. 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

RFLx[1:0] (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 RFFx bit in CAN_RFIFOx register is set, it indicates FIFOx is full, at this time, RFLx[1:0] is 3.

When a new frame arrives after the FIFO has held three frames, the RFOx bit in CAN_RFIFOx register will be set, and it indicates FIFOx is overfull. 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. when another new frames arrives after the FIFO status is overfull, the RFOVRNx bit in CAN_RFIFOx register will be set, it indicates FIFOx is overrun, and the data in FIFO is incorrect.

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.

29.3.5. 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.

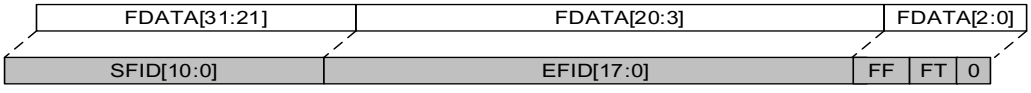
Scale

CAN0 and CAN1 has the filter consists of 28 banks: bank0 to bank27. CAN2 has independent the filter consists of 14 banks: bank0 to bank13. 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 29-5. 32-bit filter*.

Figure 29-5. 32-bit filter



16-bit: SFID [10:0], FT, FF and EFID[17:15] bits. As is shown in *Figure 29-6. 16-bit filter*.

Figure 29-6. 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 29-7. 32-bit mask mode filter*.

Figure 29-7. 32-bit mask mode filter

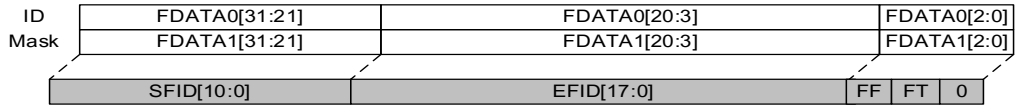
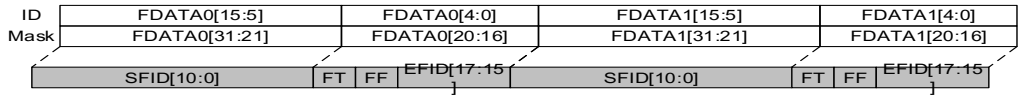


Figure 29-8. 16-bit mask mode filter



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 29-9. 32-bit list mode filter*.

Figure 29-9. 32-bit list mode filter

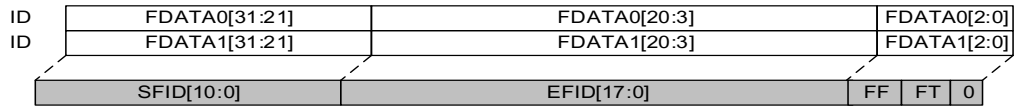
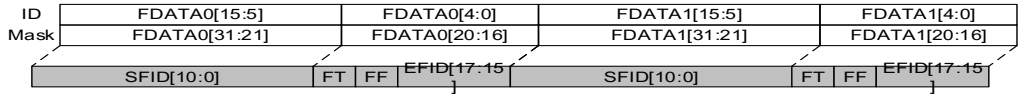


Figure 29-10. 16-bit list mode filter



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 29-1. 32-bit filter number*.

Table 29-1. 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. Config by the CAN_FW register.

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. If a frame passes through the Filter unit with filter Number 10 (Filter Number=10) in FIFO0, then the filter index of that frame is 10. At this point, the value of FI in CAN_RFIFOMPx is 10.

The example about filtering index is shown in *Table 29-2. Filtering index*.

Table 29-2. 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-		7		F6DATA1[15:0]-16bits-ID		7

Filter bank	FIFO0	Active	Filter nunber	Filter bank	FIFO1	Active	Filter nunber
	ID						
	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-ID	Yes	13	11	F11DATA0[15:0]-16bits-ID	No	11
	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:

- 32-bits mode is higher than 16-bits mode.
- List mode is higher than mask mode.
- Smaller filter number has the higher priority.

29.3.6. 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.

29.3.7. 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 MTFx bit of register CAN_TSTAT is hardware set to 1, and the sending status information can be obtained via MTFNERRx, MALx, and MTEx.

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:

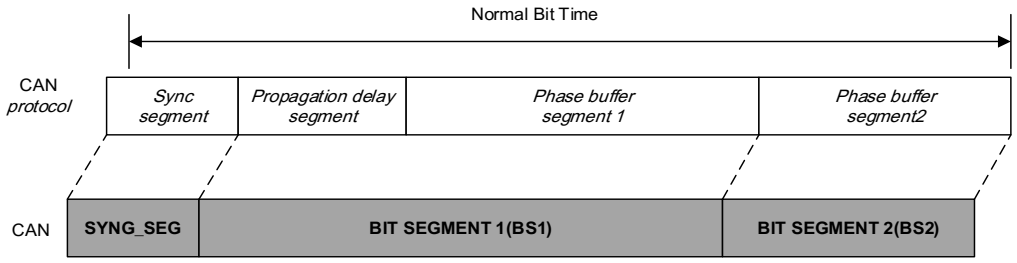
Synchronization segment (SYNC_SEG): a bit change is expected to occur within this time segment. It has a fixed length of one time quantum ($1 \times t_q$).

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. Its duration is programmable from 1 to 16 time quanta but 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. Its duration is programmable from 1 to 8 time quanta but it may also be automatically shortened to compensate for negative phase drifts.

The bit time is shown as in the ***Figure 29-11. The bit time***.

Figure 29-11. 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. It is configured by SJW[4:0] bits in CAN_BT register for nomainal bit time, or configured by DSJW[2:0] bits in CAN_DBT register for data bit time.

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 clock of the CAN derives from the APB2 bus. The CAN calculates its baud rate as follow:

$$BaudRate=\frac{1}{Normal\ Bit\ Time} \quad (29\text{-}1)$$

$$Normal\ Bit\ Time=t_{SYNC_SEG}+t_{BS1}+t_{BS2} \quad (29\text{-}2)$$

with:

$$t_{SYNC_SEG}=1\times t_q \quad (29\text{-}3)$$

$$t_{BS1}=(1+BT.BS1)\times t_q \quad (29\text{-}4)$$

$$t_{BS2}=(1+BT.BS2)\times t_q \quad (29\text{-}5)$$

$$t_q=(1+BT.BAUDPSC)\times t_{PCLK2} \quad (29\text{-}6)$$

29.3.8. 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.

In CAN-FD frame structure, 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 data phase shall start at the sample point when the BRS bit is detected as recessive and shall end at the first sample point of the CRC delimiter, or when the CAN node detects an error condition that leads to the initiation of an error frame. The bit rate of data can be switched in the period from BRS bit to CRC delimiter. The data bit rate can be switched at the sample points of the BRS bit and the CRC delimiter bit. (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.

29.3.9. 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.

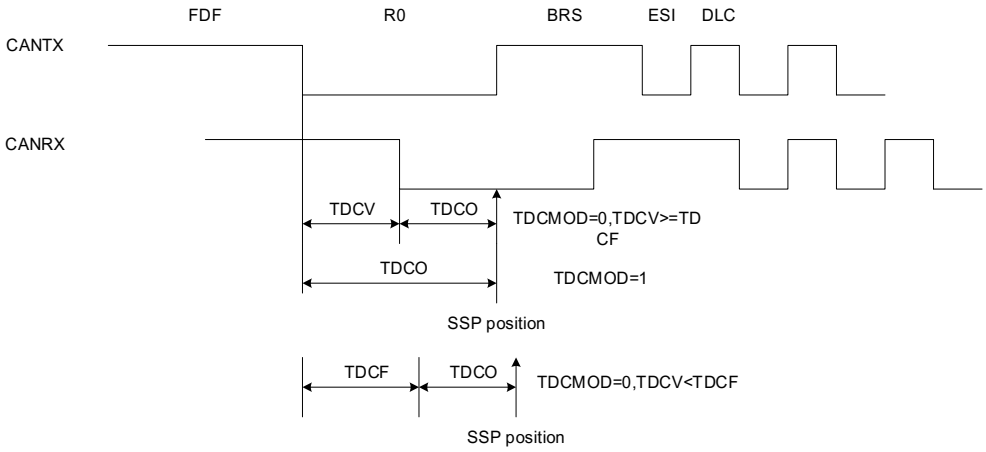
The transmitter delay compensation is enabled by setting TDCEN bit in CAN_FDCTL register.

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[7:0] 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[6:0] 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[7:0] bits of CAN_FDTDC register could be used to avoid too small TDCV[6:0] bits. So the value of SSP delay is TDCO[7:0] bits add TDCV[6:0] bits if TDCV[6:0] is larger than TDCF[7:0], or else the value of SSP delay is TDCO[7:0] bits add TDCF[7:0] bits.

The SSP delay can not exceed 3 data bit time.

Figure 29-12. Transmitter Delay Measurement



29.3.10. Error flags

The state of CAN bus can be reflected by Transmit Error Counter (TECNT[7:0]) and Receive Error Counter (RECNT[7:0]) 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.

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[7:0] 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 Bus-Off 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.

29.3.11. 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.

Receive FIFO0 interrupt

The Rx FIFO0 interrupt can be generated by the following conditions:

- Rx FIFO0 not empty: RFL0[1:0] 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 overfull: 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[1:0] 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 overfull: RFO1 bit in the CAN_RFIFO1 register is set and RFOIE1 in

CAN_INTEN register is set.

Error and working mode change (EWMC) 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 29-3. CAN Event / Interrupt flags*.

Table 29-3. 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 (RFL1[1:0])		RFNEIE1	
	Rx FIFO1 full (RFF1)		RFFIE1	
	Rx FIFO1 overfull (RFO1)		RFOIE1	
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	

29.4. CAN registers

CAN0 base address: 0x4001 1C00
CAN1 base address: 0x4001 2000
CAN2 base address: 0x4001 7400

29.4.1. 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_CTL0 register is set, this bit defines the CAN controller is in debug freezing mode or normal working mode. If the CANx_HOLD in DBG_CTL0 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

29.4.2. 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

29.4.3. 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

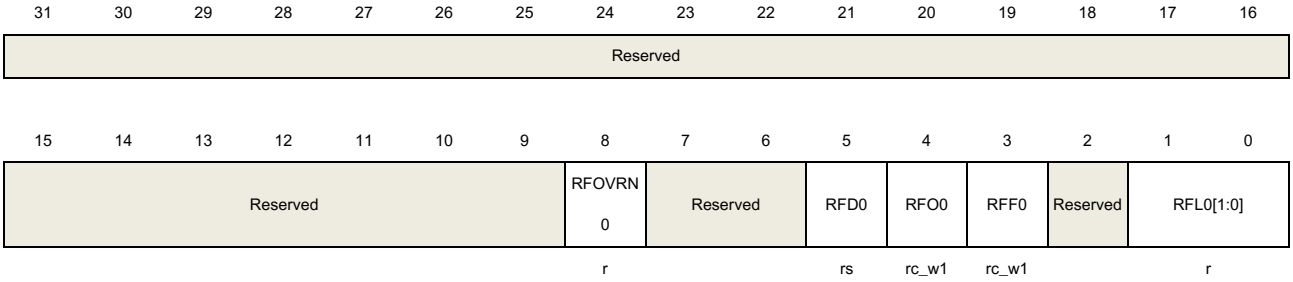
1: Mailbox 0 transmit finished

29.4.4. Receive message FIFO0 register (CAN_RFIFO0)

Address offset: 0x0C

Reset value: 0x0000 0000

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



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 writting 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 writting 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.

29.4.5. 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.

29.4.6. 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

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

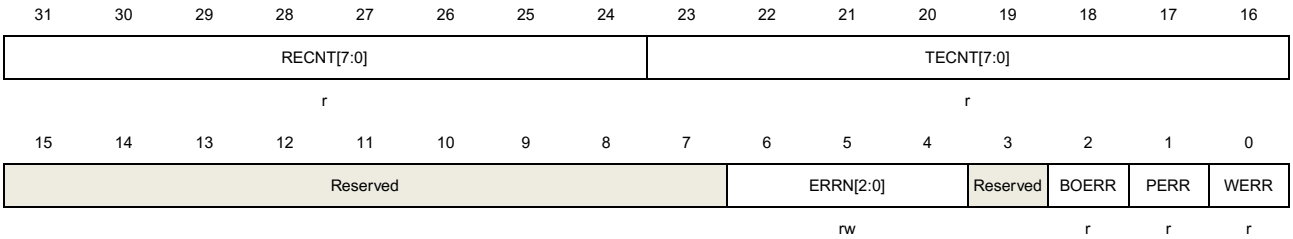
		0: Rx FIFO1 overfull interrupt disabled 1: Rx FIFO1 overfull interrupt enabled
5	RFFIE1	Rx FIFO1 full interrupt enable 0: Rx FIFO1 full interrupt disabled 1: Rx FIFO1 full interrupt enabled
4	RFNEIE1	Rx FIFO1 not empty interrupt enable 0: Rx FIFO1 not empty interrupt disabled 1: Rx FIFO1 not empty interrupt 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

29.4.7. Error register (CAN_ERR)

Address offset: 0x18

Reset value: 0x0000 0000

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



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

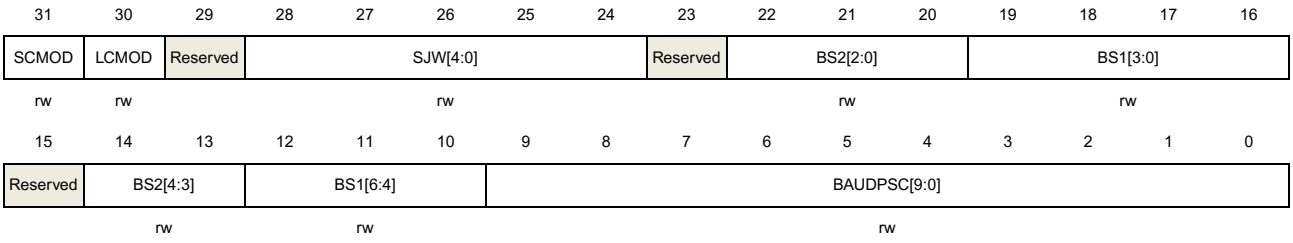
		These bits indicate the error status of bit transformation. They are updated by hardware. When the bit transformation is successful, they are equal to 0. 000: No error 001: Stuff error 010: Form error 011: Acknowledgment error 100: Bit recessive error 101: Bit dominant error 110: CRC error 111: Set by software
3	Reserved	Must be kept at reset value.
2	BOERR	Bus-Off error Whenever the TECNT is greater than 255, 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.

29.4.8. Bit timing register (CAN_BT)

Address offset: 0x1C

Reset value: 0x0123 0000

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



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

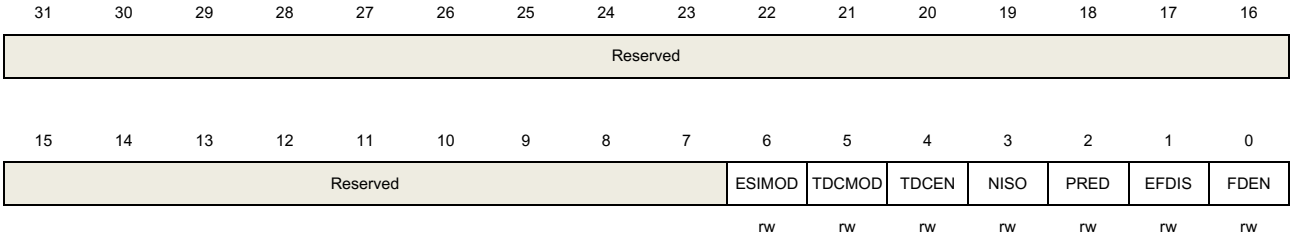
29	Reserved	Must be kept at reset value.
28:24	SJW[4:0]	Resynchronization jump width Resynchronization jump width time quantum = SJW[4:0]+1
23	Reserved	Must be kept at reset value.
22:20	BS2[2:0]	Bit segment 2 Bit segment 2 time quantum = BS2[4:0]+1
19:16	BS1[3:0]	Bit segment 1 Bit segment 1 time quantum = 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 This bit can be configured only when FDEN=1 in CAN_FDCTL register.
12:10	BS1[6:4]	Bits 6:4 of BS1 See bits 19:16 of CAN_BT This bit can be configured only when FDEN=1 in CAN_FDCTL register.
9:0	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler

29.4.9. FD control register (CAN_FDCTL)

Address offset: 0x20

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	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:

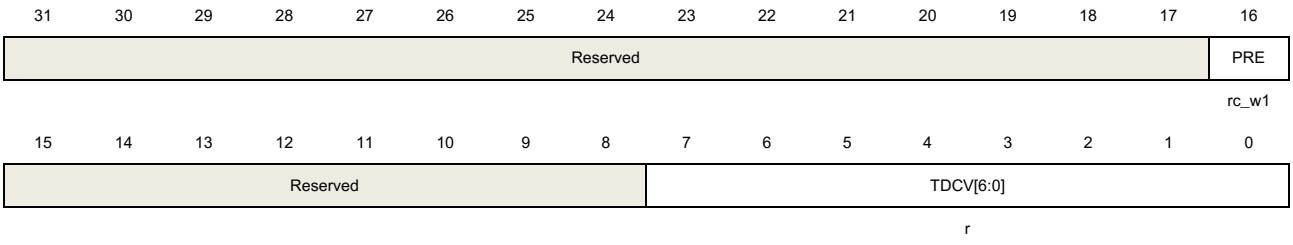
		Displays the node error state.
5	TDCMOD	Transmitter delay compensation mode 0: Measurement and offset 1: Only offset
4	TDCEN	Transmitter delay compensation enable 0: Transmitter delay compensation is disabled 1: Transmitter delay compensation is enabled
3	NISO	ISO/Bosch 0: ISO 1: Bosch
2	PRED	Protocol 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

29.4.10. 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 writting 1.
15:8	Reserved	Must be kept at reset value.

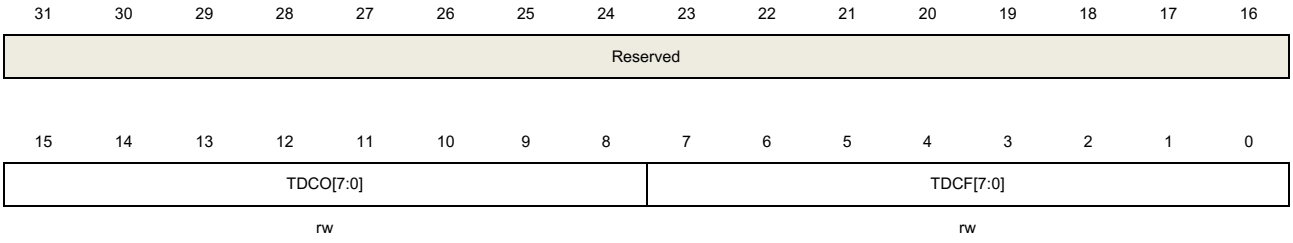
7:0	TDCV[7:0]	Transmitter delay compensation value These bits are set by hardware to display transmitter delay compensation value.

29.4.11. FD transmitter delay compensation register (CAN_FDTDC)

Address offset: 0x28

Reset value: 0x0000 0000

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



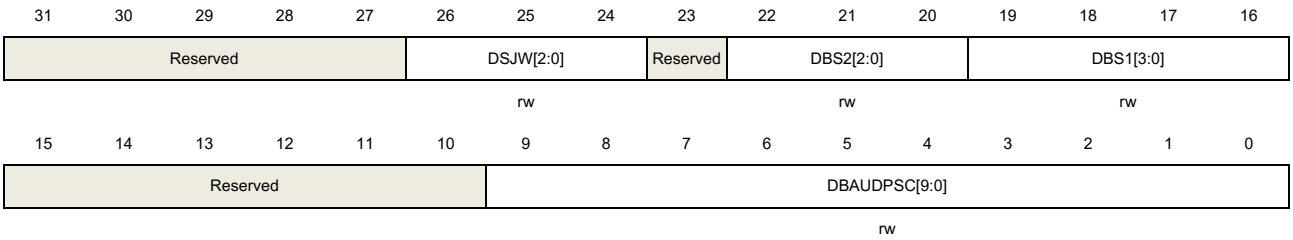
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	TDCO[7:0]	Transmitter delay compensation offset These bits are set to the transmitter delay compensation offset value which defines the distance between the measured delay from CANTX to CANRX and the second sample point.
7:0	TDCF[7:0]	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[7:0] is configured to a value greater than TDCV.

29.4.12. 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

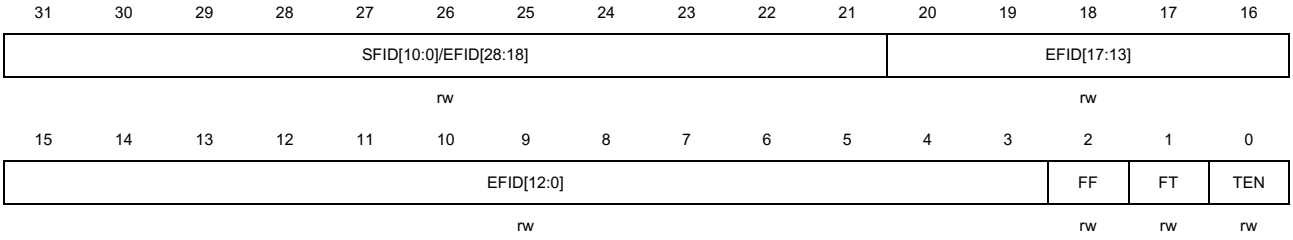
31:27	Reserved	Must be kept at reset value.
26:24	DSJW[2:0]	Resynchronization jump width Resynchronization jump width time quantum = DSJW[2:0]+1
23	Reserved	Must be kept at reset value.
22:20	DBS2[2:0]	Bit segment 2 Bit segment 2 time quantum = DBS2[2:0]+1
19:16	DBS1[3:0]	Bit segment 1 Bit segment 1 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

29.4.13. 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).



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
20:3	EFID[17:0]	The frame identifier EFID[17:0]: 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

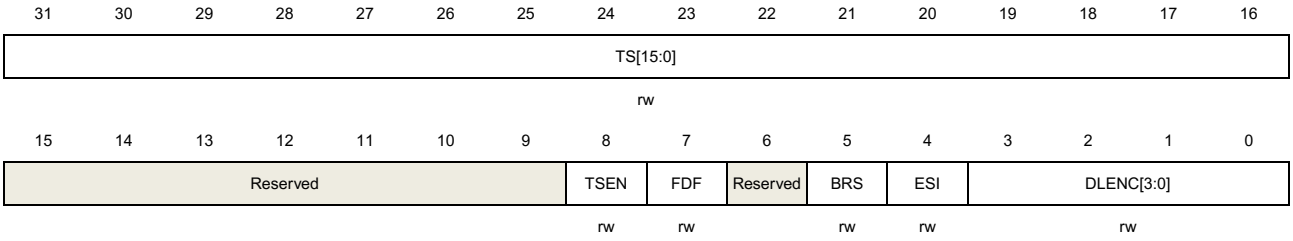
1: Transmit enabled

29.4.14. 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).



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 CAN_TMDATA1x. 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 data bytes in a frame

29.4.15. 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).



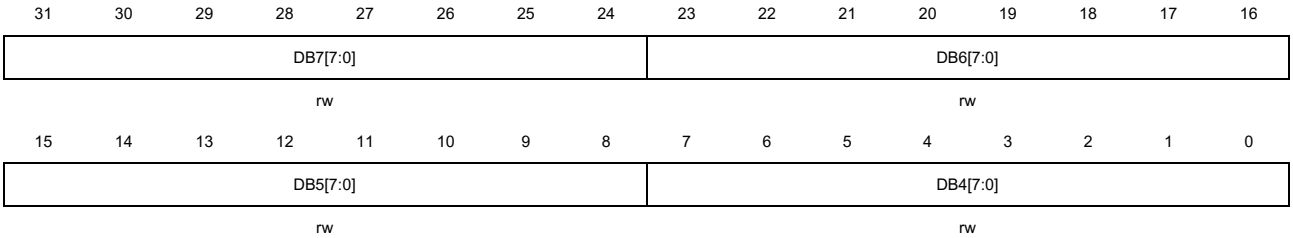
Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2
15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

29.4.16. 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

29.4.17. Receive 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.

29.4.18. Receive 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 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

29.4.19. Receive 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

29.4.20. Receive FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x = 0,1)

Address offset: 0x1BC + 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
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

29.4.21. Filter control register (CAN_FCTL)

Address offset: 0x200

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 0, no bank used for CAN0. When set 28, no bank used for CAN1. These bits are only used in CAN0 and CAN1.

7:1	Reserved	Must be kept at reset value.
0	FLD	Filter lock disable 0: Filter lock enabled 1: Filter lock disabled

29.4.22. Filter mode configuration register (CAN_FMCFG)

Address offset: 0x204

Reset value: 0x0000 0000

The filter mode configuration register inCAN0 and CAN1:

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	FMODx	Filter mode 0: Filter x with mask mode 1: Filter x with list mode

The filter mode configuration register inCAN2:

31	30	29	28	27	26	25	24	23	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		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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	FMODx	Filter mode 0: Filter x with mask mode 1: Filter x with list mode

29.4.23. Filter scale configuration register (CAN_FSCFG)

Address offset: 0x20C
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.

The filter scale configuration register in CAN0 and CAN1:

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	FSx	Filter scale 0: Filter x with 16-bit scale 1: Filter x with 32-bit scale

The filter scale configuration register in CAN2:

31	30	29	28	27	26	25	24	23	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		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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	FSx	Filter scale 0: Filter x with 16-bit scale 1: Filter x with 32-bit scale

29.4.24. Filter associated FIFO register (CAN_FAFIFO)

Address offset: 0x214
Reset value: 0x0000 0000
The filter associated FIFO register in CAN0 and CAN1.

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	FAFx	Filter associated FIFO 0: Filter x associated with FIFO0 1: Filter x associated with FIFO1

The filter associated FIFO register in CAN2:

31	30	29	28	27	26	25	24	23	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		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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	FAFx	Filter associated FIFO 0: Filter x associated with FIFO0 1: Filter x associated with FIFO1

29.4.25. Filter working register (CAN_FW)

Address offset: 0x21C

Reset value: 0x0000 0000

The filter working register in CAN0 and CAN1.

This register has 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	FWx	Filter working 0: Filter x working disabled 1: Filter x working enabled

The filter working register in CAN2.

31	30	29	28	27	26	25	24	23	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		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

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13:0	FWx	Filter working 0: Filter x working disabled 1: Filter x working enabled

29.4.26. Filter x data y register (CAN_FxDATAy) (x = 0…27, y = 0,1)

Address offset: 0x240 + 8 * x + 4 * y, (In CAN2: x = 0..13, 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	FDx	Filter data Mask mode 0: Mask match disable 1: Mask match enable List mode 0: List identifier bit is 0 1: List identifier bit is 1

30. Secure digital input/output interface (SDIO)

30.1. Introduction

The secure digital input/output interface (SDIO) defines the SD, SD I/O, MMC and CE-ATA card host interface, which provides command/data transfer between the APB2 system bus and SD memory cards, SD I/O cards, Multimedia Card (MMC) and CE-ATA devices.

The supported SD memory card and SD I/O card system specifications are defined in the SD card Association website at **www.sdcard.org**.

The supported Multimedia Card system specifications are defined through the Multimedia Card Association website at **www.jedec.org**, published by the JEDEC SOLID STATE TECHNOLOGY ASSOCIATION.

The supported CE-ATA system specifications are defined through the CE-ATA workgroup website at **www.ce-ata.org**.

30.2. Main features

The SDIO features include the following:

- **MMC:** Full support for Multimedia Card System Specification Version 4.2(and previous versions) Card and three different data bus modes: 1-bit (default), 4-bit and 8-bit.
- **SD Card:** Full support for *SD Memory Card Specifications Version 2.0.*
- **SD I/O:** Full support for *SD I/O Card Specification Version 2.0* card and two different data bus modes: 1-bit (default) and 4-bit.
- **CE-ATA:** Full compliance with *CE-ATA digital protocol Version 1.1.*
- 48MHz data transfer frequency and 8-bit data transfer mode.
- Interrupt and DMA request to processor.
- Completion Signal enables and disable feature (CE-ATA).

Note: SDIO supports only one SD, SD I/O, MMC4.2 card or CE-ATA device at any one time and a stack of MMC4.1 or previous.

30.3. SDIO bus topology

After a power-on reset, the host must initialize the card by a special message-based bus protocol.

Each message is represented by one of the following tokens:

Command: a command is a token which starts an operation. A command is sent from the host to a card. A command is transferred serially on the CMD line.

Response: a response is a token which is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Data: data can be transferred from the card to the host or vice versa. Data is transferred via the data lines. The number of data lines used for the data transfer can be 1(DAT0), 4(DAT0-DAT3) or 8(DAT0-DAT7).

The structure of commands, responses and data blocks is described in ***Card functional description***. One data transfer is a bus operation.

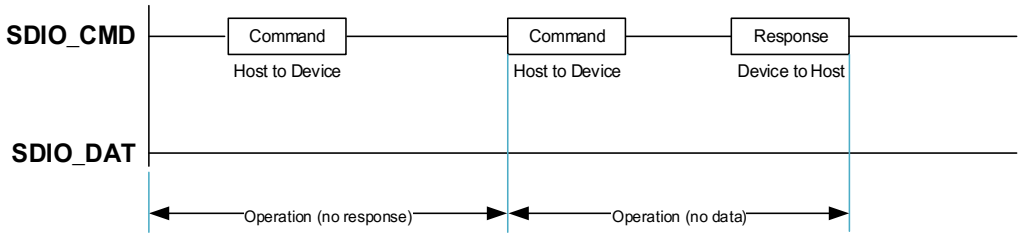
There are different types of operations. Addressed operations always contain a command and a response token. In addition, some operations have a data token; the others transfer their information directly within the command or response structure. In this case no data token is present in an operation. The bits on the DAT0-DAT7 and CMD lines are transferred synchronous to the host clock.

Two types of data transfer commands are defined:

- Stream commands: These commands initiate a continuous data stream; they are terminated only when a stop command follows on the CMD line. This mode reduces the command overhead to an absolute minimum (only MMC supports).
- Block-oriented commands: These commands send a data block successfully by CRC bits. Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the CMD line similarly to the sequential read.

The basic transaction on the bus is the command/response transaction (refer to ***Figure 30-1. SDIO "no response" and "no data" operations***). This type of bus transaction transfers their information directly within the command or response structure. In addition, some operations have a data token. Data transfers to/from the Card/Device are done in blocks.

Figure 30-1. SDIO "no response" and "no data" operations



Note that the Multiple Block operation mode is faster than Single Block operation. A multiple block transmission is terminated when a stop command follows on the CMD line. Data transfer can be configured by the host to use single or multiple data lines. ***Figure 30-2. SDIO multiple blocks read operation*** is the multiple blocks read operation and ***Figure 30-3. SDIO multiple blocks write operation*** is the multiple blocks write operation. The block write operation uses a simple busy signal of the write operation duration on the data (DAT0) line. CE-ATA device

has an optional busy before it is ready to receive the data.

Figure 30-2. SDIO multiple blocks read operation

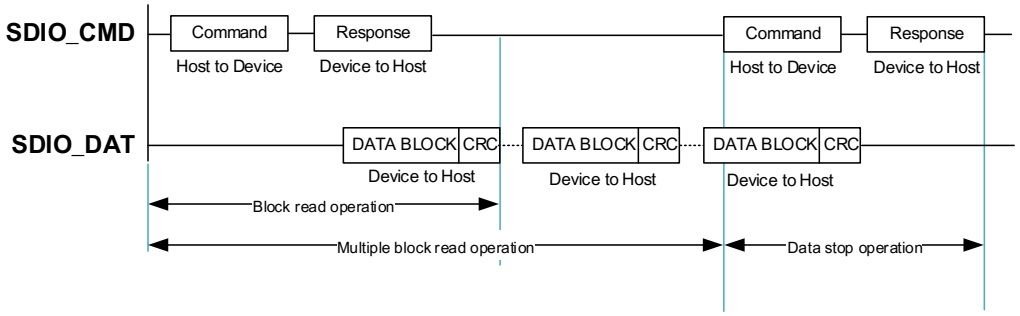
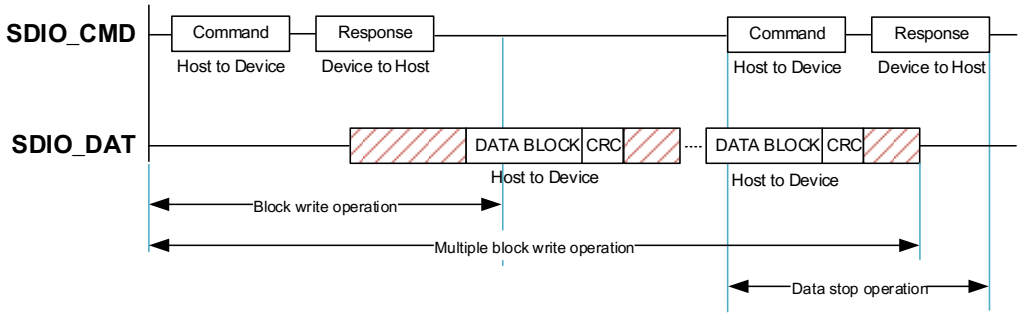


Figure 30-3. SDIO multiple blocks write operation



Data transfers to/from SD memory cards, SD I/O cards (both IO only card and combo card) and CE-ATA device are done in data blocks. Data transfers to/from MMC are done in data blocks or streams. ***Figure 30-4. SDIO sequential read operation*** and ***Figure 30-5. SDIO sequential write operation*** are the stream read and write operation.

Figure 30-4. SDIO sequential read operation

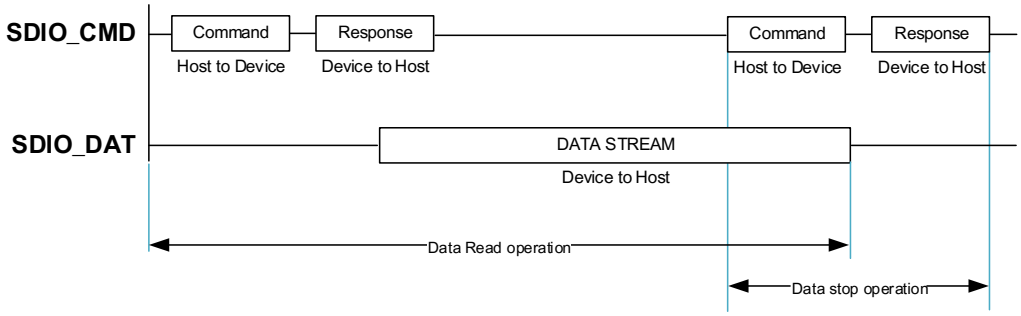
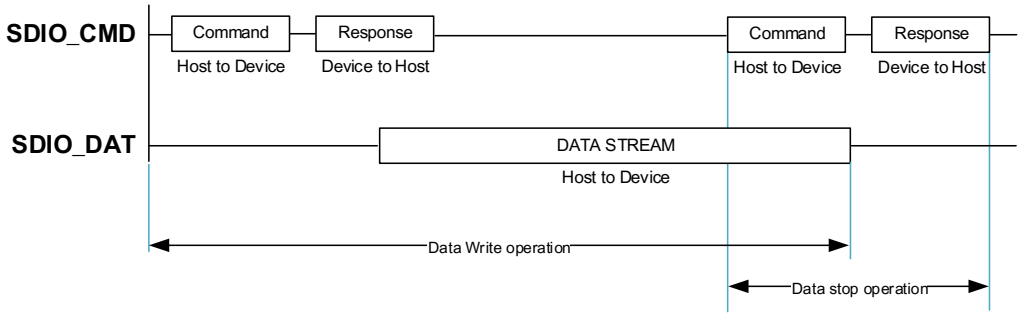


Figure 30-5. SDIO sequential write operation

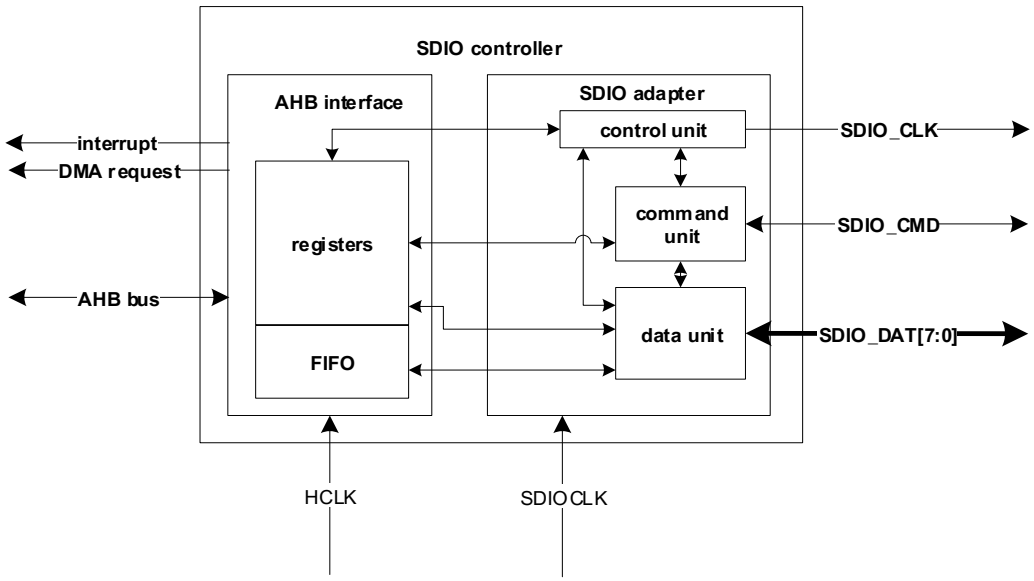


30.4. SDIO functional description

The *Figure 30-6. SDIO block diagram* shows the SDIO structure. There have two main parts:

- The SDIO adapter block consists of control unit which manage clock, command unit which manage command transfer, data unit which manage data transfer.
- The APB interface block contains access registers by APB2 bus, contains FIFO unit which is data FIFO used for data transfer, and generates interrupt and DMA request signals.

Figure 30-6. SDIO block diagram



30.4.1. SDIO adapter

The SDIO adapter contains control unit, command unit and data unit, and generates signals to cards. The signals is descript bellow:

SDIO_CLK: The SDIO_CLK is the clock provided to the card. Each cycle of this signal directs a one bit transfer on the command line (SDIO_CMD) and on all the data lines (SDIO_DAT). The SDIO_CLK frequency can vary between 0 MHz and 20 MHz for a Multimedia Card V3.31, between 0 and 48 MHz for a Multimedia Card V4.2, or between 0 and 25 MHz for an SD/SD I/O card.

The SDIO uses two clock signals: SDIO adapter clock (SDIOCLK <= 48MHz) and APB2 bus clock (CK_APB2), the frequency of PCLK2 must be no less than the frequency of SDIO_CK.

SDIO_CMD: This signal is a bidirectional command channel used for card initialization and transfer of commands. Commands are sent from the SDIO controller to the card and responses are sent from the card to the host. The CMD signal has two operation modes: open-drain for initialization (only for MMC3.31 or previous), and push-pull for command

transfer (SD/SD I/O card MMC4.2 use push-pull drivers also for initialization).

SDIO_DAT[7:0]: These are bidirectional data channels. The DAT signals operate in push-pull mode. Only the card or the host is driving these signals at a time. By default, after power up or reset, only DAT0 is used for data transfer. A wider data bus can be configured for data transfer, using either DAT0-DAT3 or DAT0-DAT7 (just for MMC4.2), by the SDIO controller. The SDIO includes internal pull-ups for data lines DAT1-DAT7. Right after entering to the 4-bit mode the card disconnects the internal pull-ups of lines DAT1 and DAT2 (DAT3 internal pull-up is left connected due to the SPI mode CS usage). Correspondingly right after entering to the 8-bit mode the card disconnects the internal pull-ups of lines DAT1, DAT2 and DAT4-DAT7.

Table 30-1. SDIO I/O definitions

Pin function	Signal	Direction	Description
SDIO_CLK	SDIO_CLK	O	SD/SD I/O /MMC clock
SDIO_CMD	SDIO_CMD	I/O	Command input/output
SDIO_D[7:0]	SDIO_DAT[7:0]	I/O	Data input/output for data lines DAT[7:0]

The SDIO adapter is an interface to SD, SD I/O, MMC and CE-ATA. It consists of three subunits:

Control unit

The control unit contains the power management functions and the clock management functions for the memory card clock. The power management is controlled by SDIO_PWRCTL register which implements power off or power on. The power saving mode configured by setting CLKPWRSAV bit in SDIO_CLKCTL register, which implements close the SDIO_CLK when the bus is idle. The clock management generates SDIO_CLK to card. The SDIO_CLK is generated by a divider of SDIOCLK when CLKBYP bit in SDIO_CLKCTL register is 0, or directly SDIOCLK when CLKBYP bit in SDIO_CLKCTL register is 1.

The Hardware clock control is enabled by setting HWCLKEN in SDIO_CLKCTL register. This functionality is used to avoid FIFO underrun and overrun errors by hardware control the SDIO_CLK on/off depending on the system bus is very busy or not. When the FIFO cannot receive or transmit data, the host will stop the SDIO_CLK and freeze SDIO state machines to avoid the corresponded error. Only state machines are frozen, the APB2 interface is still alive. So, the FIFO can access by APB2 bus.

Command unit

The command unit implements command transfer to the card. The data transfer flow is controlled by Command State Machine (CSM). After a write operation to SDIO_CMDCTL register and CSMEN in SDIO_CMDCTL register is 1, the command transfer starts. It firstly sends a command to the card. The command contains 48 bits send by SDIO_CMD signal which sends 1 bit to card at one SDIO_CLK. The 48 bits command contains 1 bit Start bit, 1 bit Transmission bit, 6 bits command index defined by CMDIDX bits in SDIO_CMDCTL register,

32 bits argument defined in SDIO_CMDAGMT register, 7 bits CRC, and 1 bit end bit. Then receive response from the card if CMDRESP in SDIO_CMDCTL register is not 0b00/0b10. There are short response which have 48 bits or long response which have 136 bits. The response stores in SDIO_RESP0 - SDIO_RESP3 registers. The command unit also generates the command status flags defined in SDIO_STAT register.

Command state machine

CS_Idle	After reset, ready to send command.	
1.CSM enabled and WAITDEND enabled	->	CS_Pend
2.CSM enabled and WAITDEND disabled	->	CS_Send
3.CSM disabled	->	CS_Idle
Note: The state machine remains in the Idle state for at least eight SDIO_CLK periods to meet the N_{CC} and N_{RC} timing constraints. N_{CC} is the minimum delay between two host commands, and N_{RC} is the minimum delay between the host command and the response.		

CS_Pend	Waits for the end of data transfer.	
1.The data transfer complete	->	CS_Send
2.CSM disabled	->	CS_Idle

CS_Send	Sending the command.	
1.The command transmitted has response	->	CS_Wait
2.The command transmitted doesn't have response	->	CS_Idle
3.CSM disabled	->	CS_Idle

CS_Wait	Wait for the start bit of the response.	
1.Receive the response(detected the start bit)	->	CS_Receive
2.Timeout is reached without receiving the response	->	CS_Idle
3.CSM disabled	->	CS_Idle
Note: The command timeout has a fixed value of 64 SDIO_CLK clock periods.		

CS_Receive	Receive the response and check the CRC.	
1.Response Received in CE-ATA mode and interrupt disabled and wait for CE-ATA Command Completion signal enabled	->	CS_Waitcompl
2.Response Received in CE-ATA mode and interrupt disabled and wait for CE-ATA Command Completion signal disabled	->	CS_Pend
3.CSM disabled	->	CS_Idle
4.Response received	->	CS_Idle
5.Command CRC failed	->	CS_Idle

CS_Waitcompl	Wait for the Command Completion signal.

1.CE-ATA Command Completion signal received	->	CS_Idle
2.CSM disabled	->	CS_Idle
3.Command CRC failed	->	CS_Idle

Data unit

The data unit performs data transfers to and from cards. The data transfer uses SDIO_DAT[7:0] signals when 8-bits data width (BUSMODE bits in SDIO_CLKCTL register is 0b10), use SDIO_DAT[3:0] signals when 4-bits data width (BUSMODE bits in SDIO_CLKCTL register is 0b01), or SDIO_DAT[0] signal when 1-bit data width (BUSMODE bits in SDIO_CLKCTL register is 0b00). The data transfer flow is controlled by Date State Machine (DSM). After a write operation to SDIO_DATACTL register and DATAEN in SDIO_DATACTL register is 1, the data transfer starts. It sends data to card when DATADIR in SDIO_DATACTL register is 0, or receive data from card when DATADIR in SDIO_DATACTL register is 1. The data unit also generates the data status flags defined in SDIO_STAT register.

Data state machine

DS_Idle	The data unit is inactive, waiting for send and receive.	
1.DSM enabled and data transfer direction is from host to card	->	DS_WaitS
2.DSM enabled and data transfer direction is from card to host	->	DS_WaitR
3.DSM enabled and Read Wait Started and SD I/O mode enabled	->	DS_Readwait

DS_WaitS	Wait until the data FIFO empty flag is deasserted or data transfer ended.	
1.Data transfer ended	->	DS_Idle
2.DSM disabled	->	DS_Idle
3.Data FIFO empty flag is deasserted	->	DS_Send

DS_Send	Transmit data to the card.	
1.Data block transmitted	->	DS_Busy
2.DSM disabled	->	DS_Idle
3.Data FIFO underrun error occurs	->	DS_Idle
4. Internal CRC error	->	DS_Idle

DS_Busy	Waits for the CRC status flag.	
1.Receive a positive CRC status	->	DS_WaitS
2.Receive a negative CRC status	->	DS_Idle
3.DSM disabled	->	DS_Idle
4.Timeout occurs	->	DS_Idle
Note: The command timeout programmed in the data timer register (SDIO_DATATO).		

DS_WaitR	Wait for the start bit of the receive data.	
1.Data receive ended	->	DS_Idle
2.DSM disabled	->	DS_Idle
3.Data timeout reached	->	DS_Idle
4.Receives a start bit before timeout	->	DS_Receive
Note: The command timeout programmed in the data timer register (SDIO_DATATO).		

DS_Receive	Receive data from the card and write it to the data FIFO.	
1.Data block received	->	DS_WaitR
2.Data transfer ended	->	DS_WaitR
3.Data FIFO overrun error occurs	->	DS_Idle
4.Data received and Read Wait Started and SD I/O mode enabled	->	DS_Readwait
5.DSM disabled or CRC fails	->	DS_Idle

DS_Readwait	Wait for the read wait stop command.	
1.ReadWait stop enabled	->	DS_WaitR
2.DSM disabled	->	DS_Idle

30.4.2. APB2 interface

The APB2 interface implements access to SDIO registers, data FIFO and generates interrupt and DMA request. It includes a data FIFO unit, registers unit, and the interrupt / DMA logic.

The interrupt logic generates interrupt when at least one of the selected status flags is high. An interrupt enable register is provided to allow the logic to generate a corresponding interrupt.

The DMA interface provides a method for fast data transfers between the SDIO data FIFO and memory. The following example describes how to implement this method:

1. Complete the card identification process

2. Increase the SDIO_CLK frequency

3. Send CMD7 to select the card and configure the bus width

4. Configure the DMA1 as follows:

Enable DMA1 controller and clear any pending interrupts. Configure the DMA1_Channel3 source address register with the memory base address and DMA1_Channel3 destination address register with the SDIO_FIFO register address. Program DMA1_Channel3 control register (memory increment, not peripheral increment, peripheral and source width is word size, M2M disable).

5. Write block to card as follows:

Write the data size in bytes in the SDIO_DATALEN register. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register; the host sends data in blocks of size BLKSZ each. Program SDIO_CMDAGMT register with the data address, where data should be written. Program the SDIO command control register (SDIO_CMDCTL): CMDIDX with 24, CMDRESP with 1 (SDIO card host waits for a short response); CSMEN with '1' (enable to send a command). Other fields are their reset value.

When the CMDRECV flag is set, program the SDIO data control register (SDIO_DATACTL): DATAEN with '1' (enable to send data); DATADIR with '0' (from controller to card); TRANSMOD with '0' (block data transfer); DMAEN with '1' (DMA enabled); BLKSZ with 0x9 (512 bytes). Other bits don't care.

Wait for DTBLKEND flag is set. Check that no channels are still enabled by polling the DMA Interrupt Flag register.

It consists the following subunits:

Register unit

The register unit which contains all system registers generates the signals to control the communication between the controller and card.

Data FIFO

The data FIFO unit has a data buffer, uses as transmit and receive FIFO. The FIFO contains a 32-bit wide, 32-word deep data buffer. The transmit FIFO is used when write data to card and TXRUN in SDIO_STAT register is 1. The data to be transferred is written to transmit FIFO by APB2 bus, the data unit in SDIO adapter read data from transmit FIFO, and then send the data to card. The receive FIFO is used when read data from card and RXRUN in SDIO_STAT register is 1. The data to be transferred is read from the card and then write to receive FIFO. The data in receive FIFO is read to APB2 bus when needed. This unit also generates FIFO flags in SDIO_STAT registers.

30.5. Card functional description

30.5.1. Card registers

Within the card interface registers are defined: OCR, CID, CSD, EXT_CSD, RCA, DSR and SCR. These can be accessed only by corresponding commands. The OCR, CID, CSD and SCR registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters. The EXT_CSD register carries both, card specific information and actual configuration parameters. For specific information, please refer to the relevant specifications.

OCR register: The 32-bit operation conditions register (OCR) stores the V_{DD} voltage profile

of the card and the access mode indication (MMC). In addition, this register includes a status information bit. This status bit is set if the card power up procedure has been finished. The register is a little different between MMC and SD card. The host can use CMD1 (MMC), ACMD41 (SD memory), CMD5 (SD I/O) to get the content of this register.

CID register: The Card Identification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase. Every individual Read/Write (RW) card shall have a unique identification number. The host can use CMD2 and CMD10 to get the content of this register.

CSD register: The Card-Specific Data register provides information regarding access to the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used, etc. The programmable part of the register can be changed by CMD27. The host can use CMD9 to get the content of this register.

Extended CSD Register: Just MMC4.2 has this register. The Extended CSD register defines the card properties and selected modes. It is 512 bytes long. The most significant 320 bytes are the Properties segment, which defines the card capabilities and cannot be modified by the host. The lower 192 bytes are the Modes segment, which defines the configuration the card is working in. These modes can be changed by the host by means of the SWITCH command. The host can use CMD8 (just MMC supports this command) to get the content of this register.

RCA register: The writable 16-bit relative card address register carries the card address that is published by the card during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The host can use CMD3 to ask the card to publish a new relative address (RCA).

Note: The default value of the RCA register is 0x0001(MMC) or 0x0000(SD / SD I/O). The default value is reserved to set all cards into the Stand-by State with CMD7.

DSR register (Optional): The 16-bit driver stage register can be optionally used to improve the bus performance for extended operating conditions (depending on parameters like bus length, transfer rate or number of cards). The CSD register carries the information about the DSR register usage. The default value of the DSR register is 0x404. The host can use CMD4 to get the content of this register.

SCR register: Just SD/SD I/O (if has memory port) have this register. In addition to the CSD register, there is another configuration register named SD CARD Configuration Register (SCR), which is only for SD card. SCR provides information on the SD Memory Card's special features that were configured into the given card. The size of SCR register is 64 bits. This register shall be set in the factory by the SD Memory Card manufacturer. The host can use ACMD51 to get the content of this register.

30.5.2. Commands

Commands types

There are four kinds of commands defined to control the Card:

- Broadcast commands (bc), no response.
- Broadcast commands with response (bcr) response from all cards simultaneously.
- Addressed (point-to-point) commands (ac) no data transfer on DAT.
- Addressed (point-to-point) data transfer commands (adtc) data transfer on DAT.

Command format

All commands have a fixed code length of 48 bits, as show in ***Figure 30-7. Command token format***, needing a transmission time of 1.92μs (25 MHz) 0.96μs (50 MHz) and 0.92μs (52 MHz).

Figure 30-7. Command token format

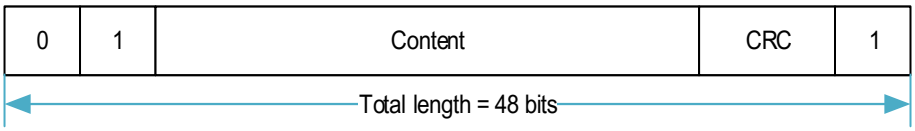


Table 30-2. Command format

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

A command always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (host = 1). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by 'x' in the table above indicates this variable is dependent on the command. All commands are protected by a CRC7. Every command code word is terminated by the end bit (always 1).

Command classes

The command set of the Card system is divided into several classes (See ***Table 30-3. Card command classes (CCCs)***). Each class supports a set of card functionalities. ***Table 30-3. Card command classes (CCCs)*** determines the setting of CCC from the card supported commands.

For SD cards, Class 0, 2, 4, 5 and 8 are mandatory and shall be supported. Class 7 except CMD40 is mandatory for SDHC. The other classes are optional. The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register

of each card, providing the host with information on how to access the card.

For MMC cards, Class 0 is mandatory and shall be supported. The other classes are either mandatory only for specific card types or optional. By using different classes, several configurations can be chosen (e.g. a block writable card or a stream readable card). The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

For CE-ATA device, the device shall support the MMC commands required to achieve the transfer state during device initialization. Other interface configuration settings, such as bus width, may require additional MMC commands also be supported. See the MMC reference. CE-ATA makes use of the following MMC commands: CMD0 - GO_IDLE_STATE, CMD12 - STOP_TRANSMISSION, CMD39 - FAST_IO, CMD60 - RW_MULTIPLE_REGISTER, CMD61 - RW_MULTIPLE_BLOCK. GO_IDLE_STATE (CMD0), STOP_TRANSMISSION (CMD12), and FAST_IO (CMD39) are as defined in the MMC reference. RW_MULTIPLE_REGISTER (CMD60) and RW_MULTIPLE_BLOCK (CMD61) are MMC commands defined by CE-ATA.

Table 30-3. Card command classes (CCCs)

	Card command class(CCC)	0	1	2	3	4	5	6	7	8	9	10	11
Supported command	Class description	basic	Stream read	Block read	Stream write	Block write	erase	write protection	Lock card	application specific	I/O mode	switch	reserved
CMD0	M	+											
CMD1	M	+											
CMD2	M	+											
CMD3	M	+											
CMD4	M	+											
CMD5	O										+		
CMD6	M											+	
CMD7	M	+											
CMD8	M	+											
CMD9	M	+											
CMD10	M	+											
CMD11	M		+										
CMD12	M	+											
CMD13	M	+											
CMD14	M	+											
CMD15	M	+											
CMD16	M			+		+			+				

CMD17	M			+									
CMD18	M			+									
CMD19	M	+											
CMD20	M				+								
CMD23	M			+		+							
CMD24	M					+							
CMD25	M					+							
CMD26	M					+							
CMD27	M					+							
CMD28	M							+					
CMD29	M							+					
CMD30	M							+					
CMD32	M						+						
CMD33	M						+						
CMD34	O											+	
CMD35	O											+	
CMD36	O											+	
CMD37	O											+	
CMD38	M						+						
CMD39											+		
CMD40											+		
CMD42									+				
CMD50	O											+	
CMD52	O										+		
CMD53	O										+		
CMD55	M									+			
CMD56	M									+			
CMD57	O											+	
CMD60	M									+			
CMD61	M									+			
ACMD6	M									+			
ACMD13	M									+			
ACMD22	M									+			
ACMD23	M									+			
ACMD41	M									+			
ACMD42	M									+			
ACMD51	M									+			

Note: 1.CMD1, CMD11, CMD14, CMD19, CMD20, CMD23, CMD26, CMD39 and CMD40 are only available for MMC.CMD5, CMD32-34, CMD50, CMD52, CMD53, CMD57 and ACMDx are only available for SD card. CMD60, CMD61 are only available for CE-ATA device.

2. All the ACMDs shall be preceded with APP_CMD command (CMD55).

3. CMD8 has different meaning for MMC and SD memory.

Detailed command description

The following tables describe in detail all bus commands. The responses R1-R7 are defined in ***Responses***. The registers CID, CSD and DSR are described in ***Card registers***. The card shall ignore stuff bits and reserved bits in an argument.

Table 30-4. Basic commands (class 0)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all cards to idle state
CMD1	bc	[31:0] OCR without busy	R3	SEND_OP_COND	Asks the card, in idle state, to send its Operating Conditions Register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks any card to send the CID numbers on the CMD line (any card that is connected to the host will respond)
CMD3	bcr	[31:0] stuff bits	R6	SEND_RELATIVE_ADDR	Ask the card to publish a new relative address (RCA)
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards
CMD5	bcr	[31:25]reserved bits [24]S18R [23:0] I/O OCR	R4	IO_SEND_OP_COND	Only for I/O cards. It is similar to the operation of ACMD41 for SD memory cards, used to inquire about the voltage range needed by the I/O card.
CMD6	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Only for MMC. Switches the mode of operation of the selected card or modifies the EXT_CSD registers.
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b	SELECT/DESELECT_CARD	Command toggles a card between the stand-by and transfer states or between the programming and disconnects states. In both cases the card is selected by its own relative address and gets deselected by any other address; address 0 deselects the card.

CMD8	bcr	[31:12]reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition, which includes host supply voltage information and asks the card whether card supports voltage. Reserved bits shall be set to '0'.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	For MMC only. The card sends its EXT_CSD register as a block of data.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card identification (CID) on CMD the line.
CMD12	ac	[31:0] stuff bits	R1b	STOP TRANSMISSION	Forces the card to stop transmission
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	adtc	[31:0] stuff bits	R1	BUSTEST_R	A host reads the reversed bus testing data pattern from a card.
CMD15	ac	[31:16] RCA [15:0] reserved-bits		GO_INACTIVE_STATE	Sends an addressed card into the Inactive State. This command is used when the host explicitly wants to deactivate a card.
CMD19	adtc	[31:0] stuff bits	R1	BUSTEST_W	A host sends the bus test data pattern to a card.

Table 30-5. Block-Oriented read commands (class 2)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	In the case of a Standard Capacity SD and MMC, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default is 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD. In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed

					block length is used. In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	In the case of a Standard Capacity SD and MMC, this command reads a block of the size selected by the SET_BLOCKLEN command. In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command. Block length is specified the same as READ_SINGLE_BLOCK command.
Note: The transferred data must not cross a physical block boundary, unless READ_BLK_MISALIGN is set in the CSD register					

Table 30-6. Stream read commands (class 1) and stream write commands (class 3)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISSION follows.
Note: The transferred data must not cross a physical block boundary, unless READ_BLK_MISALIGN is set in the CSD register					

Table 30-7. Block-Oriented write commands (class 4)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	See description in *Table 30-5. Block-Oriented read commands (class 2)*.

Cmd index	type	argument	Response format	Abbreviation	Description
CMD23	ac	[31:16] set to 0 [15:0] number of blocks	R1	SET_BLOCK_ COUNT	Defines the number of blocks which are going to be transferred in the immediately succeeding multiple block read or write command. If the argument is all 0s, the subsequent read/write operation will be open-ended.
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	In the case of a Standard Capacity SD, this command writes a block of the size selected by the SET_BLOCKLEN command. In the case of a SDHC, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE _BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows. Block length is specified the same as WRITE_BLOCK command.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
Note: 1.The data transferred shall not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD. In the case that write partial blocks is not supported, then the block length=default block length (given in CSD). 2. Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.					

Table 30-8. Erase commands (class 5)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK_S TART	Sets the address of the first write block to be erased.(SD)
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK_E ND	Sets the address of the last write block of the continuous range to be

					erased.(SD)
CMD35	ac	[31:0]data address	R1	ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.(MMC)
CMD36	ac	[31:0]data address	R1	ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.(MMC)
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erases all previously selected write blocks.
Note: 1.CMD34 and CMD37 are reserved in order to maintain backwards compatibility with older versions of the MMC. 2. Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.					

Table 30-9. Block oriented write protection commands (class 6)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). A High Capacity SD Memory Card does not support this command.
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
Note: 1. High Capacity SD Memory Card does not support these three commands.					

Table 30-10. Lock card (class 7)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCK_LEN	See description in *Table 30-5. Block-Oriented read commands (class 2)*.
CMD42	adtc	[31:0] Reserved bits (Set all 0)	R1	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the

					SET_BLOCK_LEN command. Reserved bits in the argument and in Lock Card Data Structure shall be set to 0.

Table 30-11. Application-specific commands (class 8)

Cmd index	type	argument	Response format	Abbreviation	Description
ACMD41	bcr	[31]reserved bit [30]HCS [29:24]reserved bits [23:0]V_{DD} Voltage Window(OCR[23:0])	R3	SD_SEND_OP_COND	Sends host capacity support information (HCS) and asks the accessed card to send its operating condition register(OCR) content in the response. HCS is effective when card receives SEND_IF_COND command. CCS bit is assigned to OCR[30].
ACMD42	ac	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_DETECT	Connect[1]/Disconnect[0] the 50K pull-up resistor on CD/DAT3 (pin 1) of the card.
ACMD51	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command.
CMD56	adtc	[31:1] stuff bits. [0] RD/WR	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose/application specific command. The host sets RD/WR=1 for reading data from the card and sets to 0 for writing data to the card.
CMD60	adtc	[31] WR [23:18] Address [7:2] Byte Count Other bits are reserved bits.	R1(read)/ R1b(write)	RW_MULTIPLE_REGISTER	Read or write register in address range.
CMD61	adtc	[31] WR [15:0] Data Unit Count Other bits are reserved bits	R1(read)/ R1b(write)	RW_MULTIPLE_BLOCK	Read or write data block in address range.

Cmd index	type	argument	Response format	Abbreviation	Description
Note: 1.ACMDx is Application-specific Commands for SD memory. 2. CMD60, CMD61 are Application-specific Commands for CE-ATA device.					

Table 30-12. I/O mode commands (class 9)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD39	ac	[31:16] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8 bit (register) data fields. The command addresses a card and a register and provides the data for writing if the write flag is set. The R4 response contains data read from the addressed register if the write flag is cleared to 0. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode
CMD52	adtc	[31] R/W Flag [30:28] Function Number [27] RAW Flag [26] Stuff Bits [25:9] Register Address [8] Stuff Bits [7:0] Write Data/Stuff Bits	R5	IO_RW_DIRECT	The IO_RW_DIRECT is the simplest means to access a single register within the total 128K of register space in any I/O function, including the common I/O area (CIA). This command reads or writes 1 byte using only 1 command/response pair. A common use is to initialize registers or monitor status values for I/O functions. This command is the fastest means to read or write single I/O registers, as it requires only a single command/response pair.
CMD53	adtc	[31] R/W Flag [30:28] Function Number [27] Block Mode [26] OP code [25:9] Register Address [8:0] Byte/Block		IO_RW_EXTENDED	This command allows the reading or writing of a large number of I/O registers with a single command.

Cmd index	type	argument	Response format	Abbreviation	Description
		Count			
Note: 1.CMD39, CMD40 are only for MMC. 2. CMD52, CMD53 are only for SD I/O card.					

Table 30-13. Switch function commands (class 10)

Cmd index	type	argument	Response format	Abbreviation	Description
CMD6	adtc	[31] Mode 0:Check function 1:Switch function [30:24] reserved [23:20] reserved for function group 6 (0h or Fh) [19:16] reserved for function group 5 (0h or Fh) [15:12] reserved for function group 4 (0h or Fh) [11:8] reserved for function group 3 (0h or Fh) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_FUNC	Only for SD memory and SD I/O. Checks switchable function (mode 0) and switch card function (mode 1).

30.5.3. Responses

All responses are sent on the CMD line. The response transmission always starts with the left bit of the bit string corresponding to the response code word. The code length depends on the response type.

Responses types

There are 7 types of responses show as follows.

- **R1 / R1b :** normal response command.
- **R2 :** CID, CSD register.
- **R3 :** OCR register.
- **R4 :** Fast I/O.
- **R5 :** Interrupt request.
- **R6 :** Published RCA response.
- **R7 :** Card interface condition.

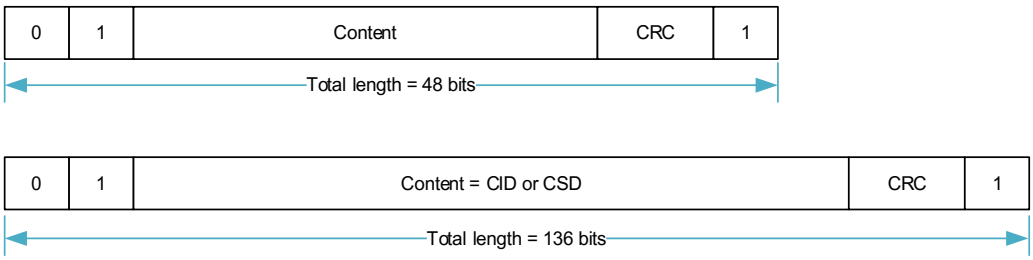
The SD Memory Card support five types of them, R1 / R1b, R2, R3, R6, R7. And the SD I/O

Card and MMC supports additional response types named R4 and R5, but they are not exactly the same for SD I/O Card and MMC.

Responses format

Responses have two formats, as show in *Figure 30-8. Response Token Format*, all responses are sent on the CMD line. The code length depends on the response type. Except R2 is 136 bits length, others are all 48 bits length.

Figure 30-8. Response Token Format



A response always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (card = 0). A value 'x' in the tables below indicates a variable entry. All responses except for the type R3 are protected by a CRC. Every command code word is terminated by the end bit (always 1).

R1 (normal response command)

Code length is 48 bits. The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits. Note that if a data transfer to the card is involved, then a busy signal may appear on the data line after the transmission of each block of data. The host shall check for busy after data block transmission. The card status is described in *Two status fields of the card*.

Table 30-14. Response R1

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'0'	x	x	x	'1'
description	start bit	transmission bit	command index	card status	CRC7	end bit

R1b

R1b is identical to R1 with an optional busy signal transmitted on the data line DAT0. The card may become busy after receiving these commands based on its state prior to the command reception. The Host shall check for busy at the response.

R2 (CID, CSD register)

Code length is 136 bits. The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127..1] of the CID and CSD are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response.

Table 30-15. Response R2

Bit position	135	134	[133:128]	[127:1]	0
Width	1	1	6	127	1
Value	'0'	'0'	'111111'	x	'1'
description	start bit	transmission bit	reserved	CID or CSD register and internal CRC7	end bit

R3 (OCR register)

Code length is 48 bits. The contents of the OCR register are sent as a response to ACMD41 (SD memory), CMD1 (MMC). The response of different cards may have a little different.

Table 30-16. Response R3

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Value	'0'	'0'	'111111'	x	'1111111'	'1'
description	start bit	transmission bit	reserved	OCR register	reserved	end bit

R4 (Fast I/O)

For MMC only. Code length 48 is bits. The argument field contains the RCA of the addressed card, the register address to be read out or written to, and its contents. The status bit in the argument is set if the operation was successful.

Table 30-17. Response R4 for MMC

Bit position	47	46	[45:40]	[39:8] Argument field				[7:1]	0
Width	1	1	6	16	1	7	8	7	1
Value	'0'	'0'	'100111'	x	x	x	x	x	'1'
description	start bit	transmission bit	CMD39	RCA [31:16]	status [15]	register address [14:8]	read register contents [7:0]	CRC7	end bit

R4b

For SD I/O only. Code length is 48 bits. The SDIO card receive the CMD5 will respond with a unique SD I/O response R4.

Table 30-18. Response R4 for SD I/O

Bit position	47	46	[45:40]	39	[38:36]	35	[34:32]	31	[30:8]	[7:1]	0
Width	1	1	6	1	3	1	3	1	23	7	1
Value	'0'	'0'	'111111'	x	x	x	'000'	x	x	'1111111'	1
description	start bit	transmission bit	Reserved	C	Number of I/O functions	Memory Present	Stuff Bits	S18A	I/O OCR	Reserved	end bit

R5 (Interrupt request)

For MMC only. Code length is 48 bits. If the response is generated by the host, the RCA field in the argument will be 0x0.

Table 30-19. Response R5 for MMC

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width	1	1	6	16	16	7	1
Value	'0'	'0'	'101000'	x	x	x	'1'
description	start bit	transmission bit	CMD40	RCA [31:16] of winning card or of the host	[15:0] Not defined. May be used for IRQ data	CRC7	end bit

R5b

For SD I/O only. The SDIO card's response to CMD52 and CMD53 is R5. If the communication between the card and host is in the 1-bit or 4-bit SD mode, the response shall be in a 48-bit response (R5).

Table 30-20. Response R5 for SD I/O

Bit position	47	46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
Width	1	1	6	16	8	8	7	1
Value	'0'	'0'	'11010X'	'0'	x	x	x	'1'
description	start bit	transmission bit	CMD52/53	Stuff Bits	Response Flags	Read or Write Data	CRC7	end bit

R6 (Published RCA response)

Code length is 48 bit. The bits [45:40] indicate the index of the command to be responded to (CMD3). The 16 MSB bits of the argument field are used for the Published RCA number.

Table 30-21. Response R6

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width	1	1	6	16	16	7	1
Value	'0'	'0'	'000011'	x	x	x	'1'
description	start	transmission bit	CMD3	New published RCA of	card status	CRC7	end

	bit			the card	bits:23,22,19,12:0		bit

R7 (Card interface condition)

For SD memory only. Code length is 48 bits. The card support voltage information is sent by the response of CMD8. Bits 19-16 indicate the voltage range that the card supports. The card that accepted the supplied voltage returns R7 response. In the response, the card echoes back both the voltage range and check pattern set in the argument.

Table 30-22. Response R7

Bit position	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Width	1	1	6	20	4	8	7	1
Value	'0'	'0'	'001000'	'00000h'	x	x	x	'1'
description	start bit	transmission bit	CMD8	Reserved bits	Voltage accepted	echo-back of check pattern	CRC7	end bit

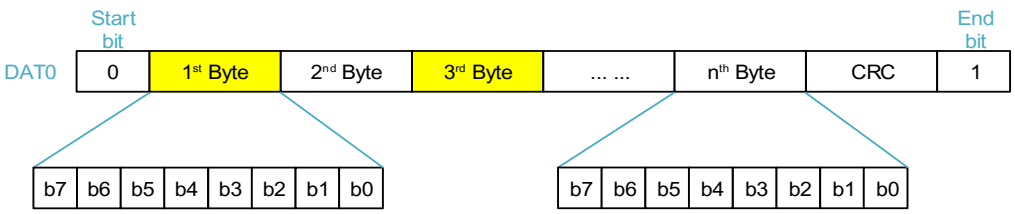
30.5.4. Data packets format

There are 3 data bus mode, 1-bit, 4-bit and 8-bit width. 1-bit mode is mandatory, 4-bit and 8-bit mode is optional. Although using 1-bit mode, DAT3 also need to notify card current working mode is SDIO or SPI, when card reset and initialize.

1-bit data packet format

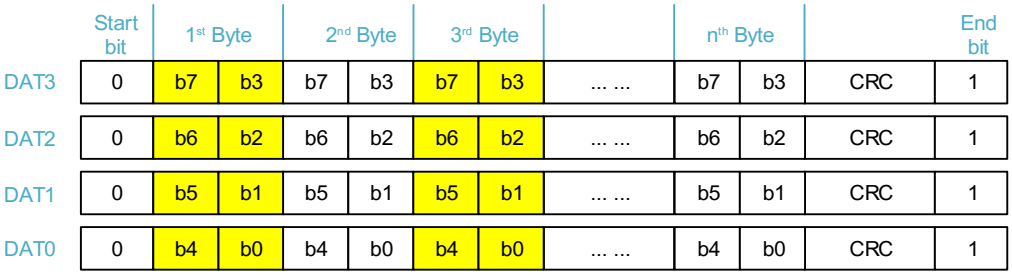
After card reset and initialize, only DAT0 pin is used to transfer data. And other pin can be used freely. ***Figure 30-9. 1-bit data bus width***, ***Figure 30-10. 4-bit data bus width*** and ***Figure 30-11. 8-bit data bus width*** show the data packet format when data bus wide is 1-bit, 4-bit and 8-bit.

Figure 30-9. 1-bit data bus width



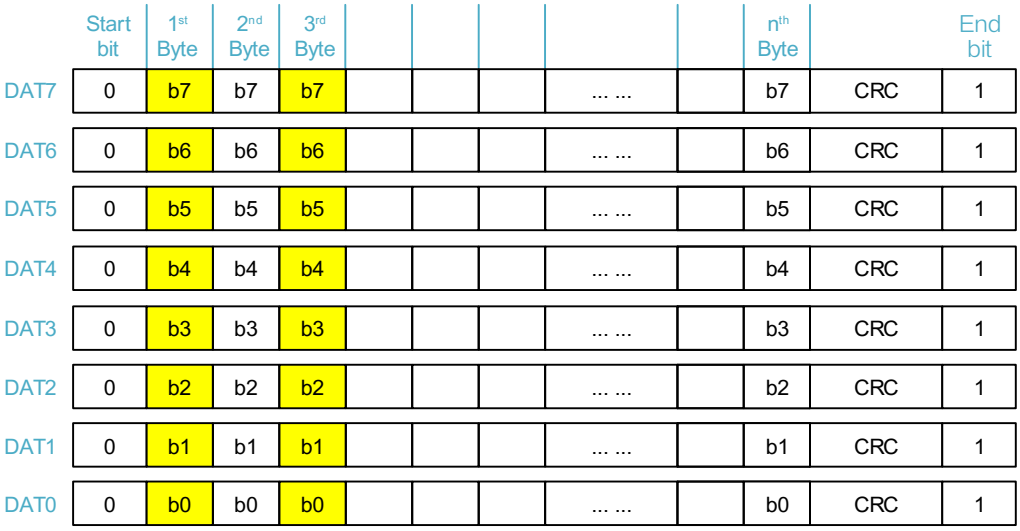
4-bit data packet format

Figure 30-10. 4-bit data bus width



8-bit data packet format

Figure 30-11. 8-bit data bus width



30.5.5. Two status fields of the card

The SD Memory supports two status fields and others just support the first one:

Card Status: Error and state information of a executed command, indicated in the response

SD Status: Extended status field of 512 bits that supports special features of the SD Memory Card and future Application-Specific features.

Card status

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command.

The type and clear condition fields in the table are abbreviated as follows:

Type

•E: Error bit. Send an error condition to the host. These bits are cleared as soon as the response (reporting the error) is sent out.

•S: Status bit. These bits serve as information fields only, and do not alter the execution of the command being responded to. These bits are persistent, they are set and cleared in accordance with the card status.

•R: Exceptions are detected by the card during the command interpretation and validation phase (Response Mode).

•X: Exceptions are detected by the card during command execution phase (Execution Mode).

Clear condition

•A: According to current state of the card.

•B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).

•C: Cleared by read

Table 30-23. Card status

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	ERX	'0'= no error '1'= error	The command's argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	ERX	'0'= no error '1'= error	A misaligned address which did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	ERX	'0'= no error '1'= error	The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.	C
28	ERASE_SEQ_ERROR	ER	'0'= no error '1'= error	An error in the sequence of erase commands occurred.	C
27	ERASE_PARAM	ERX	'0'= no error '1'= error	An invalid selection of write-blocks for erase occurred.	C
26	WP_VIOLATION	ERX	'0'= not protected '1'= protected	Set when the host attempts to write to a protected block or to the temporary or permanent write protected card.	C
25	CARD_IS_LOCKED	SX	'0' = card unlocked '1' = card locked	When set, signals that the card is locked by the host	A

24	LOCK_UNLOCK_FAILED	ERX	'0'= no error '1'= error	Set when a sequence or password error has been detected in lock/unlock card command.	C
23	COM_CRC_ERROR	ER	'0'= no error '1'= error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	ER	'0'= no error '1'= error	Command not legal for the card state.	B
21	CARD_ECC_FAILED	ERX	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data.	C
20	CC_ERROR	ERX	'0'= no error '1'= error	Internal card controller error.	C
19	ERROR	ERX	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	UNDERRUN	ERX	'0'= no error '1'= error	Only for MMC. The card could not sustain data transfer in stream read mode.	C
17	OVERRUN	ERX	'0'= no error '1'= error	Only for MMC. The card could not sustain data programming in stream write mode.	C
16	CID/ CSD_OVERWRITE	ERX	'0'= no error '1'= error	Can be either one of the following errors: - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP(unprotected) bits was made.	C
15	WP_ERASE_SKIP	ERX	'0'= not protected '1'= protected	Set when only partial address space was erased due to existing write protected blocks or the temporary or permanent write protected card was erased.	C
14	CARD_ECC_DISABLED	SX	'0'= enabled '1'= disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	SR	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received.	C
[12:9]	CURRENT_STATE	SX	0 = idle 1 = ready 2 = identification 3 = stand by 4 = transfer 5 = send data	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are	B

			6 = receive data 7 = programming 8 = disconnect 9-14 = reserved 15 = reserved for I/O mode	interpreted as a binary coded number between 0 and 15.	
8	READY_FOR_DATA	SX	'0'= not ready '1'= ready	Corresponds to buffer empty signaling on the bus.	A
7	SWITCH_ERROR	EX	'0'= no error '1'= switch error	If set, the card don't switch to the expected mode as requested by the SWITCH command.	B
6	Reserved				
5	APP_CMD	SR	'0'= enabled '1'= disabled	The card will expect ACMD, or an indication that the command has been interpreted as ACMD.	C
4	Reserved				
3	AKE_SEQ_ERROR	ER	'0'= no error '1'= error	Only for SD memory. Error in the sequence of the authentication process.	C
2	Reserved for application specific commands.				
[1:0]	Reserved for manufacturer test mode.				

Note: 18, 17, 7 bits are only for MMC. 14, 3 bits are only for SD memory.

SD status register

The SD Status contains status bits that are related to the SD Memory Card proprietary features and may be used for future application-specific usage. The size of the SD Status is one data block of 512 bits. The content of this register is transmitted to the Host over the DAT bus along with a 16-bit CRC. The SD Status is sent to the host over the DAT bus as a response to ACMD13 (CMD55 followed with CMD13). ACMD13 can be sent to a card only in 'transfer state' (card is selected). The SD Status structure is described below.

The same abbreviation for 'type' and 'clear condition' were used as for the Card Status above.

Table 30-24. SD status

Bits	Identifier	Type	Value	Description	Clear Condition
[511:510]	DAT_BUS_WIDTH	SR	'00'= 1 (default) '01'= reserved '10'= 4 bit width '11'= reserved	Shows the currently defined data bus width that was defined by SET_BUS_WIDTH command	A
509	SECURED_MODE	SR	'0'= Not in the mode '1'= In Secured Mode	Card is in Secured Mode of operation (refer to the "SD	A

Bits	Identifier	Type	Value	Description	Clear Condition
				Security Specification").	
[508:496]	reserved				
[495:480]	SD_CARD_TYPE	SR	The following cards are currently defined: '0000'= Regular SD RD/WR Card. '0001'= SD ROM Card '0002'= OTP	In the future, the 8 LSBs will be used to define different variations of an SD Memory Card (Each bit will define different SD Types). The 8 MSBs will be used to define SD Cards that do not comply with current SD Physical Layer Specification.	A
[479:448]	SIZE_OF_PROTECTED_AREA	SR	Size of protected area	(See below)	A
[447:440]	SPEED_CLASS	SR	Speed class of the card	(See below)	A
[439:432]	PERFORMANCE_MOVE	SR	Performance of move indicated by 1 [MB/s] step.	(See below)	A
[431:428]	AU_SIZE	SR	Size of AU	(See below)	A
[427:424]	reserved				
[423:408]	ERASE_SIZE	SR	Number of AUs to be erased at a time	(See below)	A
[407:402]	ERASE_TIMEOUT	SR	Timeout value for erasing areas specified by UNIT_OF_ERASE_AU	(See below)	A
[401:400]	ERASE_OFFSET	SR	Fixed offset value added to erase time.	(See below)	A
[399:312]	reserved				
[311:0]	reserved for manufacturer				

SIZE_OF_PROTECTED_AREA

Setting this field differs between SDSC and SDHC/SDXC.

In case of SDSC Card, the capacity of protected area is calculated as follows:

Protected Area = SIZE_OF_PROTECTED_AREA_* MULT * BLOCK_LEN.

SIZE_OF_PROTECTED_AREA is specified by the unit in MULT*BLOCK_LEN.

In case of SDHC and SDXC Cards, the capacity of protected area is calculated as follows:

Protected Area = SIZE_OF_PROTECTED_AREA

SIZE_OF_PROTECTED_AREA is specified by the unit in byte.

SPEED_CLASS

This 8-bit field indicates the Speed Class.

00h: Class 0

01h: Class 2

02h: Class 4

03h: Class 6

04h: Class 10

05h-FFh: Reserved

PERFORMANCE_MOVE

This 8-bit field indicates Pm and the value can be set by 1 [MB/sec] step. If the card does not move useing RUs, Pm should be considered as infinity. Setting to FFh means infinity. The minimum value of Pm is defined in *Table 30-25. Performance move field*.

Table 30-25. Performance move field

PERFORMANCE_MOVE	Value Definition
00h	Sequential Write
01h	1 [MB/sec]
02h	2 [MB/sec]
.....	.....
FEh	254 [MB/sec]
FFh	Infinity

AU_SIZE

This 4-bit field indicates AU Size and the value can be selected from 16 KB.

Table 30-26. AU_SIZE field

AU_SIZE	Value Definition
0h	Not Defined
1h	16 KB
2h	32 KB
3h	64 KB
4h	128 KB
5h	256 KB
6h	512 KB
7h	1 MB

AU_SIZE	Value Definition
8h	2 MB
9h	4 MB
Ah	8 MB
Bh	12 MB
Ch	16 MB
Dh	24 MB
Eh	32 MB
Fh	64 MB

The maximum AU size, depends on the card capacity, is defined in ***Table 30-26. AU_SIZE field***. The card can set any AU size specified in ***Table 30-27. Maximum AU size*** that is less than or equal to the maximum AU size. The card should set smaller AU size as possible.

Table 30-27. Maximum AU size

Card Capacity	up to 64MB	up to 256MB	up to 512MB	up to 32GB	up to 2TB
Maximum AU Size	512 KB	1 MB	2 MB	4 MB1	64MB

ERASE_SIZE

This 16-bit field indicates N_{ERASE}. When N_{ERASE} of AUs are erased, the timeout value is specified by ERASE_TIMEOUT (Refer to ERASE_TIMEOUT). The host should determine proper number of AUs to be erased in one operation so that the host can indicate progress of erase operation. If this field is set to 0, the erase timeout calculation is not supported.

Table 30-28. Erase size field

ERASE_SIZE	Value Definition
0000h	Erase Time-out Calculation is not supported.
0001h	1 AU
0002h	2 AU
0003h	3 AU
.....	.....
FFFFh	65535 AU

ERASE_TIMEOUT

This 6-bit field indicates the T_{ERASE} and the value indicates erase timeout from offset when multiple AUs are erased as specified by ERASE_SIZE. The range of ERASE_TIMEOUT can be defined as up to 63 seconds and the card manufacturer can choose any combination of ERASE_SIZE and ERASE_TIMEOUT depending on the implementation. Once ERASE_TIMEOUT is determined, it determines the ERASE_SIZE. The host can determine timeout for any number of AU erase by the equation below.

$$\text{Erase timeout of X AU} = \frac{T_{ERASE}}{N_{ERASE}} * X + T_{OFFSET} \quad (30\text{-}1)$$

Table 30-29. Erase timeout field

ERASE_TIMEOUT	Value Definition
00	Erase Time-out Calculation is not supported.
01	1 [sec]
02	2 [sec]
03	3 [sec]
.....	.....
63	63 [sec]

If ERASE_SIZE field is set to 0, this field shall be set to 0.

ERASE_OFFSET

This 2-bit field indicates the T_{OFFSET} and one of four values can be selected. This field is meaningless if ERASE_SIZE and ERASE_TIMEOUT fields are set to 0.

Table 30-30. Erase offset field

ERASE_OFFSET	Value Definition
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

30.6. Programming sequence

30.6.1. Card identification

The host will be in card identification mode after reset and while it is looking for new cards on the bus. While in card identification mode the host resets all the cards, validates operation voltage range, identifies cards and asks them to publish Relative Card Address (RCA). This operation is done to each card separately on its own CMD line. All data communication in the Card Identification Mode uses the command line (CMD) only.

During the card identification process, the card shall operate in the clock frequency of the identification clock rate F_{OD} (400 kHz).

Card reset

The command GO_IDLE_STATE (CMD0) is the software reset command and sets MMC and SD memory card into Idle State regardless of the current card state. The reset command (CMD0) is only used for memory or the memory portion of Combo cards. In order to reset an I/O only card or the I/O portion of a combo card, use CMD52 to write 1 to the RES bit in the CCCR. Cards in Inactive State are not affected by this command.

After power-on by the host, all cards are in Idle State, including the cards that have been in Inactive State before. After power-on or CMD0, all cards' CMD lines are in input mode, waiting

for start bit of the next command. The cards are initialized with a default relative card address (RCA) and with a default driver strength with 400 KHz clock frequency.

Operating voltage range validation

At the start of communication between the host and the card, the host may not know the card supported voltage and the card may not know whether it supports the current supplied voltage. To verify the voltage, the following commands are defined in the related specification.

The SEND_OP_COND (CMD1 for MMC), SD_SEND_OP_COND (ACMD41 for SD memory), IO_SEND_OP_COND (CMD5 for SD I/O) command is designed to provide hosts with a mechanism to identify and reject cards which do not match the V_{DD} range desired by the host. This is accomplished by the host sending the required V_{DD} voltage window as the operand of this command. If the card cannot perform data transfer in the specified range it must discard itself from further bus operations and go into Inactive State. Otherwise, the card shall respond sending back its V_{DD} range.

If the card can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument.

If the card cannot operate on the supplied voltage, it returns no response and stays in idle state. It is mandatory to issue CMD8 prior to ACMD41 to initialize SDHC Card. Receipt of CMD8 makes the cards realize that the host supports the Physical Layer Version 2.00 and the card can enable new functions.

Card identification process

The card identification process differs in different cards. The card can be of the type MMC, CE-ATA, SD, or SD I/O. All types of SD I/O cards are supported, that is, SDIO_IO_ONLY, SDIO_MEM_ONLY, and SDIO COMBO cards. The identification process sequence includes the following steps:

1. Check if the card is connected.

2. Identify the card type; SD, MMC(CE-ATA), or SD I/O.

- Send CMD5 first. If a response is received, then the card is SD I/O

- If not, send ACMD41; if a response is received, then the card is SD.

- Otherwise, the card is an MMC or CE-ATA.

3. Initialization the card according to the card type.

Use a clock source with a frequency = F_{OD} (that is, 400 KHz) and use the following command sequence:

- SD card - Send CMD0, ACMD41, CMD2, CMD3.

- SDHC card - send CMD0, CMD8, ACMD41, CMD2, CMD3.

- SD I/O - Send CMD52, CMD0, CMD5, if the card doesn't have memory port, send CMD3; otherwise send ACMD41, CMD11 (optional), CMD2, and CMD3.

- MMC/CE-ATA - Send CMD0, CMD1, CMD2, CMD3.

4. Identify the MMC/CE-ATA device.

- CPU should query the byte 504 (S_CMD_SET) of EXT_CSD register by sending CMD8. If bit 4 is set to 1, then the device supports ATA mode.

- If ATA mode is supported, the CPU should select the ATA mode by setting the ATA bit (bit 4) in the EXT_CSD register slice 191(CMD_SET) to activate the ATA command set. The CPU selects the command set using the SWITCH (CMD6) command.

- In the presence of a CE-ATA device, the FAST_IO (CMD39) and RW_MULTIPLE_REGISTER (CMD60) commands will succeed and the returned data will be the CE-ATA reset signature.

30.6.2. No data commands

To send any non-data command, the software needs to program the SDIO_CMDCTL register and the SDIO_CMDAGMT register with appropriate parameters. Using these two registers, the host forms the command and sends it to the command bus. The host reflects the errors in the command response through the error bits of the SDIO_STAT register.

When a response is received the host sets the CMDRECV (CRC check passed) or CCRCERR (CRC check error) bit in the SDIO_STAT register. A short response is copied in SDIO_RESP0, while a long response is copied to all four response registers. The SDIO_RESP3 bit 31 represents the MSB, and the SDIO_RESP0 bit 0 represents the LSB of a long response.

30.6.3. Single block or multiple block write

During block write (CMD24 - 27) one or more blocks of data are transferred from the host to the card. The block consists of start bits (1 or 4 bits LOW), data block, CRC and end bits(1 or 4 bits HIGH). If the CRC fails, the card indicates the failure on the SDIO_DAT line and the transferred data are discarded and not written, and all further transmitted blocks are ignored.

If the host uses partial blocks whose accumulated length is not block aligned, block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card will detect the block misalignment error before the beginning of the first misaligned block. The card shall set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer. The write operation will also be aborted if the host tries to write data on a write protected area. In this case, however, the card will set the WP_VIOLATION bit (in the status register).

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer.

If this match fails, then the card reports an error and does not change any register contents.

Some cards may require long and unpredictable time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT0 line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the Disconnect State and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable.

For SD card. Setting a number of write blocks to be pre-erased (ACMD23) will make a following Multiple Block Write operation faster compared to the same operation without preceding ACMD23. The host will use this command to define how many number of write blocks are going to be send in the next write operation.

Steps involved in a single-block or multiple-block write are:

1. Write the data size in bytes in the SDIO_DATALEN register.

2. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register; the host sends data in blocks of size BLKSZ.

3. Program SDIO_CMDAGMT register with the data address to which data should be written.

4. Program the SDIO_CMDCTL register. For SD memory and MMC cards, use CMD24 for a single-block write and CMD25 for a multiple-block write. For SD I/O cards, use CMD53 for both single-block and multiple-block transfers. For CE-ATA, first use CMD60 to write the ATA task file, then use CMD61 to write the data. After writing to the CMD register, host starts executing a command, when the command is sent to the bus, the CMDRECV flag is set.

5. Write data to SDIO_FIFO.

6. Software should look for data error interrupts. If required, software can terminate the data transfer by sending the STOP command (CMD12).

7. When a DTEND interrupt is received, the data transfer is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the host should send the STOP command.

30.6.4. Single block or multiple block read

Block read is block oriented data transfer. The basic unit of data transfer is a block whose maximum size is defined in the CSD (READ_BL_LEN), it is always 512 bytes. If READ_BL_PARTIAL(in the CSD) is set, smaller blocks whose starting and ending address are entirely contained within 512 bytes boundary may be transmitted.

CMD17 (READ_SINGLE_BLOCK) initiates a block read and after completing the transfer, the card returns to the Transfer state. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. CRC is appended to the end of each block, ensuring data transfer integrity.

Block Length set by CMD16 can be set up to 512 bytes regardless of READ_BL_LEN.

Blocks will be continuously transferred until a STOP_TRANSMISSION command (CMD12) is issued. The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

When the last block of user area is read using CMD18, the host should ignore OUT_OF_RANGE error that may occur even the sequence is correct.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first misaligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command.

Steps involved in a single block or multiple block read are:

1. Write the data size in bytes in the SDIO_DATALEN register.

2. Write the block size in bytes (BLKSZ) in the SDIO_DATACTL register. The host expects data from the card in blocks of size BLKSZ each.

3. Program the SDIO_CMDAGMT register with the data address of the beginning of a data read.

4. Program the SDIO_CMDCTL. For SD and MMC cards, using CMD17 for a single-block read and CMD18 for a multiple-block read. For SD I/O cards, using CMD53 for both single-block and multiple-block transfers. For CE-ATA, first using CMD60 to write the ATA task file, then using CMD 61 to read the data. After writing to the CMD register, the host starts executing the command, when the command is sent to the bus, the CMDRECV flag is set.

5. Software should look for data error interrupts. If required, software can terminate the data transfer by sending a STOP command.

6. The software should read data from the FIFO and make space in the FIFO for receiving more data.

7. When a DTEND interrupt is received, the software should read the remaining data in the FIFO.

30.6.5. Stream write and stream read (MMC only)

Stream write

Stream write (CMD20) starts the data transfer from the host to the card beginning from the

starting address until the host issues a stop command. If partial blocks are allowed (if CSD parameter WRITE_BL_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it shall start and stop only at block boundaries. Since the amount of data to be transferred is not determined in advance, CRC cannot be used.

If the host provides an out of range address as an argument to CMD20, the card will reject the command, remain in Tran state and respond with the ADDRESS_OUT_OF_RANGE bit set.

Note that the stream write command works only on a 1 bit bus configuration (on DAT0). If CMD20 is issued in other bus configurations, it is regarded as an illegal command.

In order to sustain data transfer in stream mode of the card, the time it takes to receive the data (defined by the bus clock rate) must be less than the time it takes to program it into the main memory field (defined by the card in the CSD register). Therefore, the maximum clock frequency for the stream write operation is given by the following formula:

$$\text{max write frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{WRITE_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (30\text{-}2)$$

TRAN_SPEED: Max bus clock frequency.

WRITE_BL_LEN: Max write data block length.

NSAC: Data read access-time 2 in CLK cycles.

TAAC: Data read access-time 1.

R2W_FACTOR: Write speed factor.

All the parameters are defined in CSD register. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. As the host sends CMD12, the card will respond with the TXURE bit set and return to Transfer state.

Stream read

There is a stream oriented data transfer controlled by READ_DAT_UNTIL_STOP (CMD11). This command instructs the card to send its data, starting at a specified address, until the host sends a STOP_TRANSMISSION command (CMD12). The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

If the host provides an out of range address as an argument to CMD11, the card will reject the command, remain in Transfer state and respond with the ADDRESS_OUT_OF_RANGE bit set.

Note that the stream read command works only on a 1 bit bus configuration (on DAT0). If CMD11 is issued in other bus configurations, it is regarded as an illegal command.

If the end of the memory range is reached while sending data, and no stop command has been sent yet by the host, the contents of the further transferred payload is undefined. As the host sends CMD12 the card will respond with the ADDRESS_OUT_OF_RANGE bit set and return to Tran state.

In order to sustain data transfer in stream mode of the card, the time it takes to transmit the data (defined by the bus clock rate) must be less than the time it takes to read it out of the main memory field (defined by the card in the CSD register). Therefore, the maximum clock frequency for stream read operation is given by the following formula:

$$\text{max read frequence} = \min\left(\text{TRAN_SPEED}, \frac{8*2^{\text{READ_BL_LEN}}-100*\text{NSAC}}{\text{TAAC}*\text{R2W_FACTOR}}\right) \quad (30\text{-}3)$$

TRAN_SPEED: Max bus clock frequency.

READ_BL_LEN: Max read data block length.

NSAC: Data read access-time 2 in CLK cycles.

TAAC: Data read access-time 1.

R2W_FACTOR: Write speed factor.

All the parameters are defined in CSD register. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. As the host sends CMD12, the card will respond with the RXORE bit set and return to Transfer state.

30.6.6. Erase

The erasable unit of the MMC/SD memory is the "Erase Group"; Erase group is measured in write blocks which are the basic writable units of the card. The size of the Erase Group is a card specific parameter and defined in the CSD.

The host can erase a contiguous range of Erase Groups. Starting the erase process is a three steps sequence. First the host defines the start address of the range using the ERASE_GROUP_START (CMD35)/ERASE_WR_BLK_START(CMD32) command, next it defines the last address of the range using the ERASE_GROUP_END (CMD36)/ERASE_WR_BLK_END(CMD33) command and finally it starts the erase process by issuing the ERASE (CMD38) command. The address field in the erase commands is an Erase Group address in byte units. The card will ignore all LSB's below the Erase Group size, effectively rounding the address down to the Erase Group boundary.

If an erase command (CMD35, CMD36, and CMD38) is received out of the defined erase sequence, the card shall set the ERASE_SEQ_ERROR bit in the status register and reset the whole sequence.

If the host provides an out of range address as an argument to CMD35 or CMD36, the card will reject the command, respond with the ADDRESS_OUT_OF_RANGE bit set and reset the

whole erase sequence.

If an 'non erase' command (neither of CMD35, CMD36, CMD38 or CMD13) is received, the card shall respond with the ERASE_RESET bit set, reset the erase sequence and execute the last command.

If the erase range includes write protected blocks, they shall be left intact and only the non-protected blocks shall be erased. The WP_ERASE_SKIP status bit in the status register shall be set.

As described above for block write, the card will indicate that an erase is in progress by holding DAT0 low. The actual erase time may be quite long, and the host may issue CMD7 to deselect the card.

30.6.7. Bus width selection

After the host has verified the functional pins on the bus it should change the bus width configuration.

For MMC, using the SWITCH command (CMD6).The bus width configuration is changed by writing to the BUS_WIDTH byte in the Modes Segment of the EXT_CSD register (using the SWITCH command to do so). After power-on or software reset, the contents of the BUS_WIDTH byte is 0x00. If the host tries to write an invalid value, the BUS_WIDTH byte is not changed and the SWITCH_ERROR bit is set. This register is write only.

For SD memory, using SET_BUS_WIDTH command (ACMD6) to change the bus width. The default bus width after power up or GO_IDLE_STATE command (CMD0) is 1 bit. SET_BUS_WIDTH (ACMD6) is only valid in a transfer state, which means that the bus width can be changed only after a card is selected by SELECT/DESELECT_CARD (CMD7).

30.6.8. Protection management

In order to allow the host to protect data against erase or write, three methods for the cards are supported in the card:

CSD register for card protection (optional)

The entire card may be write protected by setting the permanent or temporary write protect bits in the CSD. Some cards support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD. It is defined in units of WP_GRP_SIZE erase groups as specified in the CSD. The SET_WRITE_PROT command sets the write protection of the addressed write protected group, and the CLR_WRITE_PROT command clears the write protection of the addressed write protected group.

The High Capacity SD Memory Card does not support Write Protection and does not respond to write protection commands (CMD28, CMD29 and CMD30).

Write protect switch on the card (SD memory and SD I/O card)

A mechanical sliding tablet on the side of the card will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open it means that the card is write protected. If the window is closed the card is not write protected.

Password card Lock/Unlock Operation

The Password Card Lock/Unlock protection is described in ***Card Lock/Unlock operation***.

30.6.9. Card Lock/Unlock operation

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size are kept in a 128-bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the basic command class (class 0), ACMD41, CMD16 and lock card command class (class 7). Thus, the host is allowed to reset, initialize, select, query for status, but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0), the card will be locked automatically after power on.

Similar to the existing CSD register write commands, the lock/unlock command is available in "transfer state" only. This means that it does not include an address argument and the card shall be selected before using it.

The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). ***Table 30-31. Lock card data structure*** describes the structure of the command data block.

Table 30-31. Lock card data structure

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved(all set to 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Password data(PWD)							
……								
PWDS_LEN+1								

ERASE: 1 Defines Forced Erase Operation. In byte 0, bit 3 will be set to 1 (all other bits shall be 0). All other bytes of this command will be ignored by the card.

LOCK/UNLOCK: 1 = Locks the card. 0 = Unlock the card (note that it is valid to set this bit together with SET_PWD but it is not allowed to set it together with CLR_PWD).

CLR_PWD: 1 = Clears PWD.

SET_PWD: 1 = Set new password to PWD.

PWDS_LEN: Defines the following password(s) length (in bytes). In case of a password change, this field includes the total password length of old and new passwords. The password length is up to 16 bytes. In case of a password change, the total length of the old password and the new password can be up to 32 bytes.

Password data: In case of setting a new password, it contains the new password. In case of a password change, it contains the old password followed by the new password.

Setting the password

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the new password. In the case that a password replacement is done, then the block size shall consider that both passwords (the old and the new one) are sent with the command.
- Send the Card Lock/Unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWDS_LEN) and the password itself. In the case that a password replacement is done, then the length value (PWDS_LEN) shall include both passwords (the old and the new one) and the password data field shall include the old password (currently used) followed by the new password. Note that the card shall handle the calculation of the new password length internally by subtracting the old password length from PWDS_LEN field.
- In the case that the sent old password is not correct (not equal in size and content), then the LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In the case that the sent old password is correct (equal in size and content), then the given new password and its size will be saved in the PWD and PWD_LEN registers, respectively.

Reset the password

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode CLR_PWD, the length (PWDS_LEN), and the password itself. If the PWD and PWD_LEN content match the sent password and its size, then the content of the PWD register is cleared and PWD_LEN is set to 0. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Locking a card

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit

password size (in bytes), and the number of bytes of the currently used password.

- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWDS_LEN) and the password itself.

If the PWD content is equal to the sent password, then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Unlocking the card

- Select a card (CMD7), if not previously selected.
- Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
- Send the card lock/unlock command with the appropriate data block size on the data line including the 16-bit CRC. The data block shall indicate the mode UNLOCK, the length (PWDS_LEN) and the password itself.

If the PWD content is equal to the sent password, then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

30.7. Specific operations

30.7.1. SD I/O specific operations

The SD I/O only card and SD I/O combo card support these specific operations:

Read Wait operation

Suspend/resume operation

Interrupts

The SD I/O supports these operations only if the SDIO_DATACTL[11] bit is set, except for read suspend that does not need specific hardware implementation.

SD I/O read wait operation

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The Read Wait operation allows a host to signal a card that is executing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SD I/O card. To determine if a card supports the Read Wait protocol, the host shall test SRW capability bit in the Card Capability byte of the CCCR. The timing for Read Wait is based on the Interrupt Period. If a card does not support the Read Wait protocol, the only means a host has to stall (not abort) data in the middle of a read multiple command is to control the SDIO_CLK. The limitation of this method is that with the clock stopped, the

host cannot issue any commands, and so cannot perform other operations during the delay time. Read Wait support is mandatory for the card to support suspend/resume. ***Figure 30-12. Read wait control by stopping SDIO_CLK*** and ***Figure 30-13. Read wait operation using SDIO_DAT[2]*** show the Read Wait mode about stop the SDIO_CLK and use SDIO_DAT[2].

Figure 30-12. Read wait control by stopping SDIO_CLK

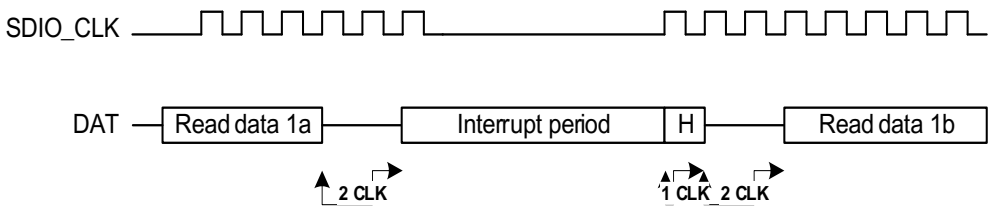
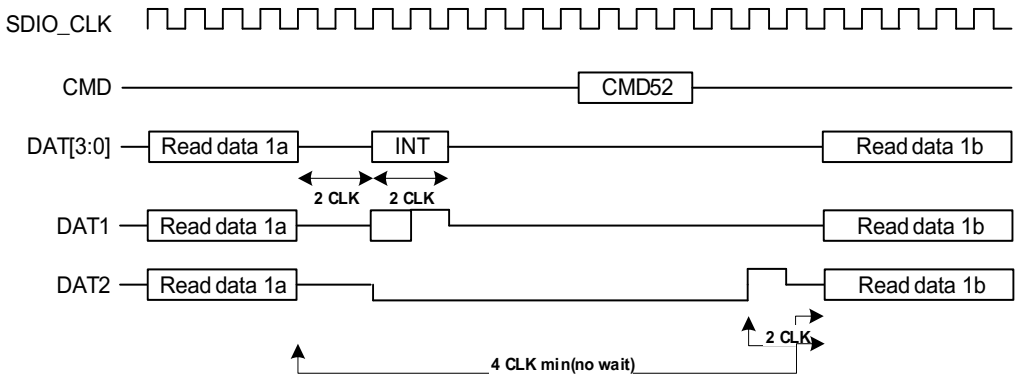


Figure 30-13. Read wait operation using SDIO_DAT[2]



We can start the Read Wait interval before the data block is received: when the data unit is enabled (SDIO_DATACTL[0] bit set), the SD I/O specific operation is enabled (SDIO_DATACTL[11] bit set), Read Wait starts (SDIO_DATACTL[10] = 0 and SDIO_DATACTL[8] = 1) and data direction is from card to SD I/O (SDIO_DATACTL[1] = 1), the DSM directly moves from Idle to Read Wait. In Read Wait the DSM drives SDIO_DAT[2] to 0 after 2 SDIO_CLK clock cycles. In this state, when you set the RWSTOP bit (SDIO_DATACTL[9]), the DSM remains in Wait for two more SDIO_CLK clock cycles to drive SDIO_DAT[2] to 1 for one clock cycle. The DSM then starts waiting again until it receives data from the card. The DSM will not start a Read Wait interval while receiving a block even if Read Wait start is set: the Read Wait interval will start after the CRC is received. The RWSTOP bit has to be cleared to start a new Read Wait operation. During the Read Wait interval, the SDIO can detect SD I/O interrupts on SDIO_DAT[1].

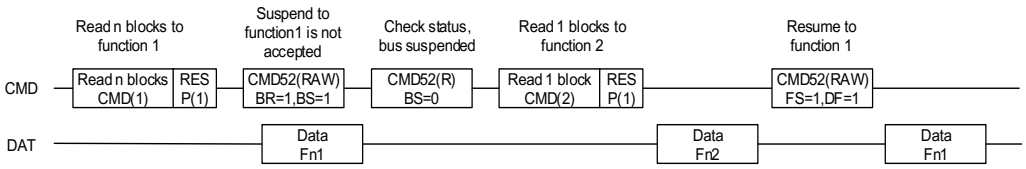
SD I/O suspend/resume operation

Within a multi-function SD I/O or a Combo card, there are multiple devices (I/O and memory) that share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SD I/O and combo cards can implement the optional concept of suspend/resume. If a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function or memory. Once this higher-priority transfer is

completed, the original transfer is re-started where it left off (resume).

Figure 30-14. Function2 read cycle inserted during function1 multiple read cycle shows a condition where the first suspend request is not immediately accepted. The host then checks the status of the request with a read and determines that the bus has now been released (BS=0). At this time, a read to function 2 is started. Once that single block read is completed, the resume is issued to function, causing the data transfer to resume (DF=1).

Figure 30-14. Function2 read cycle inserted during function1 multiple read cycle



When the host sends data to the card, the host can suspend the write operation. The SDIO_CMDCTL[11] bit is set and indicates to the CSM that the current command is a suspend command. The CSM analyzes the response and when the response is received from the card (suspend accepted), it acknowledges the DSM that goes Idle after receiving the CRC token of the current block.

To suspend a read operation, the DSM waits in the WaitR state, when the function to be suspended sends a complete packet just before stopping the data transaction. The application should continue reading receive FIFO until the FIFO is empty, and the DSM goes Idle state automatically.

Interrupts

In order to allow the SD I/O card to interrupt the host, an interrupt function is added to a pin on the SD interface. Pin number 8, which is used as SDIO_DAT[1] when operating in the 4-bit SD mode, is used to signal the card's interrupt to the host. The use of interrupt is optional for each card or function within a card. The SD I/O interrupt is "level sensitive", that is, the interrupt line shall be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period. Once the host has serviced the interrupt, it is cleared via function unique I/O operation.

When setting the SDIO_DATACTL[11] bit SD I/O interrupts can detect on the SDIO_DAT[1] line.

Figure 30-15. Read Interrupt cycle timing shows the timing of the interrupt period for single data transaction read cycles.

Figure 30-15. Read Interrupt cycle timing

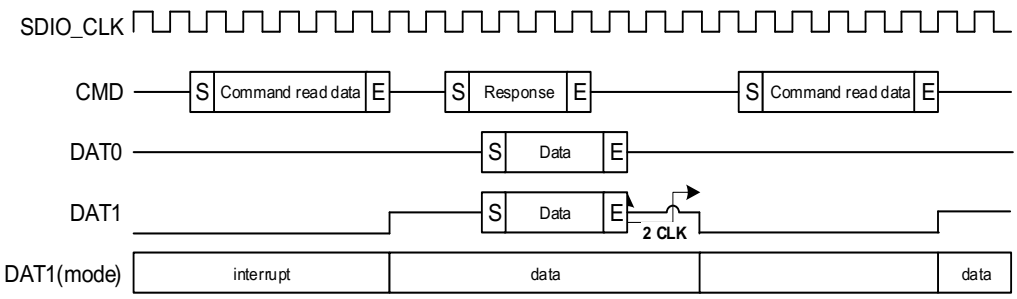
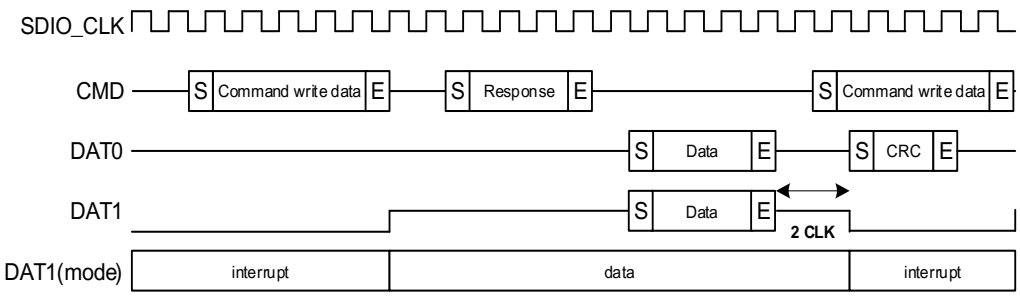


Figure 30-16. Write interrupt cycle timing



When transferring multiple blocks of data in the 4-bit SD mode, a special definition of the interrupt period is required. In order to allow the highest speed of communication, the interrupt period is limited to a 2-clock interrupt period. Card that wants to send an interrupt signal to the host shall assert DAT1 low for the first clock and high for the second clock. The card shall then release DAT1 into the hi-Z State. ***Figure 30-17. Multiple block 4-Bit read interrupt cycle timing*** shows the operation for an interrupt during a 4-bit multi-block read and ***Figure 30-18. Multiple block 4-Bit write interrupt cycle timing*** shows the operation for an interrupt during a 4-bit multi-block write.

Figure 30-17. Multiple block 4-Bit read interrupt cycle timing

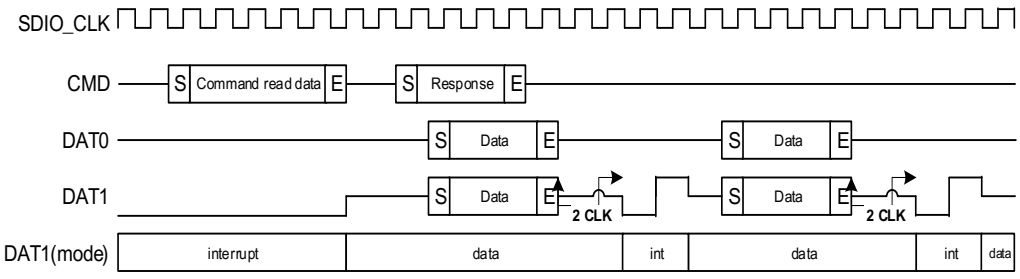
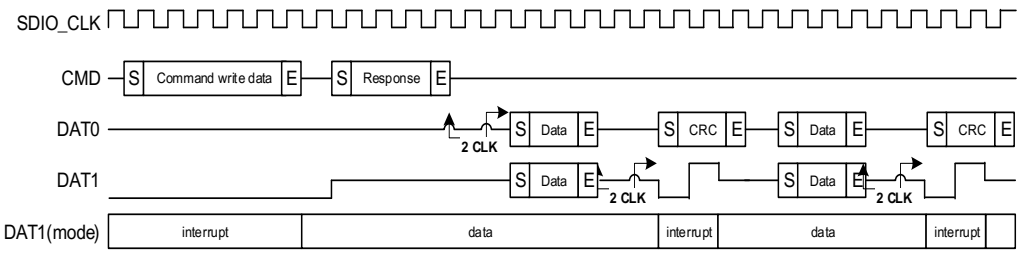


Figure 30-18. Multiple block 4-Bit write interrupt cycle timing



30.7.2. CE-ATA specific operations

The CE-ATA device supports these specific operations:

Receive command completion signal

Send command completion disable signal

The SDIO supports these operations only when SDIO_CMDCTL[14] is set.

Command completion signal

CE-ATA defines a command completion signal that the device uses to notify the host upon normal ATA command completion or when ATA command termination has occurred due to an error condition the device has encountered.

If the 'enable CMD completion' bit SDIO_CMDCTL[12] is set and the 'not interrupt Enable' bit SDIO_CMDCTL[13] is reset, the CSM waits for the command completion signal in the Waitcompl state.

When start bit is received on the CMD line, the CSM enters the Idle state. No new command can be sent for 7 bit cycles. Then, for the last 5 cycles (out of the 7) the CMD line is driven to'1' in push-pull mode.

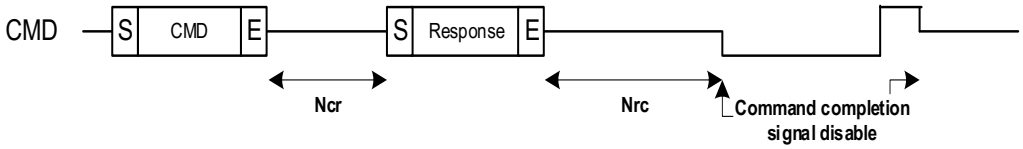
After the host detects a command completion signal from the device, it should issue a FAST_IO (CMD39) command to read the ATA Status register to determine the ending status for the ATA command.

Command completion disable signal

The host may cancel the ability for the device to return a command completion signal by issuing the command completion signal disable. The host shall only issue the command completion signal disable when it has received an R1b response for an outstanding RW_MULTIPLE_BLOCK (CMD61) command.

Command completion signal disable is sent 8 bit cycles after the reception of a short response if the 'enable CMD completion' bit, SDIO_CMDCTL[12] is not set and the 'not interrupt Enable' bit SDIO_CMDCTL[13] is reset.

Figure 30-19. The operation for command completion disable signal



30.8. SDIO registers

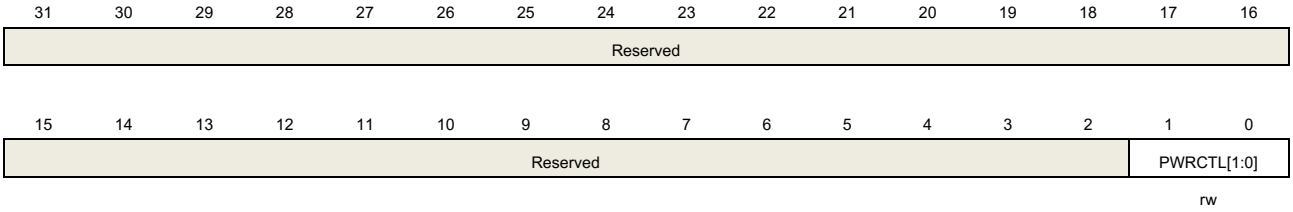
SDIO base address: 0x4001 8000

30.8.1. Power control register (SDIO_PWRCTL)

Address offset: 0x00

Reset value: 0x0000 0000

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



Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	PWRCTL[1:0]	SDIO power control bits. These bits control the SDIO state, card input or output. 00: SDIO power off: SDIO cmd/data state machine reset to IDLE, clock to card stopped, no cmd/data output to card 01: Reserved 10: Reserved 11: SDIO Power on

Note: Between Two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

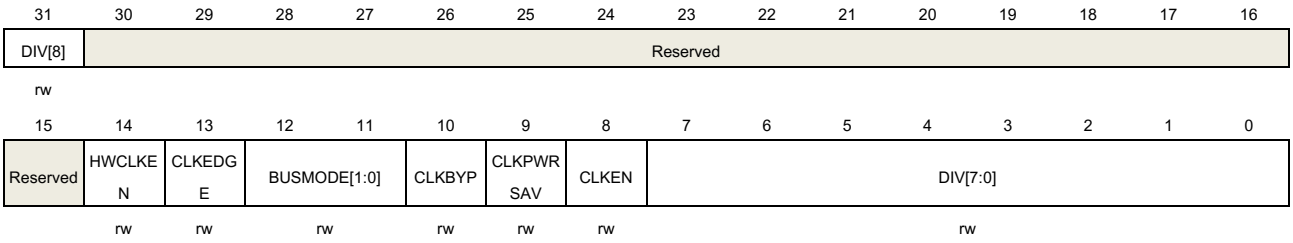
30.8.2. Clock control register (SDIO_CLKCTL)

Address offset: 0x04

Reset value: 0x0000 0000

This register controls the output clock SDIO_CLK.

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



Bits	Fields	Descriptions
31	DIV[8]	MSB of Clock division This field defines the MSB division between the input clock (SDIOCLK) and the output clock, refer to bit 7:0 of SDIO_CLKCTL.
30:15	Reserved	Must be kept at reset value.
14	HWCLKEN	Hardware Clock Control enable bit If this bit is set, hardware controls the SDIO_CLK on/off depending on the system bus is very busy or not. There is no underrun/overrun error when this bit is set, because hardware can close the SDIO_CLK when almost underrun/overrun. 0: HW Clock control is disabled 1: HW Clock control is enabled
13	CLKEDGE	SDIO_CLK clock edge selection bit 0: Select the rising edge of the SDIOCLK to generate SDIO_CLK 1: Select the falling edge of the SDIOCLK to generate SDIO_CLK
12:11	BUSMODE[1:0]	SDIO card bus mode control bit 00: 1-bit SDIO card bus mode selected 01: 4-bit SDIO card bus mode selected 10: 8-bit SDIO card bus mode selected
10	CLKBYP	Clock bypass enable bit This bit defines the SDIO_CLK is directly SDIOCLK or not. 0: NO bypass, the SDIO_CLK refers to DIV bits in SDIO_CLKCTL register. 1: Clock bypass, the SDIO_CLK is directly from SDIOCLK (SDIOCLK/1).
9	CLKPWRSAV	SDIO_CLK clock dynamic switch on/off for power saving. This bit controls SDIO_CLK clock dynamic switch on/off when the bus is idle for power saving 0: SDIO_CLK clock is always on 1: SDIO_CLK closed when bus idle
8	CLKEN	SDIO_CLK clock output enable bit 0: SDIO_CLK is disabled 1: SDIO_CLK is enabled
7:0	DIV[7:0]	Clock division This field and DIV[8] bit defines the division factor to generator SDIO_CLK clock to card. The SDIO_CLK is divider from SDIOCLK if CLKBYP bit is 0, and the SDIO_CLK frequency = SDIOCLK / (DIV[8:0] + 2).

Note: Between two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

30.8.3. Command argument register (SDIO_CMDAGMT)

Address offset: 0x08
Reset value: 0x0000 0000

This register defines 32 bit command argument, which will be used as part of the command (bit 39 to bit 8).

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

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

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

Bits	Fields	Descriptions
31:0	CMDAGMT[31:0]	SDIO card command argument This field defines the SDIO card command argument which sent to card. This field is the bits [39:8] of command message. If the command message contains an argument, this field must update before writing SDIO_CMDCTL register when sending a command.

30.8.4. Command control register (SDIO_CMDCTL)

Address offset: 0x0C
Reset value: 0x0000 0000

The SDIO_CMDCTL register contains the command index and other command control bits to control command state machine.

This register has 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	ATAEN	NINTEN	ENCMDC	SUSPEND	CSMEN	WAITDEND	INTWAIT	CMDRESP[1:0]		CMDIDX[5:0]					
	rw	rw	rw	rw	rw	rw	rw	rw		rw					

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	ATAEN	CE-ATA command enable(CE-ATA only) If this bit is set, the host enters the CE-ATA mode, and the CSM transfers CMD61. 0: CE-ATA disable 1: CE-ATA enable

13	NINTEN	No CE-ATA Interrupt (CE-ATA only) This bit defines if there is CE-ATA interrupt or not. This bit is only used when CE-ATA card. 0: CE-ATA interrupt enable 1: CE_ATA interrupt disable
12	ENCMDC	CMD completion signal enabled (CE-ATA only) This bit defines if there is command completion signal or not in CE-ATA card. 0: no completion signal 1: have completion signal
11	SUSPEND	SD I/O suspend command(SD I/O only) This bit defines whether the CSM to send a suspend command or not. This bit is only used for SDIO card. 0: no effect 1: suspend command
10	CSMEN	Command state machine (CSM) enable bit 0: Command state machine disable (stay on CS_Idle) 1: Command state machine enable
9	WAITDEND	Waits for ends of data transfer. If this bit is set, the command state machine starts to send a command must wait the end of data transfer. 0: no effect 1: Wait the end of data transfer
8	INTWAIT	Interrupt wait instead of timeout This bit defines the command state machine to wait card interrupt at CS_Wait state in command state machine. If this bit is set, no command wait timeout generated. 0: Not wait interrupt. 1: Wait interrupt.
7:6	CMDRESP[1:0]	Command response type bits These bits define the response type after sending a command message. 00: No response 01: Short response 10: No response 11: Long response
5:0	CMDIDX[5:0]	Command index This field defines the command index to be sent to SDIO card.

Note: Between Two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

30.8.5. Command index response register (SDIO_RSPCMDIDX)

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										RSPCMDIDX[5:0]					
										r					

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	RSPCMDIDX[5:0]	Last response command index Read-only bits field. This field contains the command index of the last command response received. If the response doesn't have the command index (long response and short response of R3), the content of this register is undefined.

30.8.6. Response register (SDIO_RESPx x=0..3)

Address offset: 0x14+(4*x), x=0..3
Reset value: 0x0000 0000

These registers contain the content of the last card response received.

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

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

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

Bits	Fields	Descriptions
31:0	RESPx[31:0]	Card state. The content of the response, see ***Table 30-32. SDIO_RESPx register at different response type***.

The short response is 32 bits, the long response is 127 bits (bit 128 is the end bit 0).

Table 30-32. SDIO_RESPx register at different response type

Register	Short response	Long response
SDIO_RESP0	Card response[31:0]	Card response[127:96]
SDIO_RESP1	reserved	Card response [95:64]
SDIO_RESP2	reserved	Card response [63:32]

Register	Short response	Long response
SDIO_RESP3	reserved	Card response [31:1],plus bit 0

30.8.7. Data timeout register (SDIO_DATATO)

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
DATATO[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	DATATO[31:0]	Data timeout period These bits define the data timeout period count by SDIO_CLK. When the DSM enter the state WaitR or BUSY, the internal counter which loads from this register starts decrement. The DSM timeout and enter the state Idle and set the DTTMOUT flag when the counter decreases to 0.

Note: The data timer register and the data length register must be updated before being written to the data control register when need a data transfer.

30.8.8. Data length register (SDIO_DATALEN)

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							DATALEN[24:16]								
							rw								

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

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24:0	DATALEN[24:0]	Data transfer length This register defined the number of bytes to be transferred. When the data transfer starts, the data counter loads this register and starts decrement.

Note: If block data transfer selected, the content of this register must be a multiple of the

block size (refer to SDIO_DATACTL). The data timer register and the data length register must be updated before being written to the data control register when need a data transfer.

30.8.9. Data control register (SDIO_DATACTL)

Address offset: 0x2C
Reset value: 0x0000 0000

This register controls the DSM.

This register has 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				IOEN	RWTYPE	RWSTOP	RWEN	BLKSZ[3:0]				DMAEN	TRANSMOD	DATADIR	DATAEN
				rw	rw	rw	rw	rw				rw	rw	rw	rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	IOEN	SD I/O specific function enable(SD I/O only) 0: Not SD I/O specific function 1: SD I/O specific function
10	RWTYPE	Read wait type(SD I/O only) 0: Read Wait control using SDIO_DAT[2] 1: Read Wait control by stopping SDIO_CLK
9	RWSTOP	Read wait stop(SD I/O only) 0: No effect 1: Stop the read wait process if RWEN bit is set
8	RWEN	Read wait mode enabled(SD I/O only) 0: Read wait mode is disabled 1: Read wait mode is enabled
7:4	BLKSZ[3:0]	Data block size These bits defined the block size when data transfer is block transfer. 0000: block size = 2^0 = 1 byte 0001: block size = 2^1 = 2 bytes 0010: block size = 2^2 = 4 bytes 0011: block size = 2^3 = 8 bytes 0100: block size = 2^4 = 16 bytes 0101: block size = 2^5 = 32 bytes 0110: block size = 2^6 = 64 bytes 0111: block size = 2^7 = 128 bytes

		1000: block size = 2^8 = 256 bytes 1001: block size = 2^9 = 512 bytes 1010: block size = 2^{10} = 1024 bytes 1011: block size = 2^{11} = 2048 bytes 1100: block size = 2^{12} = 4096 bytes 1101: block size = 2^{13} = 8192 bytes 1110: block size = 2^{14} = 16384 bytes 1111: reserved
3	DMAEN	DMA enable bit 0: DMA is disabled. 1: DMA is enabled.
2	TRANSMOD	Data transfer mode 0: Block transfer 1: Stream transfer or SDIO multibyte transfer
1	DATADIR	Data transfer direction 0: Write data to card. 1: Read data from card.
0	DATAEN	Data transfer enable bit Write 1 to this bit to start data transfer regardless this bit is 0 or 1. The DSM moves to Readwait state if RWEN is set or to the WaitS, WaitR state depend on DATADIR bit. Start a new data transfer, it not need to clear this bit to 0.

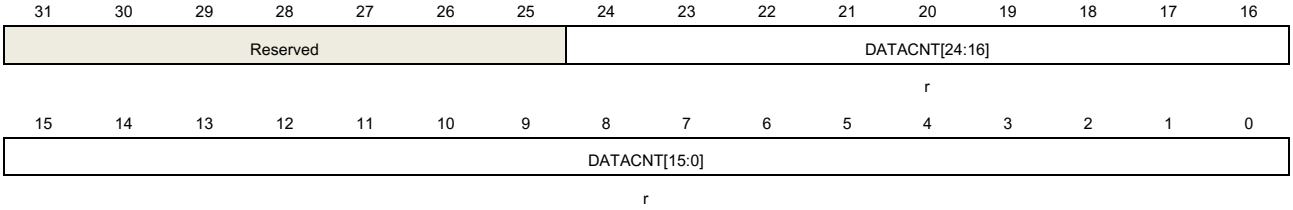
Note: Between Two write accesses to this register, it needs at least 3 SDIOCLK + 2 pclk2 which used to sync the registers to SDIOCLK clock domain.

30.8.10. Data counter register (SDIO_DATACNT)

Address offset: 0x30
Reset value: 0x0000 0000

This register is read only. When the DSM from Idle to WaitR or WaitS, it loads value from data length register (SDIO_DATALEN). It decrements with the data transferred, when it reaches 0, the flag DTEND is set.

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



Bits	Fields	Descriptions

31:25	Reserved	Must be kept at reset value.
24:0	DATACNT[24:0]	Data count value Read-only bits field. When these bits are read, the number of remaining data bytes to be transferred is returned.

30.8.11. Status register (SDIO_STAT)

Address offset: 0x34
Reset value: 0x0000 0000

This register is read only. The following descripts the types of flag:

The flags of bit [23:22, 10:0] can only be cleared by writing 1 to the corresponding bit in interrupt clear register (SDIO_INTC).

The flags of bit [21:11] are changing depend on the hardware logic.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ATAEND	SDIOINT	RXDTVAL	TXDTVAL	RFE	TFE	RFF	TFF
								r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFH	TFH	RXRUN	TXRUN	CMDRUN	DTBLKEND	STBITE	DTEND	CMDSEND	CMDRECV	RXORE	TXURE	DTTMOUT	CMDTMOUT	DTCRCERR	CCRCERR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	ATAEND	CE-ATA command completion signal received (only for CMD61)
22	SDIOINT	SD I/O interrupt received
21	RXDTVAL	Data is valid in receive FIFO
20	TXDTVAL	Data is valid in transmit FIFO
19	RFE	Receive FIFO is empty
18	TFE	Transmit FIFO is empty When HW Flow control is enabled, TFE signals becomes activated when the FIFO contains 2 words.
17	RFF	Receive FIFO is full When HW Flow control is enabled, RFF signals becomes activated 2 words before the FIFO is full.
16	TFF	Transmit FIFO is full

15	RFH	Receive FIFO is half full: at least 8 words can be read in the FIFO
14	TFH	Transmit FIFO is half empty: at least 8 words can be written into the FIFO
13	RXRUN	Data reception in progress
12	TXRUN	Data transmission in progress
11	CMDRUN	Command transmission in progress
10	DTBLKEND	Data block sent/received (CRC check passed)
9	STBITE	Start bit error in the bus.
8	DTEND	Data end (data counter, SDIO_DATACNT, is zero)
7	CMDSEND	Command sent (no response required)
6	CMDRECV	Command response received (CRC check passed)
5	RXORE	Received FIFO overrun error occurs
4	TXURE	Transmit FIFO underrun error occurs
3	DTTMOUT	Data timeout The data timeout period depends on the SDIO_DATATO register.
2	CMDTMOUT	Command response timeout The command timeout period has a fixed value of 64 SDIO_CLK clock periods.
1	DTCRCERR	Data block sent/received (CRC check failed)
0	CCRCERR	Command response received (CRC check failed)

30.8.12. Interrupt clear register (SDIO_INTC)

Address offset: 0x38

Reset value: 0x0000 0000

This register is write only. Writing 1 to the bit can clear the corresponding bit in the SDIO_STAT register.

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DTBLKENDC	STBITEC	DTENDC	CMDSENDC	CMDRECVC	RXOREC	TXUREC	DTTMOUTC	CMDTMOUTC	DTCRCERRC	CCRCERRC
					w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions

31:24	Reserved	Must be kept at reset value.
23	ATAENDC	ATAEND flag clear bit Write 1 to this bit to clear the flag.
22	SDIOINTC	SDIOINT flag clear bit Write 1 to this bit to clear the flag.
21:11	Reserved	Must be kept at reset value.
10	DTBLKENDC	DTBLKEND flag clear bit Write 1 to this bit to clear the flag.
9	STBITEC	STBITE flag clear bit Write 1 to this bit to clear the flag.
8	DTENDC	DTEND flag clear bit Write 1 to this bit to clear the flag.
7	CMDSENDC	CMDSEND flag clear bit Write 1 to this bit to clear the flag.
6	CMDRECVC	CMDRECV flag clear bit Write 1 to this bit to clear the flag.
5	RXOREC	RXORE flag clear bit Write 1 to this bit to clear the flag.
4	TXUREC	TXURE flag clear bit Write 1 to this bit to clear the flag.
3	DTTMOUTC	DTTMOUT flag clear bit Write 1 to this bit to clear the flag.
2	CMDTMOUTC	CMDTMOUT flag clear bit Write 1 to this bit to clear the flag.
1	DTCRCERRC	DTCRCERR flag clear bit Write 1 to this bit to clear the flag.
0	CCRCERRC	CCRCERR flag clear bit Write 1 to this bit to clear the flag.

30.8.13. Interrupt enable register (SDIO_INTEN)

Address offset: 0x3C
Reset value: 0x0000 0000

This register enables the corresponding interrupt in the SDIO_STAT register.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ATAENDIE	SDIOINTIE	RXDTVALIE	TXDTVALIE	RFEIE	TFEIE	RFFIE	TFFIE
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFHIE	TFHIE	RXRUNIE	TXRUNIE	CMDRUNIE	DTBLKENDIE	STBITEIE	DTENDIE	CMDSENDIE	CMDRECVIE	RXOREIE	TXUREIE	DTTMOUTIE	CMDTMOUTIE	DTCRCERRIE	CCRCERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	ATAENDIE	CE-ATA command completion signal received interrupt enable 0: CE-ATA command completion signal received interrupt is disabled. 1: CE-ATA command completion signal received interrupt is enabled.
22	SDIOINTIE	SD I/O interrupt received interrupt enable 0: SD I/O interrupt received interrupt is disabled. 1: SD I/O interrupt received interrupt is enabled.
21	RXDTVALIE	Data valid in receive FIFO interrupt enable 0: Data valid in receive FIFO interrupt is disabled. 1: Data valid in receive FIFO interrupt is enabled.
20	TXDTVALIE	Data valid in transmit FIFO interrupt enable 0: Data valid in transmit FIFO interrupt is disabled. 1: Data valid in transmit FIFO interrupt is enabled.
19	RFEIE	Receive FIFO empty interrupt enable 0: Receive FIFO empty interrupt is disabled. 1: Receive FIFO empty interrupt is enabled.
18	TFEIE	Transmit FIFO empty interrupt enable 0: Transmit FIFO empty interrupt is disabled. 1: Transmit FIFO empty interrupt is enabled.
17	RFFIE	Receive FIFO full interrupt enable 0: Receive FIFO full interrupt is disabled. 1: Receive FIFO full interrupt is enabled.
16	TFFIE	Transmit FIFO full interrupt enable 0: Transmit FIFO full interrupt is disabled. 1: Transmit FIFO full interrupt is enabled.
15	RFHIE	Receive FIFO half full interrupt enable 0: Receive FIFO half full interrupt is disabled. 1: Receive FIFO half full interrupt is enabled.
14	TFHIE	Transmit FIFO half empty interrupt enable 0: Transmit FIFO half empty interrupt is disabled.

		1: Transmit FIFO half empty interrupt is enabled.
13	RXRUNIE	Data reception interrupt enable 0: Data reception interrupt is disabled. 1: Data reception interrupt is enabled.
12	TXRUNIE	Data transmission interrupt enable 0: Data transmission interrupt is disabled. 1: Data transmission interrupt is enabled.
11	CMDRUNIE	Command transmission interrupt enable 0: Command transmission interrupt is disabled. 1: Command transmission interrupt is enabled.
10	DTBLKENDIE	Data block end interrupt enable 0: Data block end interrupt is disabled. 1: Data block end interrupt is enabled.
9	STBITEIE	Start bit error interrupt enable 0: Start bit error interrupt is disabled. 1: Start bit error interrupt is enabled.
8	DTENDIE	Data end interrupt enable 0: Data end interrupt is disabled. 1: Data end interrupt is enabled.
7	CMDSENDIE	Command sent interrupt enable 0: Command sent interrupt is disabled. 1: Command sent interrupt is enabled.
6	CMDRECVIE	Command response received interrupt enable 0: Command response received interrupt is disabled. 1: Command response received interrupt is enabled.
5	RXOREIE	Received FIFO overrun error interrupt enable 0: Received FIFO overrun error interrupt is disabled. 1: Received FIFO overrun error interrupt is enabled.
4	TXUREIE	Transmit FIFO underrun error interrupt enable 0: Transmit FIFO underrun error interrupt is disabled. 1: Transmit FIFO underrun error interrupt is enabled.
3	DTTMOUTIE	Data timeout interrupt enable 0: Data timeout interrupt is disabled. 1: Data timeout interrupt is enabled.
2	CMDTMOUTIE	Command response timeout interrupt enable 0: Command response timeout interrupt is disabled. 1: Command response timeout interrupt is enabled.

1	DTCRCERRIE	Data CRC fail interrupt enable 0: Data CRC fail interrupt is disabled. 1: Data CRC fail interrupt is enabled.
0	CCRCERRIE	Command response CRC fail interrupt enable 0: Command response CRC fail interrupt is disabled. 1: Command response CRC fail interrupt is enabled.

30.8.14. FIFO counter register (SDIO_FIFOCNT)

Address offset: 0x48
Reset value: 0x0000 0000

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

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

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

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	FIFOCNT[23:0]	FIFO counter. These bits define the remaining number words to be written or read from the FIFO. It loads the data length register (SDIO_DATALEN[24:2] if SDIO_DATALEN is word-aligned or SDIO_DATALEN[24:2]+1 if SDIO_DATALEN is not word-aligned) when DATAEN is set, and start count decrement when a word write to or read from the FIFO.

30.8.15. FIFO data register (SDIO_FIFO)

Address offset: 0x80
Reset value: 0x0000 0000

This register occupies 32 entries of 32-bit words, the address offset is from 0x80 to 0xFC.

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

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

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

Bits	Fields	Descriptions

31:0	FIFODT[31:0]	Receive FIFO data or transmit FIFO data These bits are the data of receive FIFO or transmit FIFO. Write to or read from this register is write data to FIFO or read data from FIFO.

31. Universal serial bus full-speed interface (USBFS)

31.1. Overview

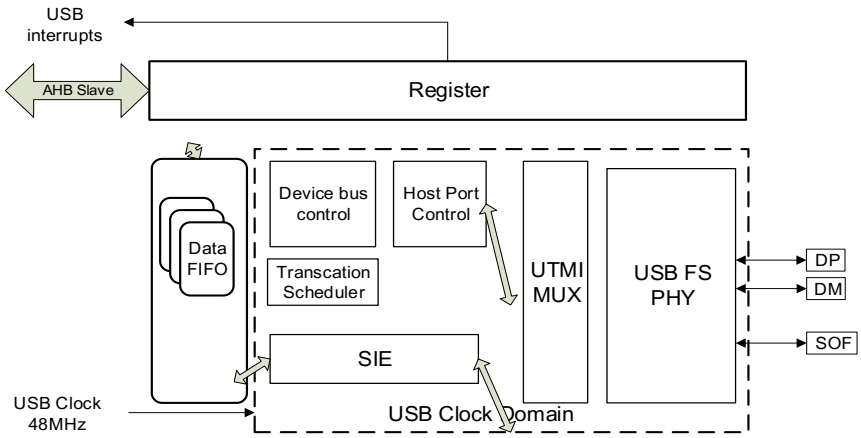
The USB Full-Speed (USBFS) controller provides a USB-connection solution for portable devices. USBFS supports host and device modes. USBFS contains a full-speed internal USB PHY and external PHY chip is not contained. USBFS supports all the four types of transfer (control, bulk, Interrupt and isochronous) which defined in USB 2.0 protocol.

31.2. 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 12 channels in host mode.
- Includes 2 transmit FIFOs (periodic and non-periodic) and a receive FIFO (shared by all channels) in host mode.
- Includes 6 transmit FIFOs (one for each IN endpoint) and a receive FIFO (shared by all OUT endpoints) in device mode.
- Supports 6 OUT and 6 IN endpoints in device mode.
- Supports remote wakeup in device mode.
- Includes a Full-Speed USB PHY.
- Time intervals of SOFs are dynamic adjustable in host mode.
- SOF pulse supports output to PAD.
- Needs external component to supply power for connected USB device in host mode.

31.3. Block diagram

Figure 31-1. USBFS block diagram



31.4. Signal description

Table 31-1. 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

31.5. Function overview

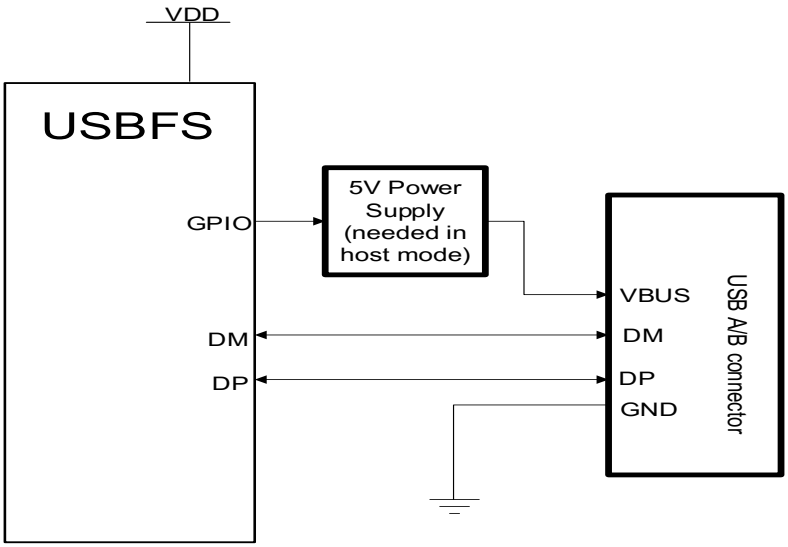
31.5.1. USBFS clocks and working modes

USBFS can operate as a host or a 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 31-2. Connection with host or device mode***

Figure 31-2. Connection with host or device mode



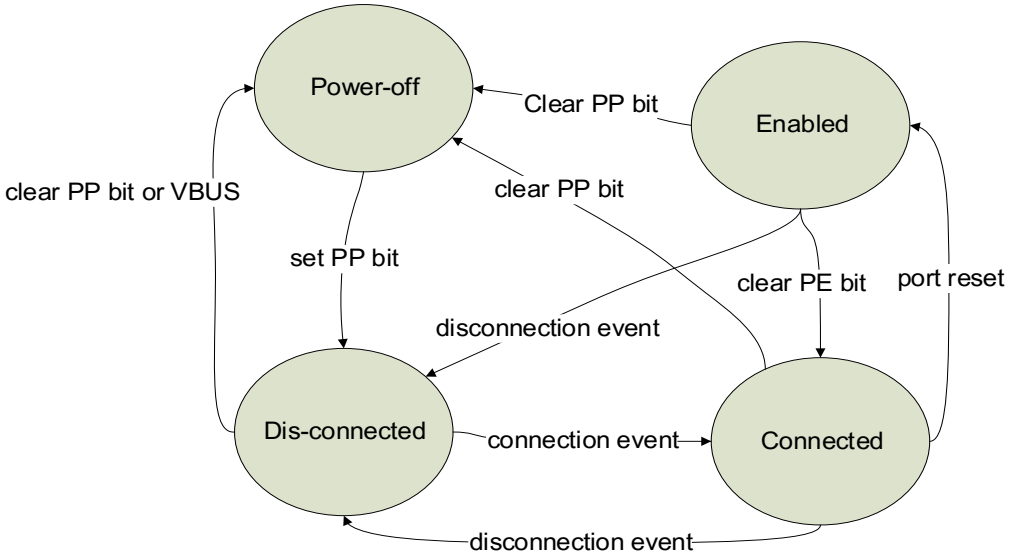
When USBFS works in host mode (FHM bit is set and FDM bit is cleared), the VBUS is 5V power detecting pin used for voltage detection defined in USB protocol. The internal PHY cannot supply 5V VBUS power. So if application needs VBUS power, an external power supply IC is needed.

31.5.2. 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 31-3. 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 12 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.

31.5.3. 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.

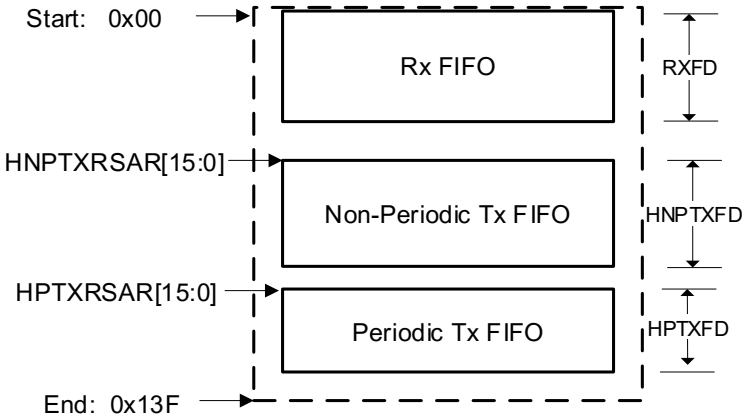
31.5.4. Data FIFO

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

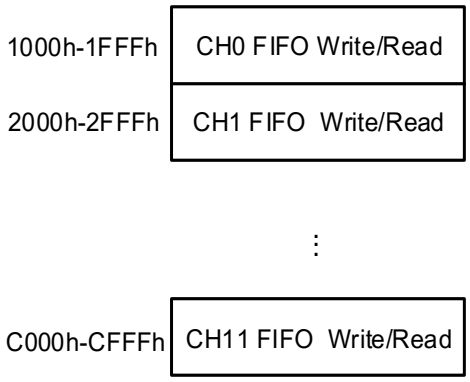
In host mode, the data FIFO space is divided into 3 parts: Rx FIFO for received packet, Non-Periodic Tx FIFO for non-period transmission packet and Periodic Tx FIFO for periodic transmission packet. All IN channels shares the Rx FIFO for packets reception. All the periodic OUT channels share the periodic 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 31-4. 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 31-4. HOST mode FIFO space in SRAM



USBFS provides a special register area for the internal data FIFO reading and writing. ***Figure 31-3. 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.

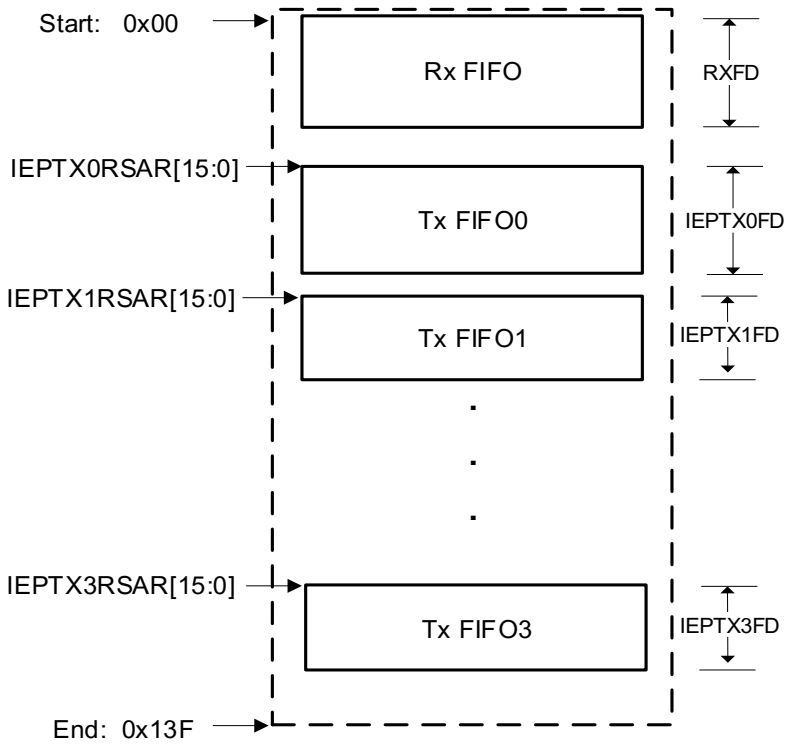
Figure 31-3. Host mode FIFO access register map



Device mode

In device mode, the data FIFO is divided into several parts: one Rx FIFO, and 6 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…5) registers. ***Figure 31-4. 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.

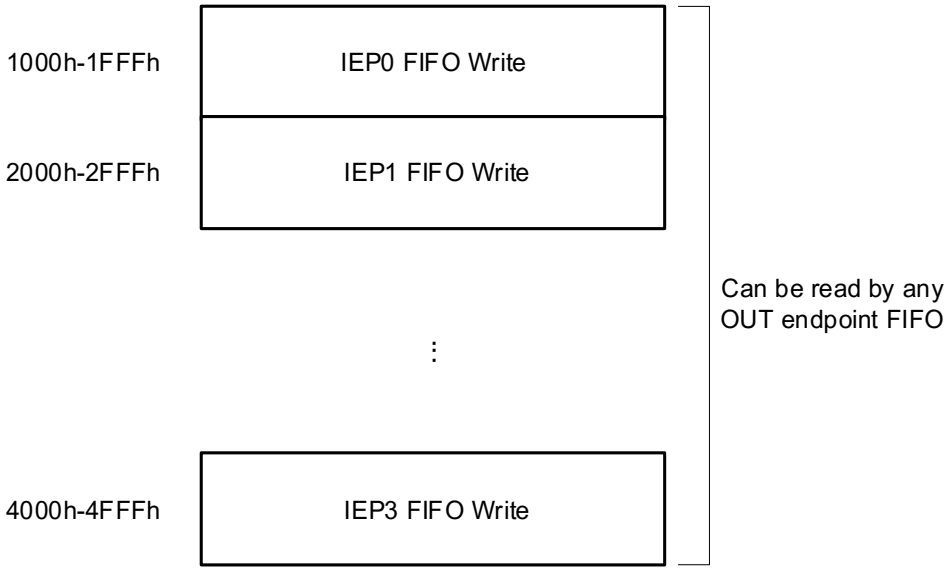
Figure 31-4. Device mode FIFO space in SRAM



USBFS provides a special register area for the internal data FIFO reading and writing. ***Figure 31-5. 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 31-5. Device mode FIFO access register map



31.5.5. 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.

Note: Before starting to write the transmit FIFO, it is necessary to ensure that there is remaining space in the transmit FIFO and the request queue.

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.

31.6. 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 31-2. USBFS global interrupt*.

Table 31-2. USBFS global interrupt

Interrupt Flag	Description	Operation 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.

31.7. Register definition

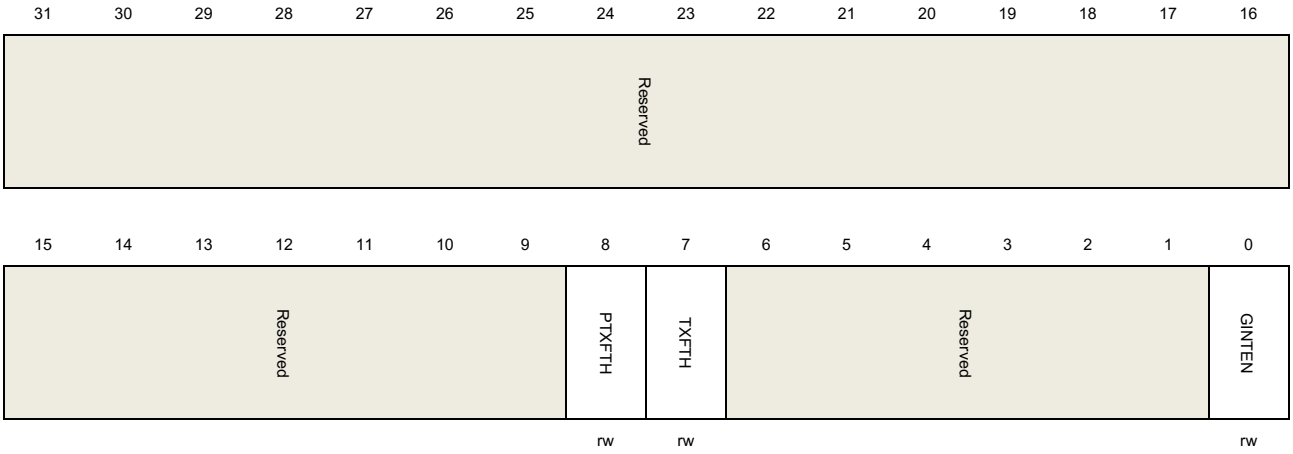
USBFS base address: 0x5000 0000

31.7.1. 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.

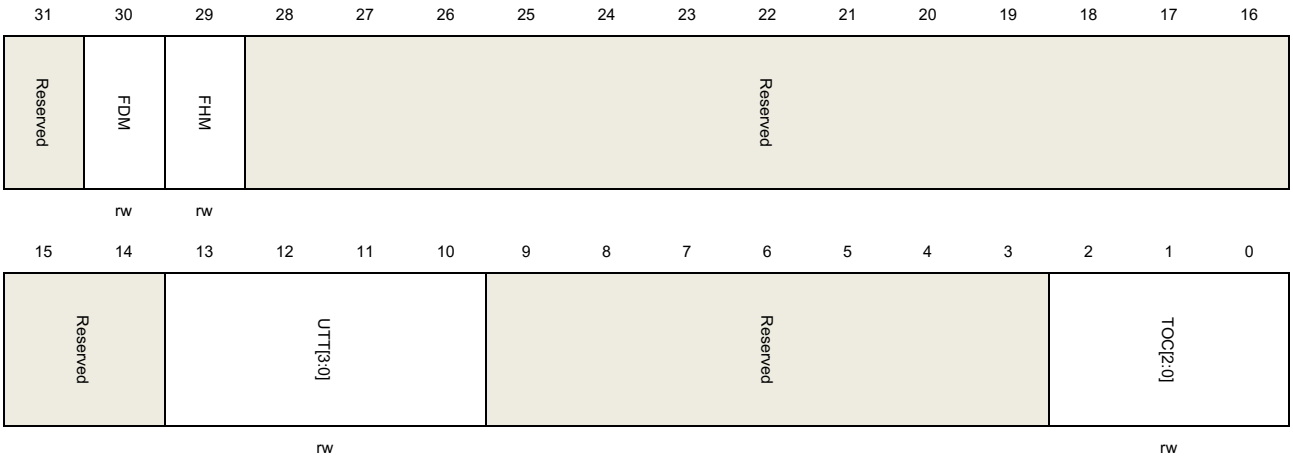
Note: Accessible in both device and host modes.

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.

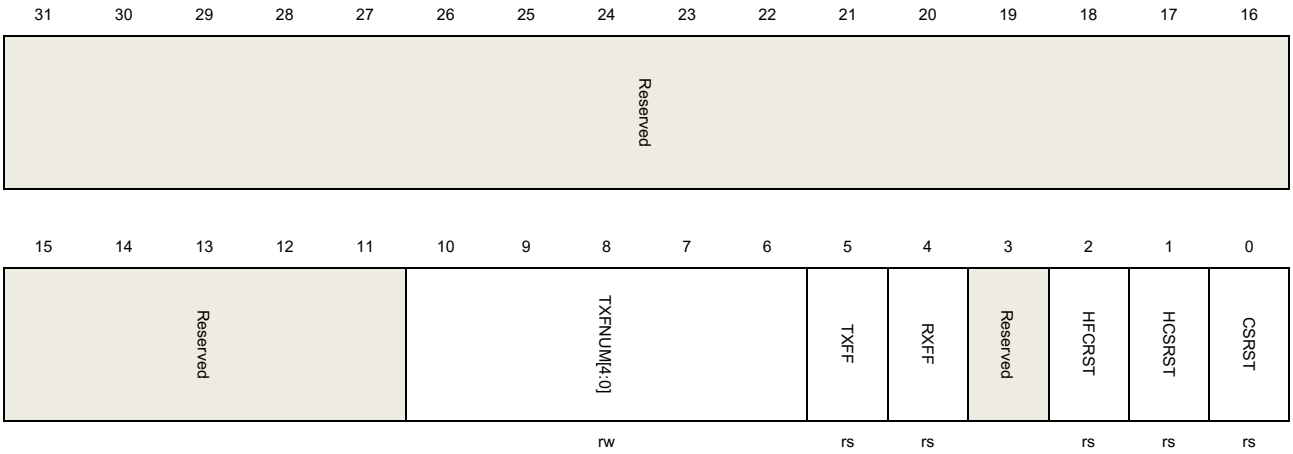
13:10	UTT[3:0]	USB turnaround time Turnaround time in PHY clocks. **Note:** Only accessible in device mode.
9:3	Reserved	Must be kept at reset value.
2:0	TOC[2:0]	Timeout calibration USBFS always uses time-out value required in USB 2.0 when waiting for a packet. Application may use TOC [2:0] to add the 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 …

		00101: Only Tx FIFO5 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 1. 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: 0x0000 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.

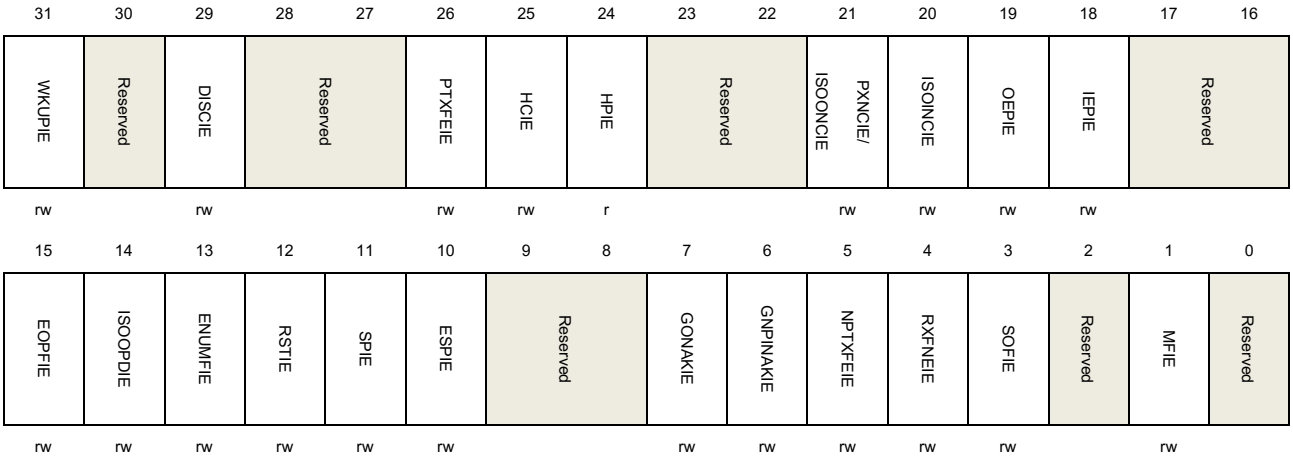
Global interrupt enable register (USBFS_GINTEN)

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)

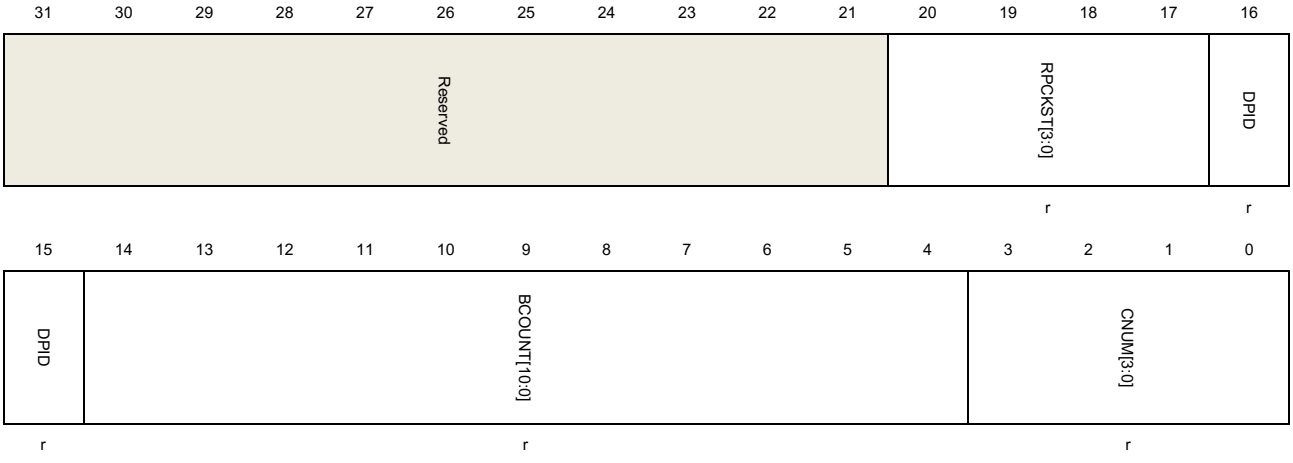
Address offset for Read: 0x001C
Address offset for Pop: 0x0020
Reset value: 0x0000 0000

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)

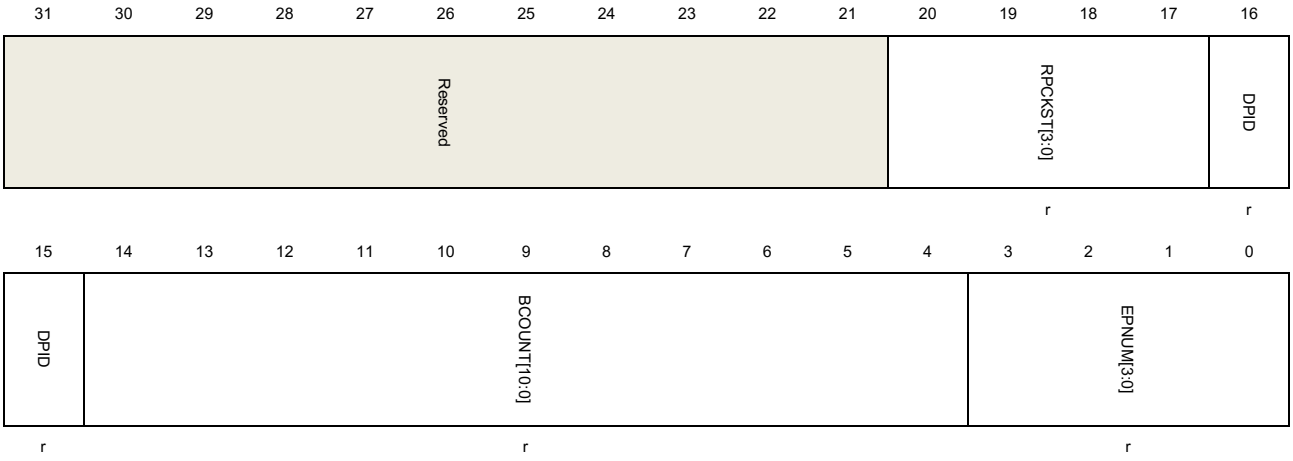
Host mode:



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

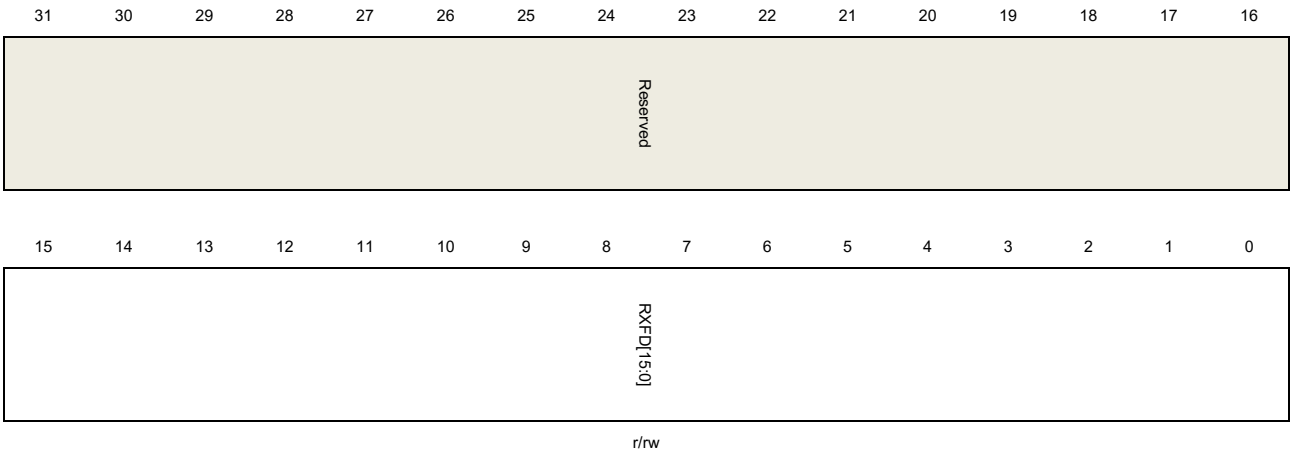
		The byte count of the received data packet.
3:0	EPNUM[3:0]	Endpoint number The endpoint number to which the current received packet belongs.

Global receive FIFO length register (USBFS_GRFLEN)

Address offset: 0x024

Reset value: 0x0000 0080

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. 1<=RXFD<=1024

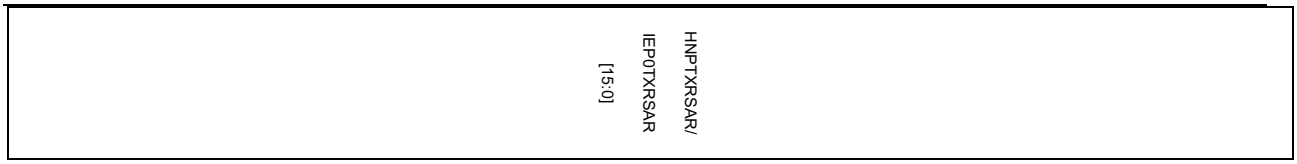
Host non-periodic transmit FIFO length register / Device IN endpoint 0 transmit FIFO length (USBFS_HNPTFLEN _DIEP0TFLEN)

Address offset: 0x028

Reset value: 0x0080 0080

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. 1<=HNPTXFD<=1024
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. 16<=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.

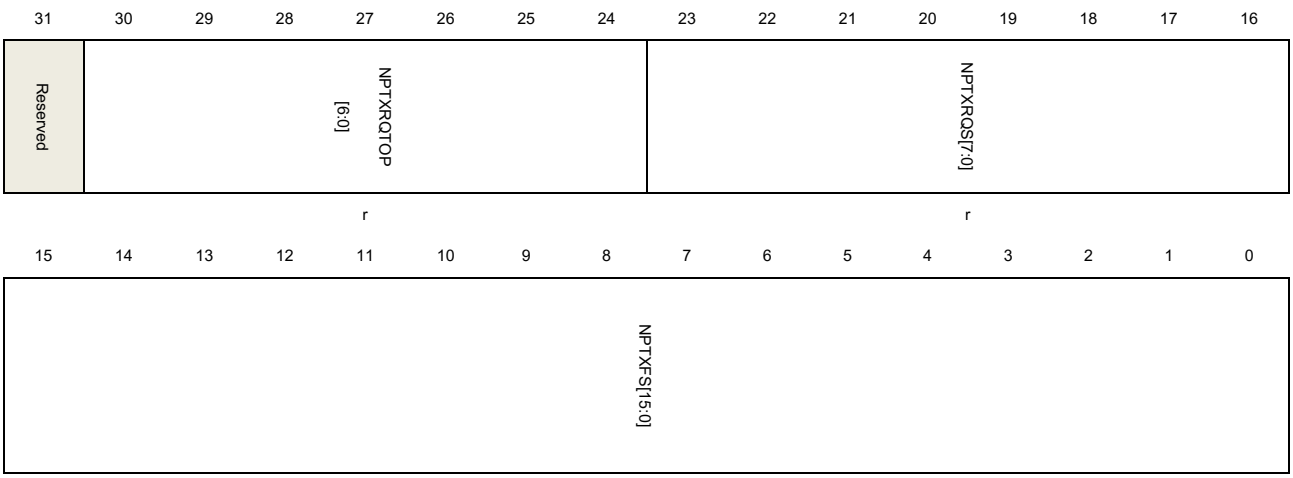
Host non-periodic transmit FIFO/queue status register (USBFS_HNPTFQSTAT)

Address offset: 0x002C
Reset value: 0x0008 0080

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)



r

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

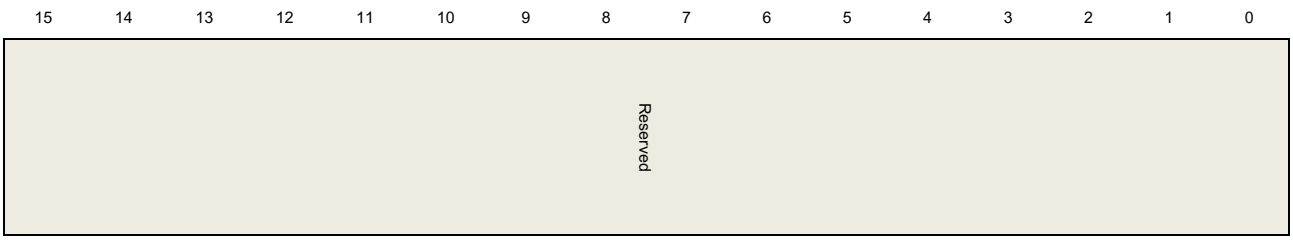
Global core configuration register (USBFS_GCCFG)

Address offset: 0x0038

Reset value: 0x000C 0000

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





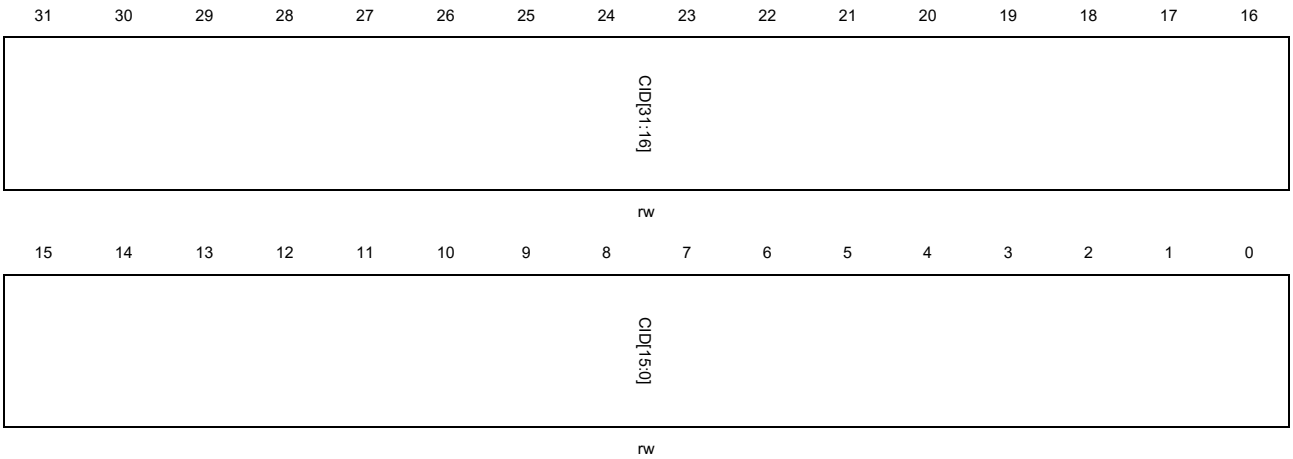
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	SOFOEN	SOF output enable 0: SOF pulse output disabled. 1: SOF pulse output enabled.
19:17	Reserved	Must be kept at reset value.
16	PWRON	Power on This bit is the power switch for the internal embedded Full-Speed PHY. 0: Embedded 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)



Bits	Fields	Descriptions

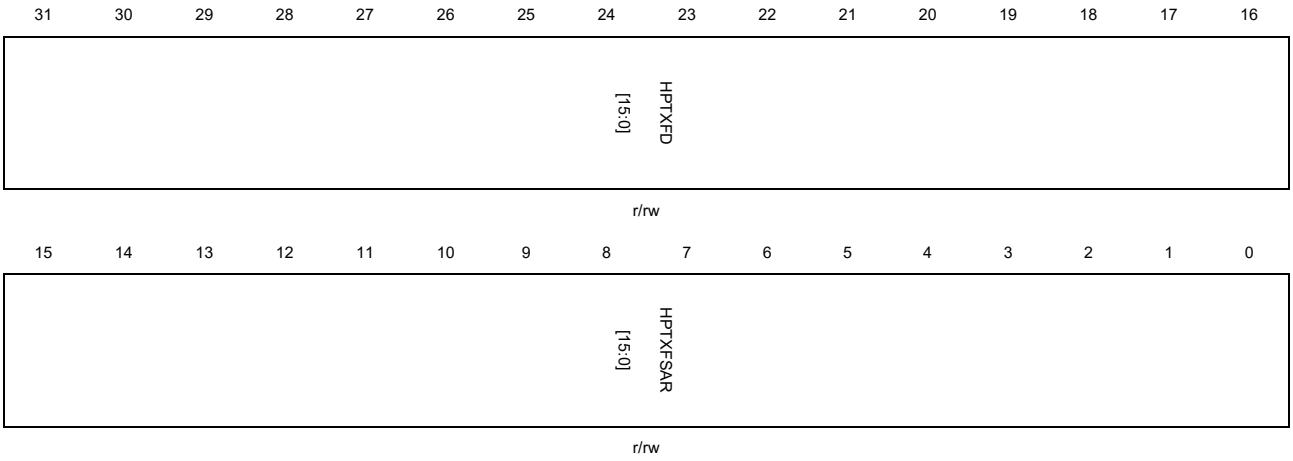
31:0	CID[31:0]	Core ID Software can write or read this field and uses this field as a unique ID for its application

Host periodic transmit FIFO length register (USBFS_HPTFLEN)

Address offset: 0x0100

Reset value: 0x0000 0000

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



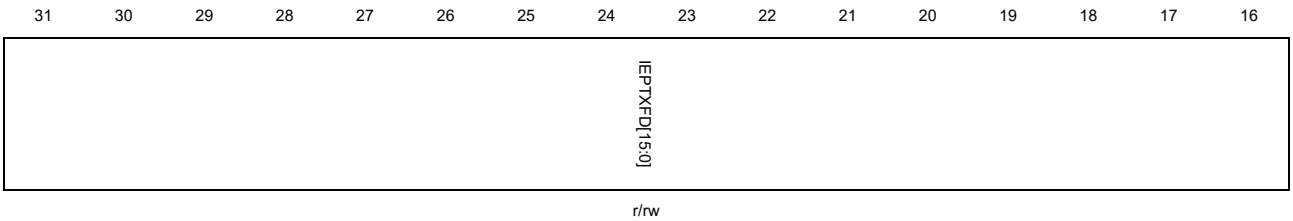
Bits	Fields	Descriptions
31:16	HPTXFD[15:0]	Host Periodic Tx FIFO depth In terms of 32-bit words. 1<=HPTXFD<=1024
15:0	HPTXFSAR[15:0]	Host periodic Tx FIFO RAM start address The start address for host periodic transmit FIFO RAM is in term of 32-bit words.

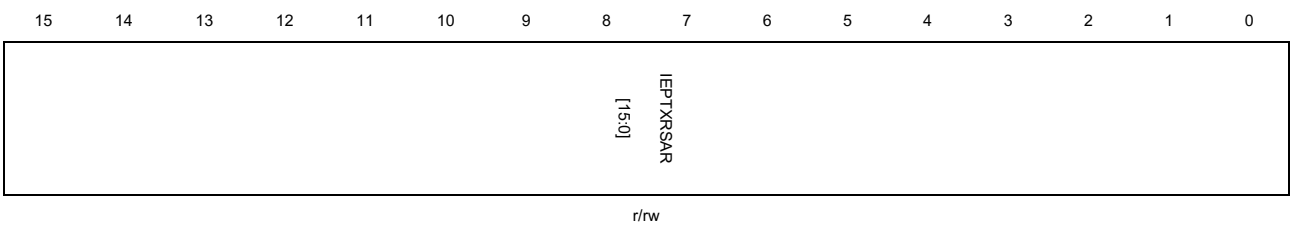
Device IN endpoint transmit FIFO length register (USBFS_DIEPxTFLEN) (x = 1…5, where x is the FIFO_number)

Address offset: 0x0104 + (FIFO_number - 1) × 0x04

Reset value: 0x0000 0000

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.

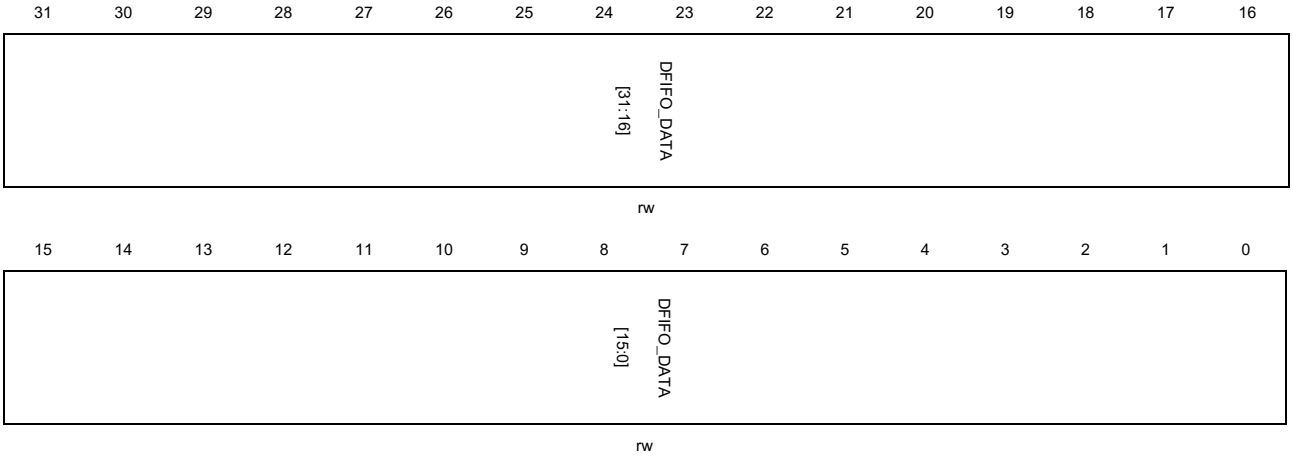
Data FIFO register (USBFS_DFIFO)

Address offset: 0x1000

Host mode: write address range: [0x1000,0x8FFF], read address range: [0x1000, 0xFFFF]

Device mode: write address range: [0x1000,0x4FFF], read address range: [0x1000, 0xFFFF]

Reset value: 0x0000 0000



Bits	Fields	Descriptions
31:0	DFIFO_DATA[31:0]	Write the field would push the data into corresponding data FIFO， and read the field would pop the data out corresponding RX FIFO, if read the overflowing address, the last pop data would be gained.

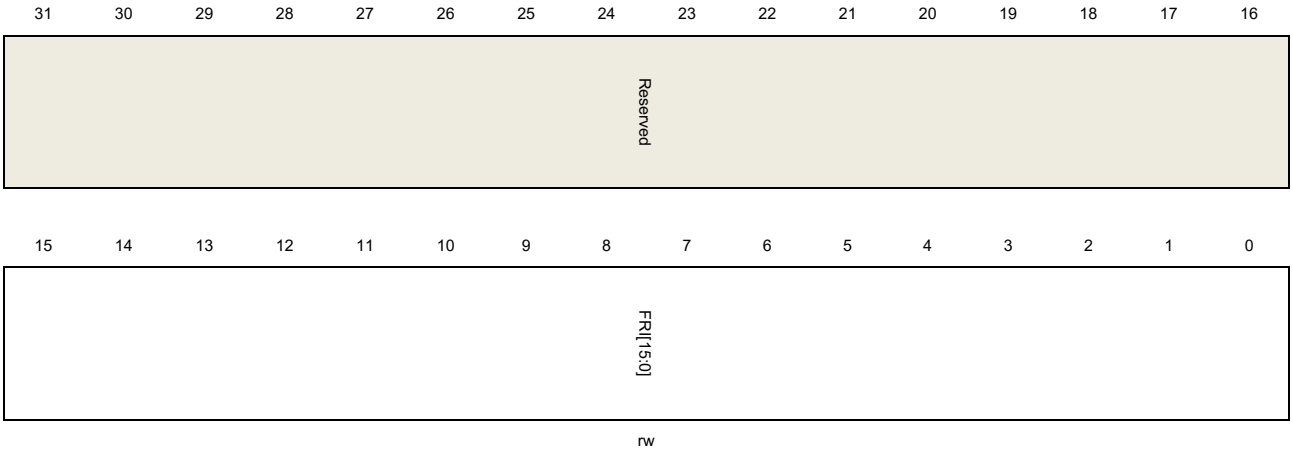
31.7.2. Host control and status registers

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)

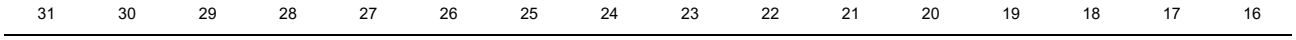


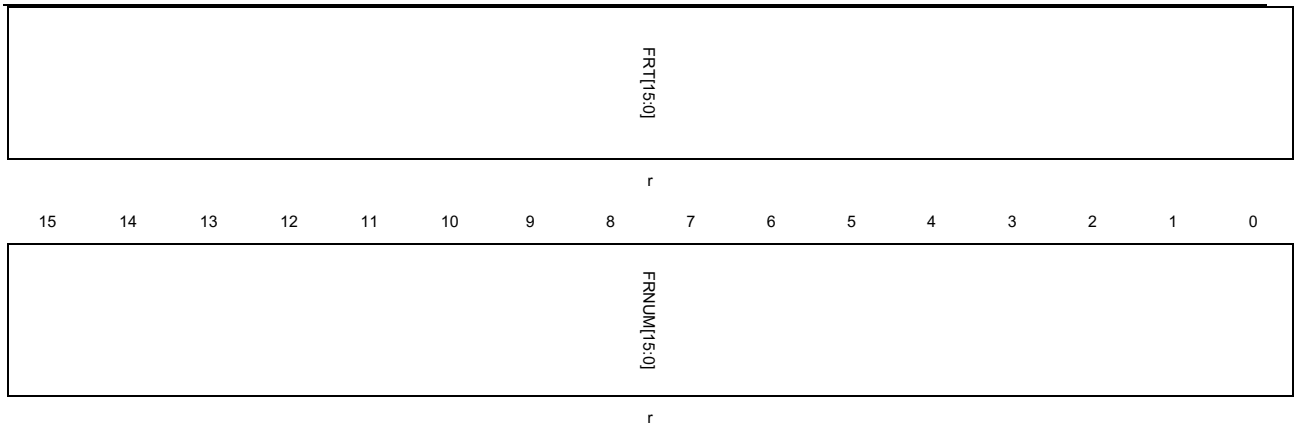
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	FRI[15:0]	Frame interval This value describes the frame time in terms of PHY clocks. Each time when port is enabled after a port reset operation, USBFS use a proper value according to the current speed, and software can write to 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)





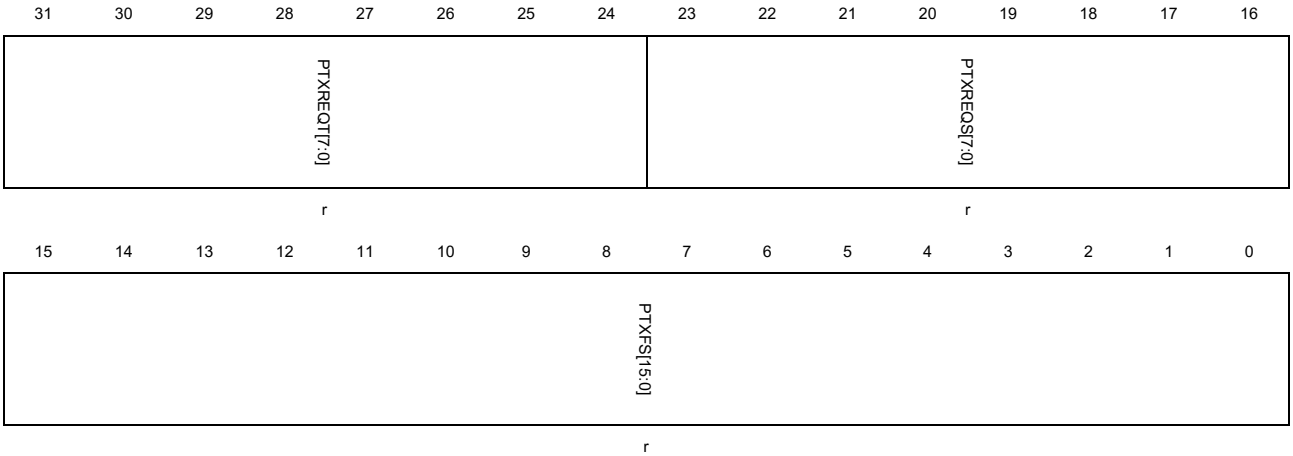
Bits	Fields	Descriptions
31:16	FRT[15:0]	Frame remaining time This field reports the remaining time of current frame in terms of PHY clocks.
15:0	FRNUM[15:0]	Frame number This field reports the frame number of current frame and returns to 0 after it reaches 0x3FFF.

Host periodic transmit FIFO/queue status register (USBFS_HPTFQSTAT)

Address offset: 0x0410
Reset value: 0x0008 0000

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.

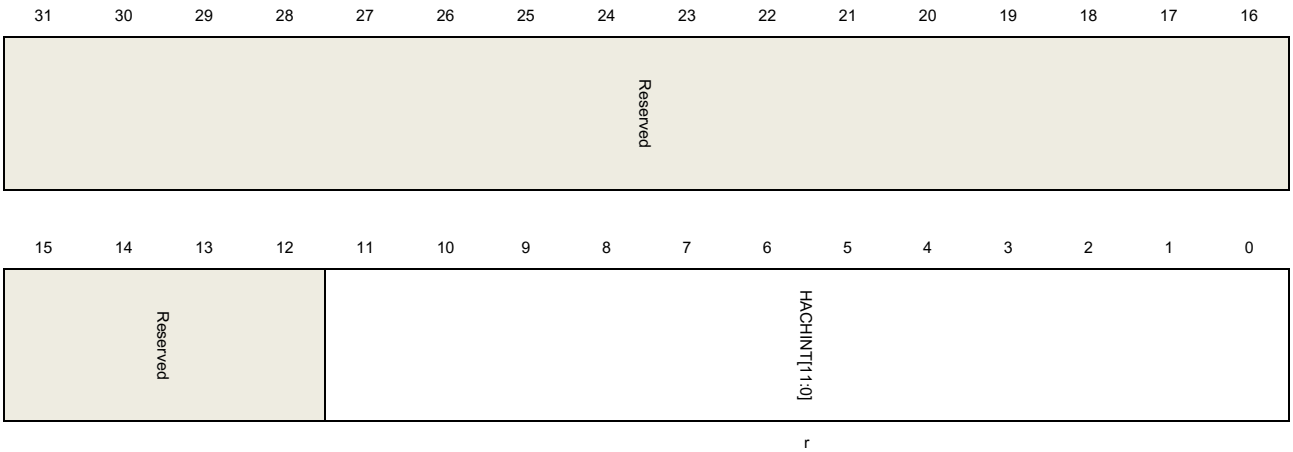
		Bits 30:27: Channel Number Bits 26:25: 00: IN/OUT token 01: Zero-length OUT packet 11: Channel halt request Bit 24: Terminate Flag, indicating last entry for selected channel.
23:16	PTXREQS[7:0]	Periodic Tx request queue space The remaining space of the periodic transmit request queue. 0: Request queue 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)



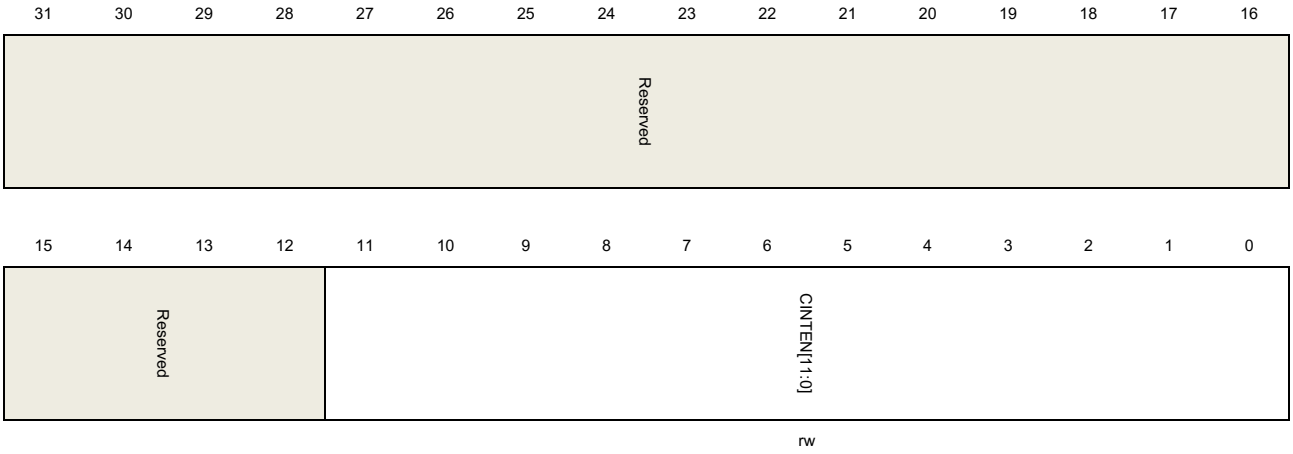
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	HACHINT[11:0]	Host all channel interrupts Each bit represents a channel: Bit 0 for channel 0, bit 11 for channel 11.

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)



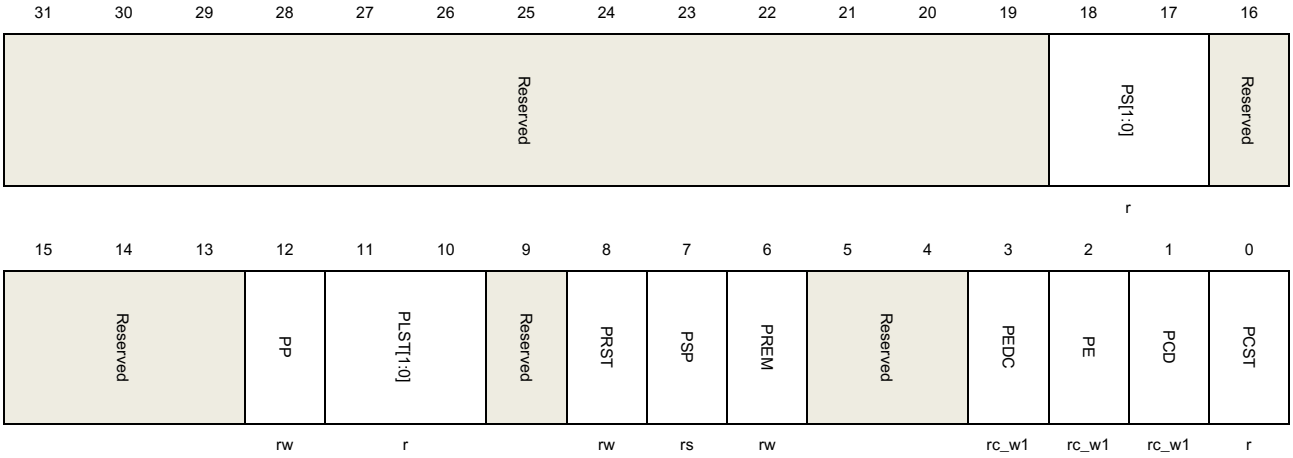
Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	CINTEN[11:0]	Channel interrupt enable 0: Disable channel-n interrupt 1: Enable channel-n interrupt Each bit represents a channel: Bit 0 for channel 0, bit 11 for channel 11.

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

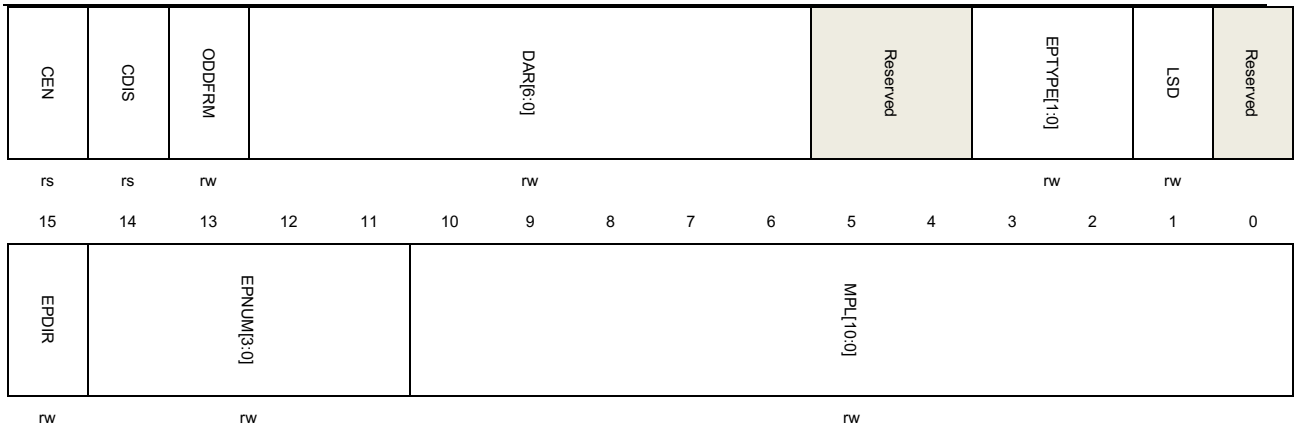
		stops sending SOF tokens. This bit can only be cleared by the following operations: PRST bit in this register is set by application PREM bit in this register is set A remote wakeup signal is detected A device disconnect is detected 0: Port is not in suspend state 1: Port is in suspend state
6	PREM	Port 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…11, where x = channel_number)

Address offset: 0x0500 + (channel_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



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.

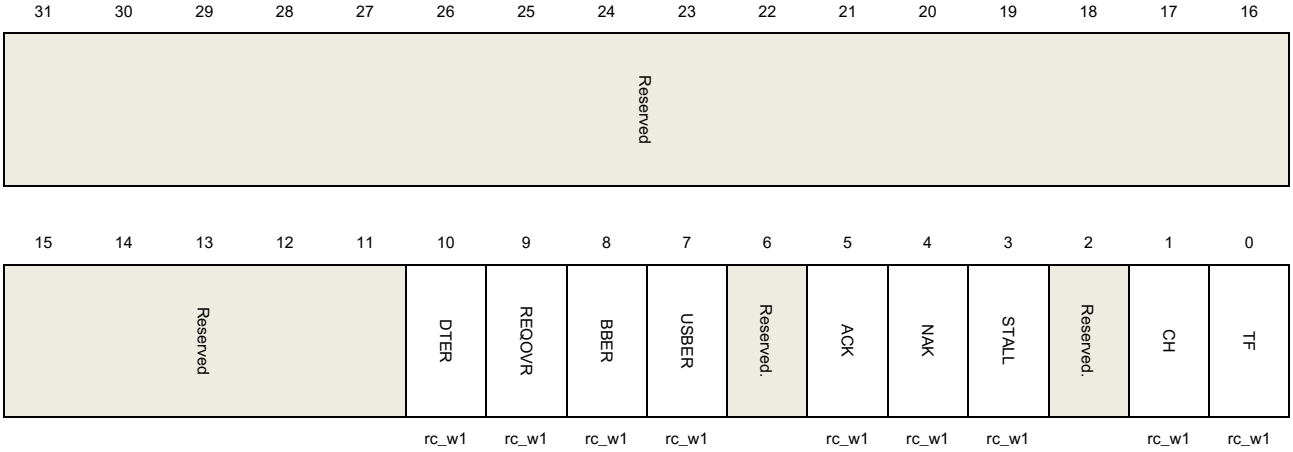
15	EPDIR	Endpoint direction The transfer direction of the endpoint that this channel wants to communicate with. 0: OUT 1: IN
14:11	EPNUM[3:0]	Endpoint number The number of the endpoint that this channel wants to communicate with.
10:0	MPL[10:0]	Maximum packet length The target endpoint's maximum packet length.

Host channel-x interrupt flag register (USBFS_HCHxINTF) (x = 0…11, 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.

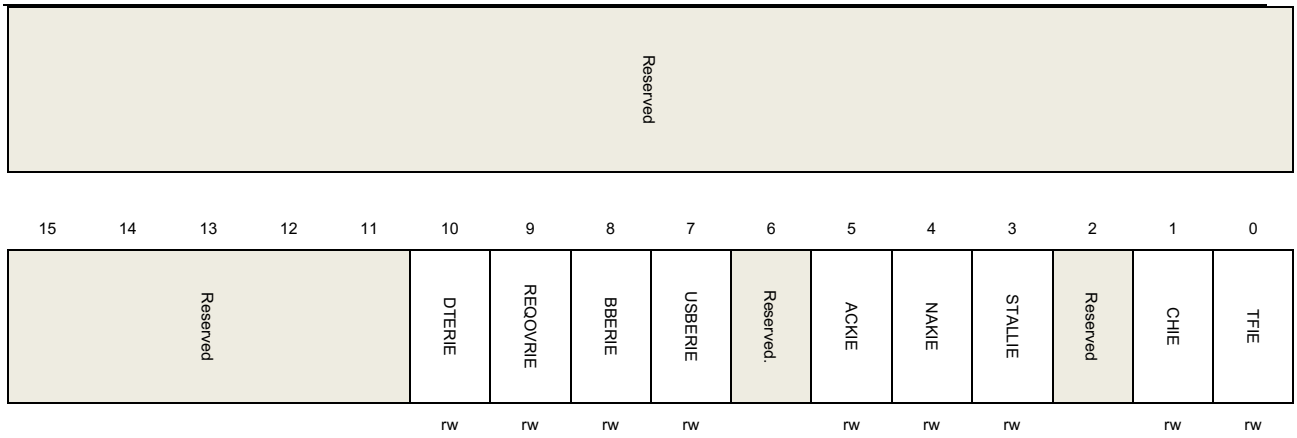
Host channel-x interrupt enable register (USBFS_HCHxINTEN) (x = 0…11, where x = channel number)

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)

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



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

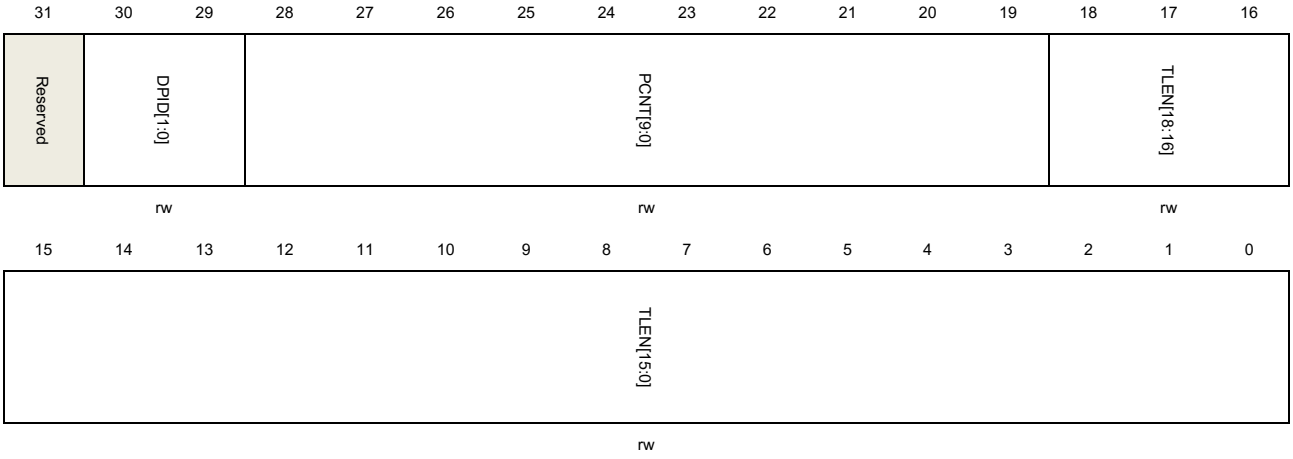
0: Disable channel halted interrupt
1: Enable channel halted interrupt

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 ... 11, 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.

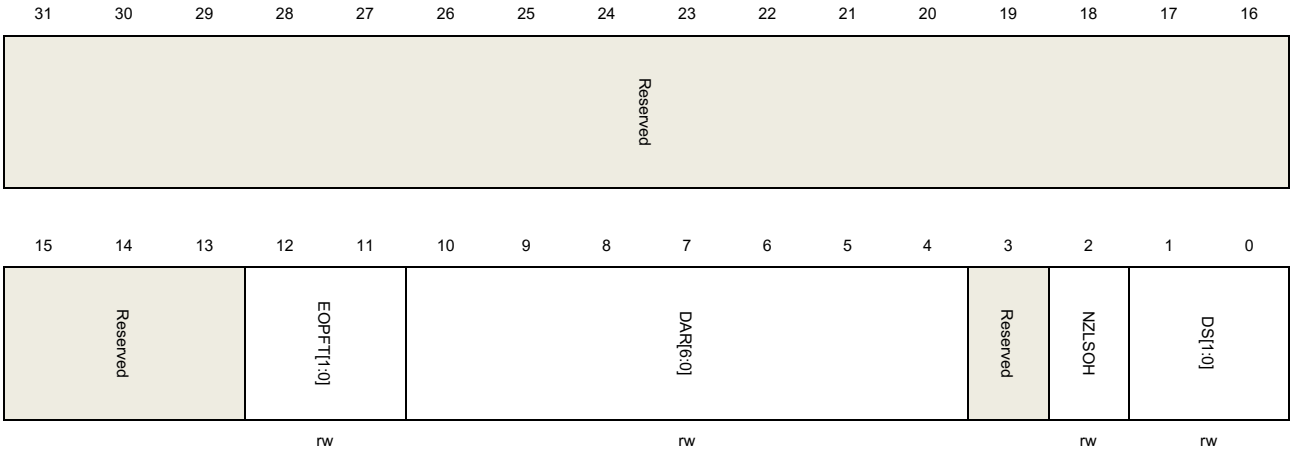
31.7.3. Device control and status registers

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

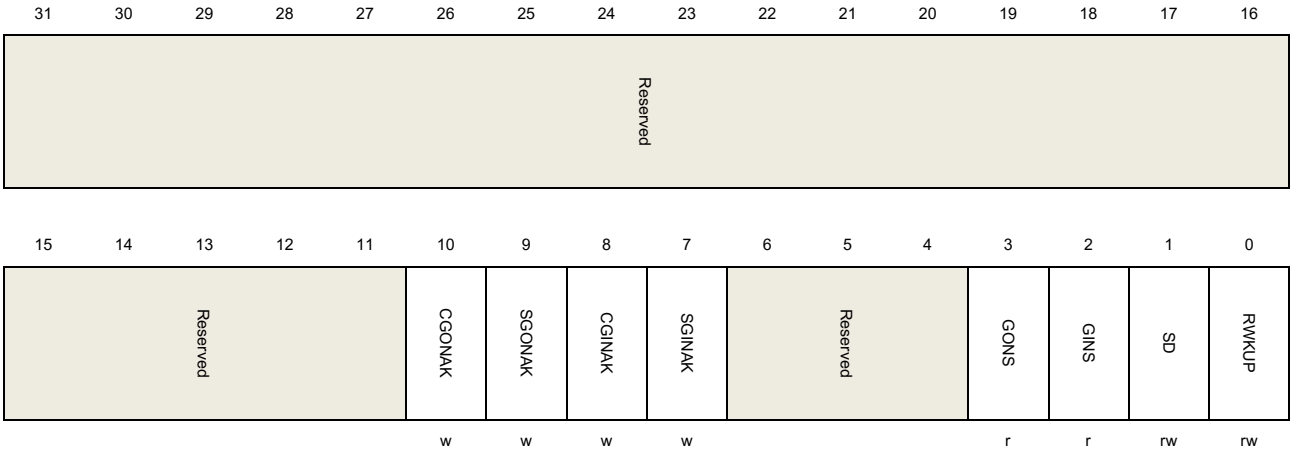
		11: 95% of the frame time
10:4	DAR[6:0]	Device address This field defines the USB device's address. USBFS uses this field to match with the incoming token's device address field. Software should program this field after receiving a Set Address command from USB host.
3	Reserved	Must be kept at reset value.
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
31:11	Reserved	Must be kept at reset value.
10	CGONAK	Clear global OUT NAK Software sets this bit to clear GONS bit in this register.

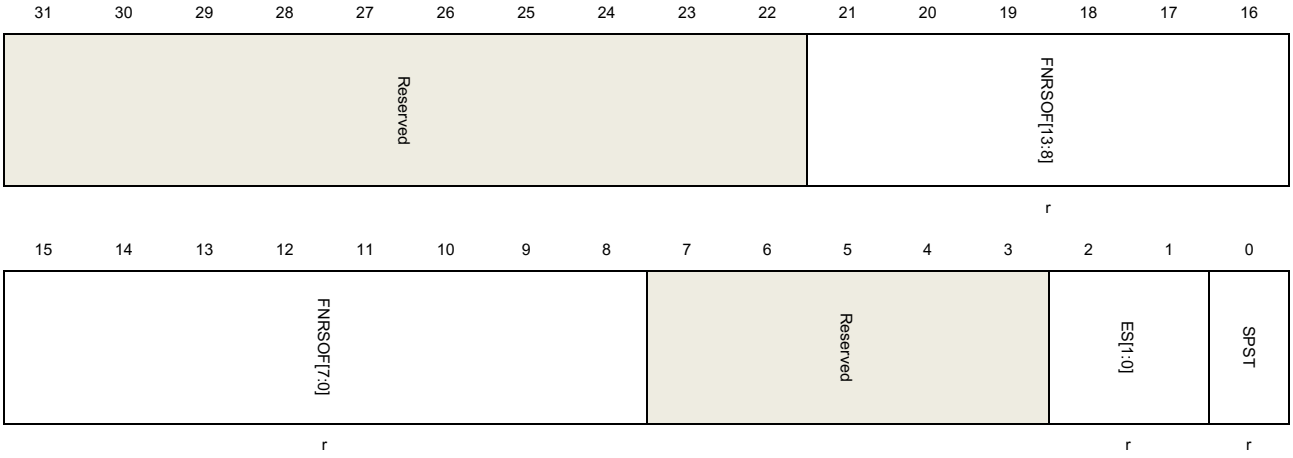
9	SGONAK	Set global OUT NAK Software sets this bit to set GONS bit in this register. When GONS bit is zero, setting this bit will also cause GONAK flag in USBFS_GINTF register triggered after a while. Software should clear the GONAK flag before writing this bit again.
8	CGINAK	Clear global IN NAK Software 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.

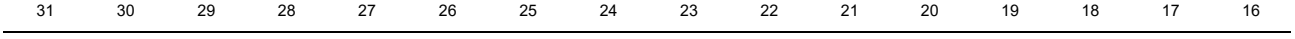
Device IN endpoint common interrupt enable register (USBFS_DIEPINTEN)

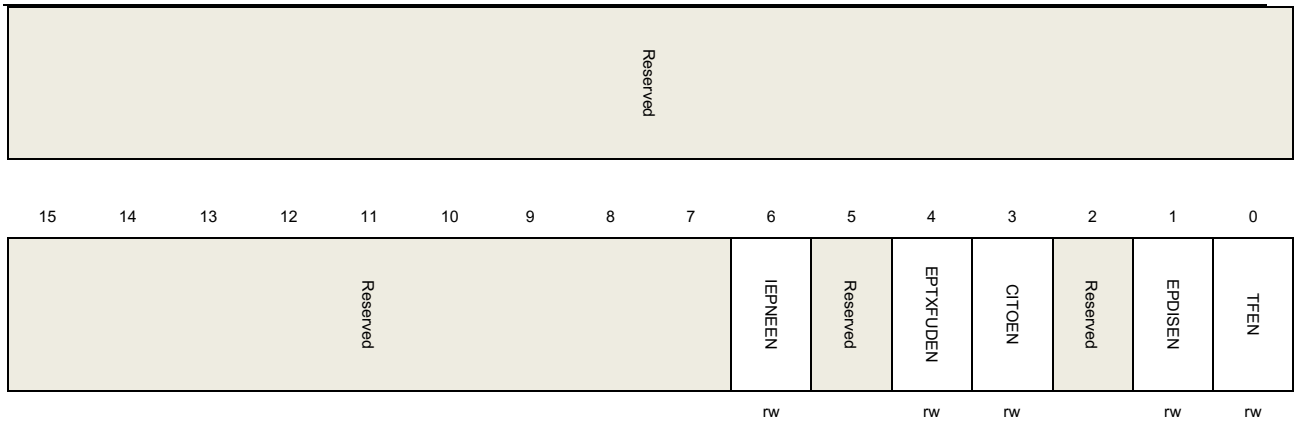
Address offset: 0x810

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)





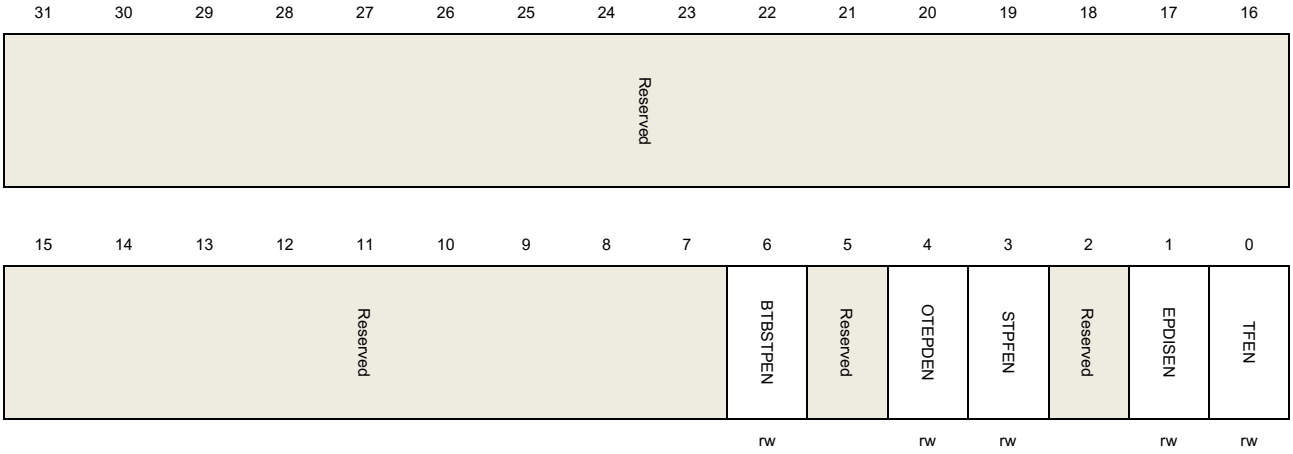
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	IEPNEEN	IN endpoint NAK effective interrupt enable bit 0: Disable IN endpoint NAK effective interrupt 1: Enable IN endpoint NAK effective interrupt
5	Reserved	Must be kept at reset value.
4	EPTXFUDEN	Endpoint Tx FIFO underrun interrupt 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	OTEPDEN	An OUT token is received when the endpoint disabled (Only for control OUT endpoint) interrupt enable bit 0: Disable endpoint disable to receive OUT token interrupt 1: Enable endpoint disable to receive OUT token 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

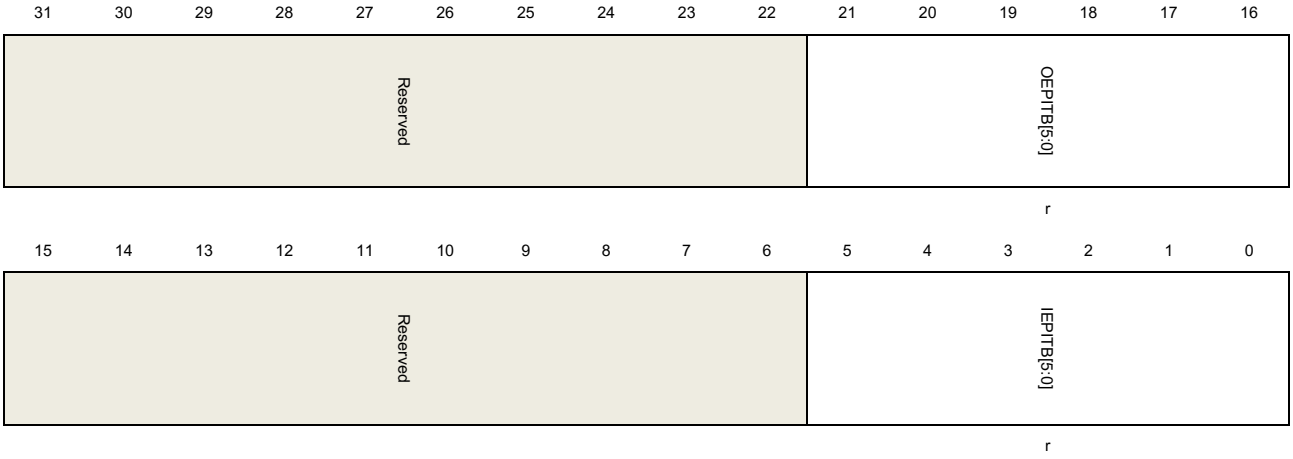
Device all endpoints interrupt register (USBFS_DAEPINT)

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)



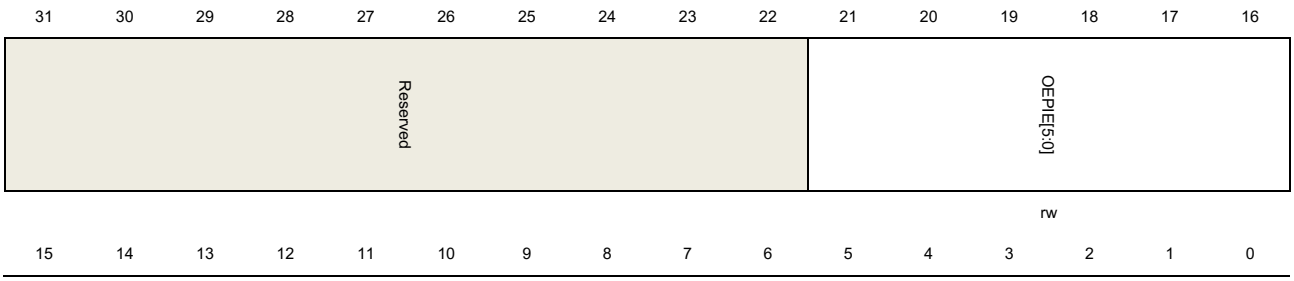
Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:16	OEPITB[5:0]	Device all OUT endpoint interrupt bits Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 21 for OUT endpoint 5.
15:6	Reserved	Must be kept at reset value.
5:0	IEPITB[5:0]	Device all IN endpoint interrupt bits Each bit represents an IN endpoint: Bit 0 for IN endpoint 0, bit 5 for IN endpoint 5.

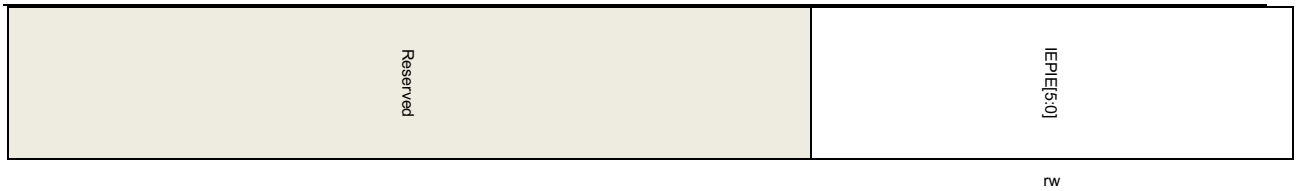
Device all endpoints interrupt enable register (USBFS_DAEPINTEN)

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)





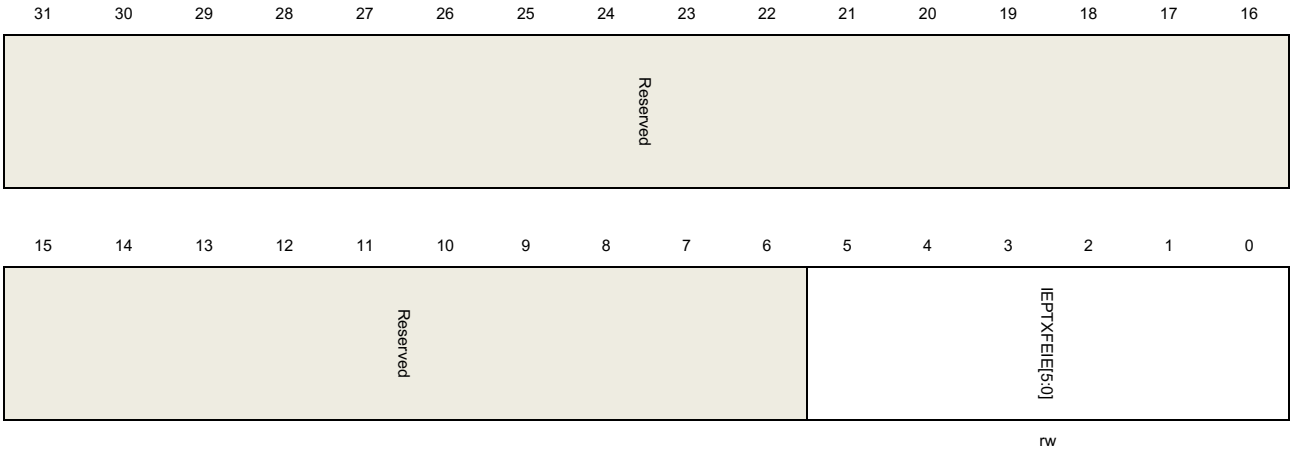
Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21:16	OEPIE[5:0]	Out endpoint interrupt enable 0: Disable OUT endpoint-n interrupt 1: Enable OUT endpoint-n interrupt Each bit represents an OUT endpoint: Bit 16 for OUT endpoint 0, bit 21 for OUT endpoint 5.
15:6	Reserved	Must be kept at reset value.
5:0	IEPIE[5:0]	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 5 for IN endpoint 5.

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

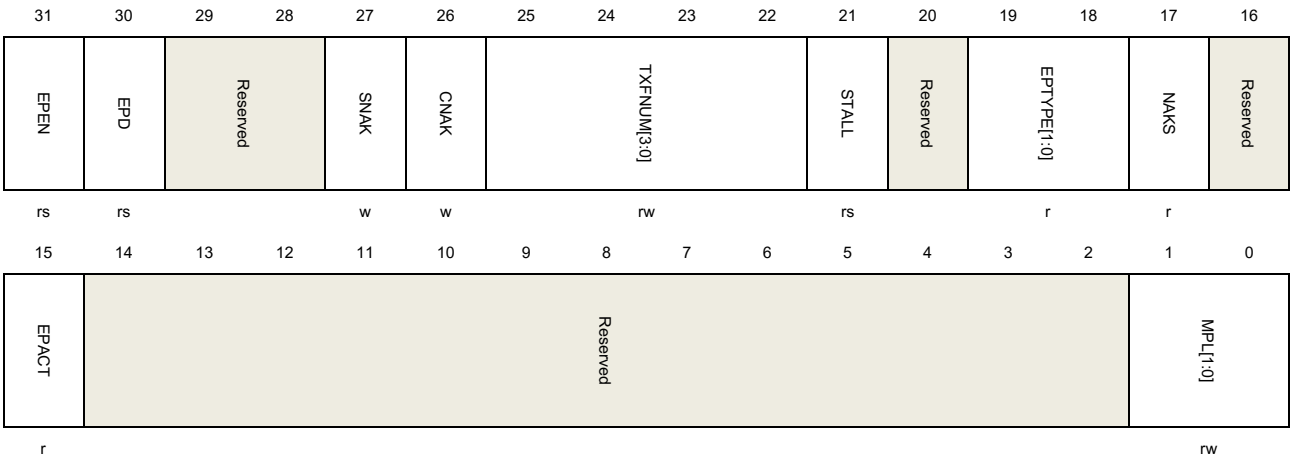
31:6	Reserved	Must be kept at reset value.
5:0	IEPTXFEIE[5:0]	IN endpoint Tx FIFO empty interrupt enable bits This field controls whether the TXFE bits in USBFS_DIEPxINTF registers are able to generate an endpoint interrupt bit in USBFS_DAEPINT register. Bit 0 for IN endpoint 0, bit 5 for IN endpoint 5 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.

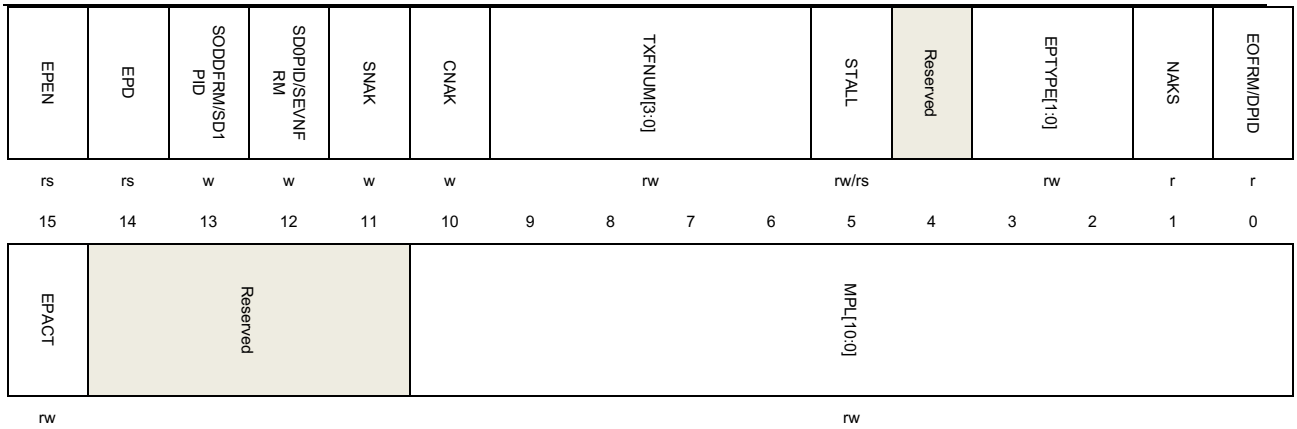
25:22	TXFNUM[3:0]	Tx FIFO number Defines the Tx FIFO number of IN endpoint 0.
21	STALL	STALL handshake Software can set this bit to make USBFS sends STALL handshake when receiving IN token. USBFS will clear this bit after a SETUP token is received on the corresponding OUT endpoint 0. This bit has a higher 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...5, 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



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.

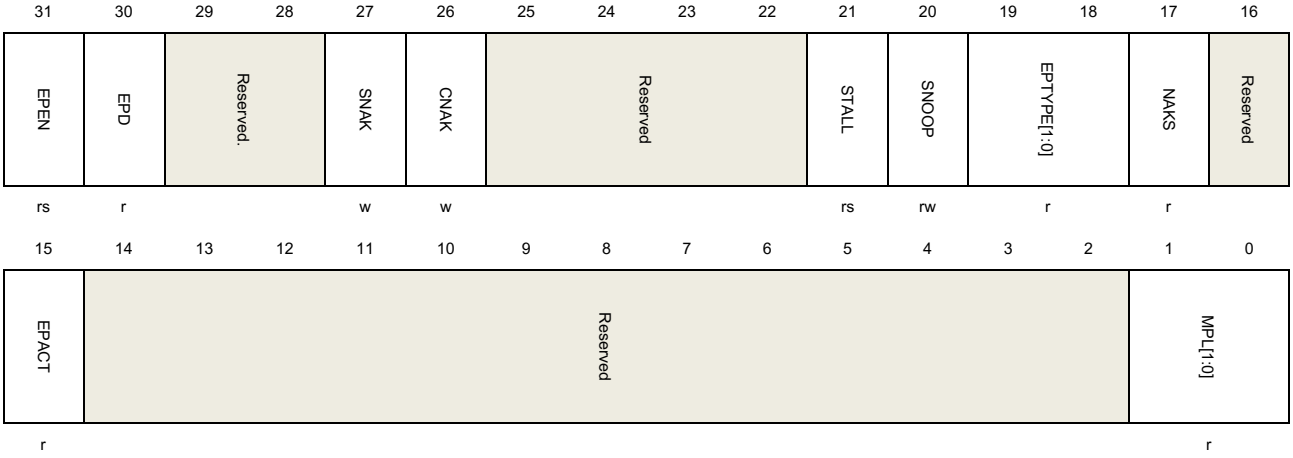
10:0	MPL[10:0]	This field defines the maximum packet length in bytes.

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

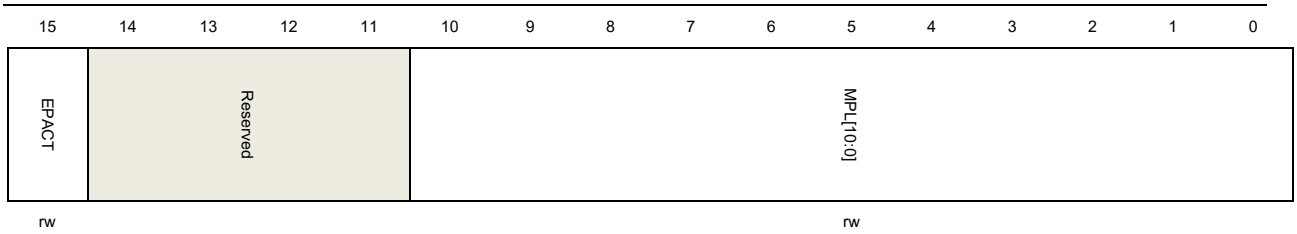
Device OUT endpoint-x control register (USBFS_DOEPxCTL) (x = 1...5, where x = endpoint_number)

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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/SD1PID	SEVNFRM/SD0PID	SNAK	CNAK	Reserved				STALL	SNOOP	EPTYPE[1:0]		NAKS	EOFRM/DPID
rs	rs	w	w	w	w					rw/rs	rw	rw		r	r



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:

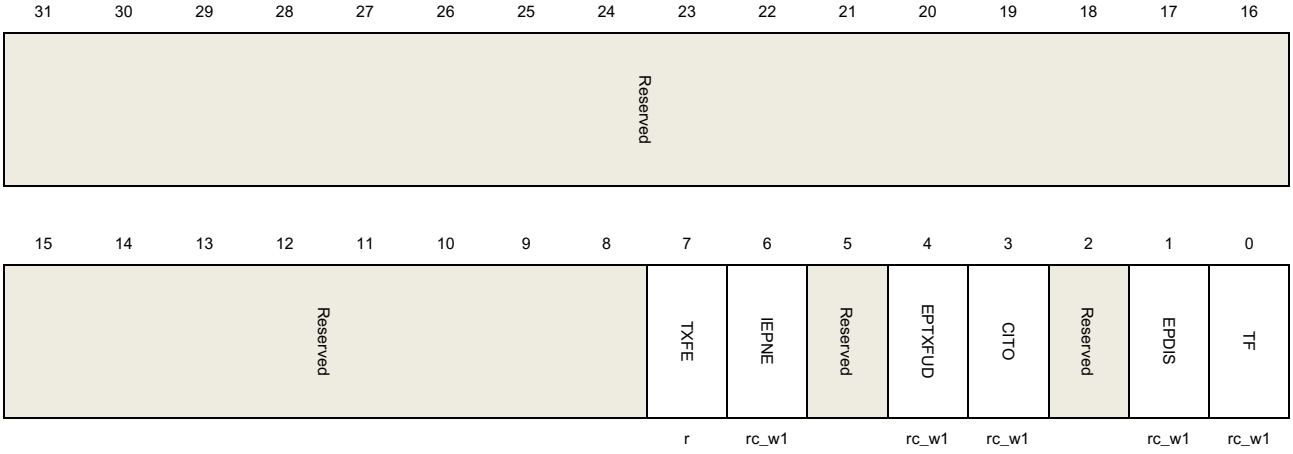
		Only software can clear this bit.
20	SNOOP	Snoop mode This bit controls the snoop mode of an OUT endpoint. In snoop mode, USBFS doesn't check the received data packet's CRC value. 0: Snoop mode disabled 1: Snoop mode enabled
19:18	EPTYPE[1:0]	Endpoint type This field defines the transfer type of this 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...5, 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.

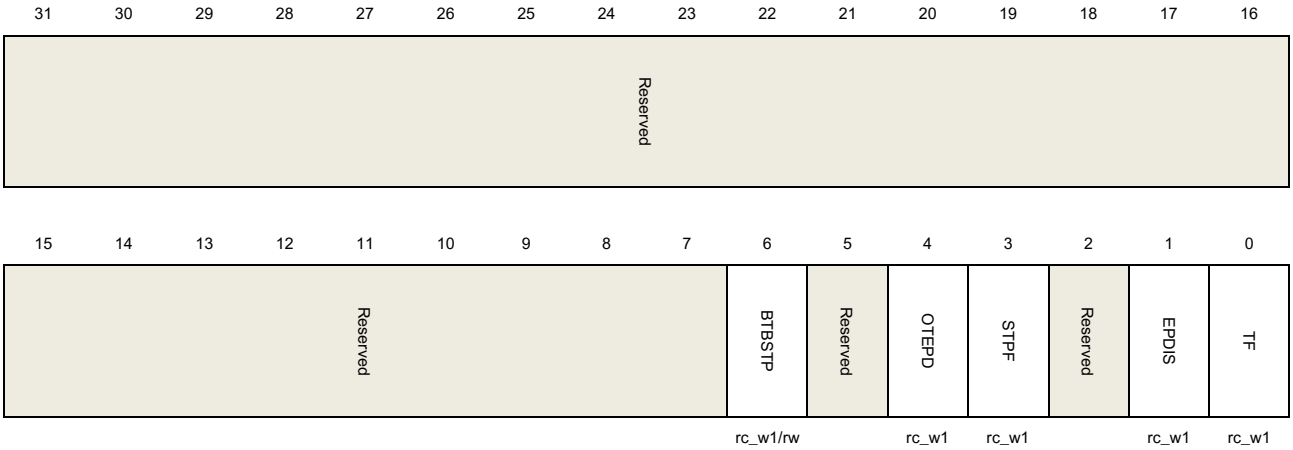
1	EPDIS	Endpoint disabled This flag is triggered when an endpoint is disabled by the software's request.
0	TF	Transfer finished This flag is triggered when all the IN transactions assigned to this endpoint have been finished.

Device OUT endpoint-x interrupt flag register (USBFS_DOEPxINTF) (x = 0...5, 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)



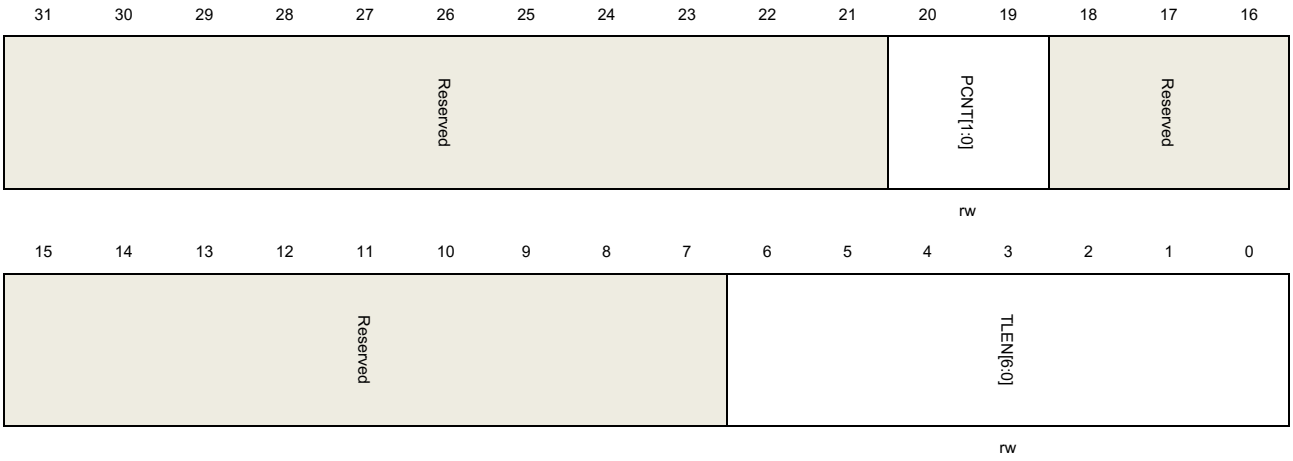
Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	BTBSTP	Back-to-back SETUP packets (Only for control OUT endpoint) This flag is triggered when a control out endpoint has received more than 3 back-to-back setup packets.
5	Reserved	Must be kept at reset value.
4	OTEPD	An OUT token is received when the endpoint disabled (Only for control OUT endpoint) When the endpoint is not enabled, an interrupt will be triggered if an OUT token is received.
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)



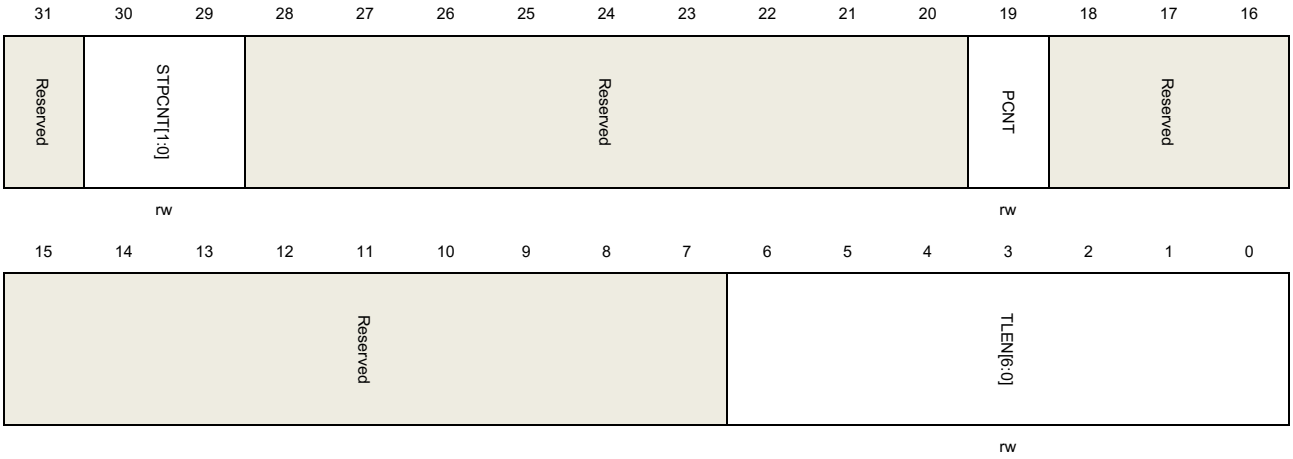
Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:19	PCNT[1:0]	Packet count The number of data packets desired to be transmitted in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each 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)



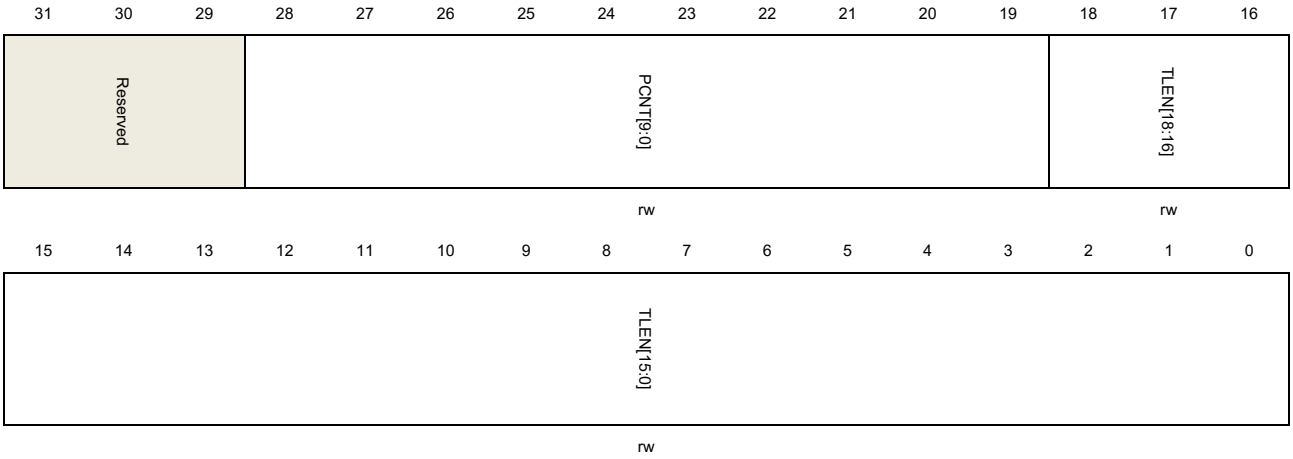
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	STPCNT[1:0]	SETUP packet count This field defines the maximum number of back-to-back SETUP packets this endpoint can accept. Program this field before setup transfers. Each time a back-to-back setup packet is received, USBFS decrease 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...5, where x = endpoint_number)

Address offset: 0x910 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

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

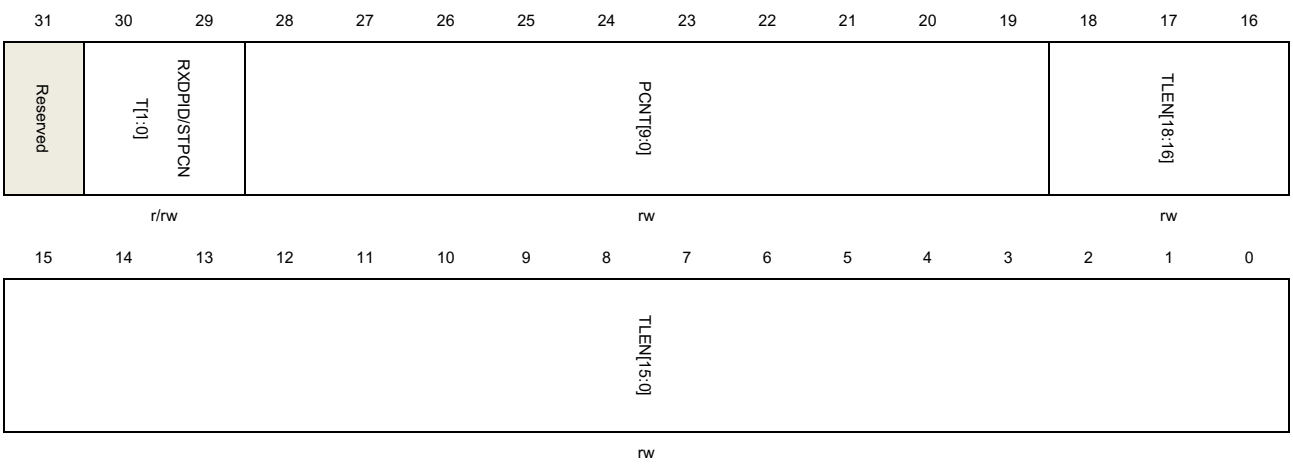


Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:19	PCNT[9:0]	Packet count The number of data packets desired to be transmitted in a transfer. Program this field before the endpoint is enabled. After the transfer starts, this field is decreased automatically by USBFS after each 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...5, where x = endpoint_number)

Address offset: 0x0B10 + (endpoint_number × 0x20)
Reset value: 0x0000 0000

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



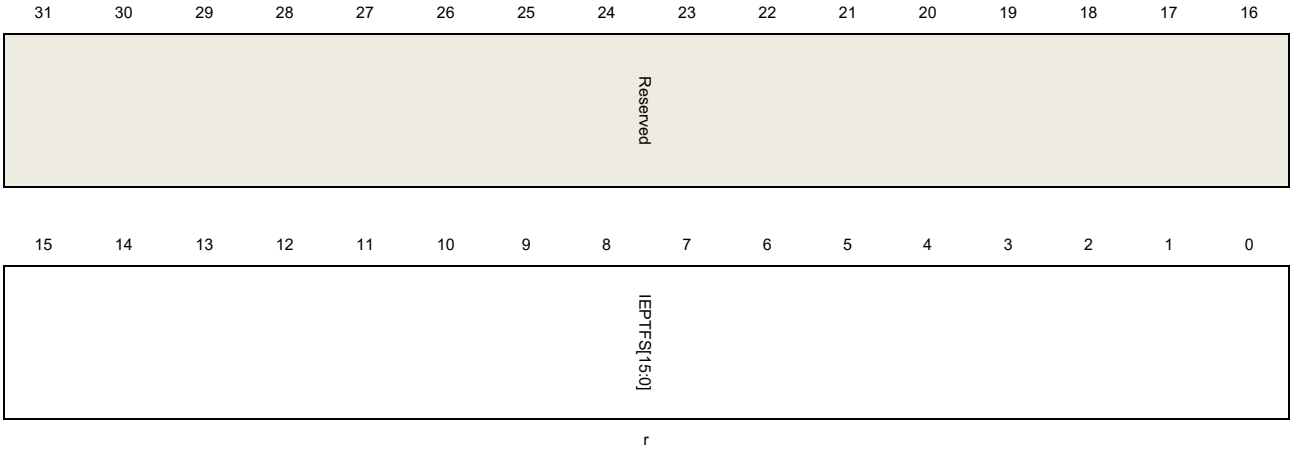
Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:29	RXDPID[1:0]	Received data PID (For isochronous OUT endpoints) This field saves the PID of the latest received data packet on this endpoint. 00: DATA0 10: DATA1 Others: Reserved
	STPCNT[1:0]	SETUP packet count (For control OUT Endpoints.) 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…5, where x = endpoint_number)

Address offset: 0x0918 + (endpoint_number × 0x20)
Reset value: 0x0000 0080

This register contains the information of each endpoint's Tx FIFO.

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

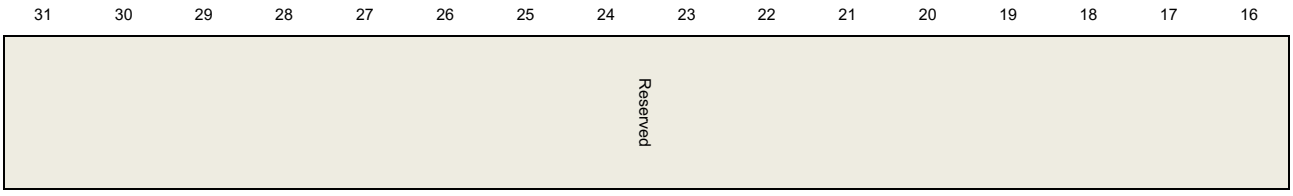


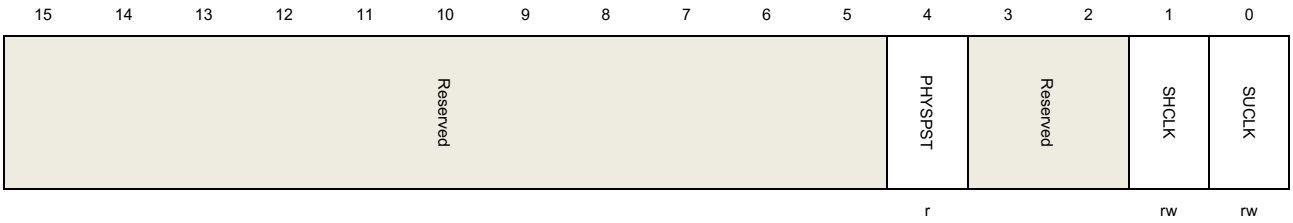
Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	IEPTFS[15:0]	IN endpoint's Tx FIFO space remaining IN endpoint's Tx FIFO space remaining in 32-bit words: 0: FIFO is full 1: 1 word available … n: n words available **Note:** This reset value is related to Tx FIFO number, and the reset situation is Tx FIFO0.

31.7.4. 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
31:5	Reserved	Must be kept at reset value.
4	PHYSPST	Suspended status The bit indicates whether the PHY is in a suspended state. 0: PHY is not in suspend state 1: PHY is in suspend state
3:2	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

32. Appendix

32.1. List of abbreviations used in register

Table 32-1. 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-	Software can only read to this bit. Writing 0 triggers the event but has no

abbreviations for registers	Descriptions
trigger (rt_w0)	effect on the bit value.
toggle (t)	Software can toggle this bit by writing 1. Writing 0 has no effect.

32.2. List of terms

Table 32-2. 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.

32.3. Available peripherals

For availability of peripherals and their number across all MCU series types, refer to the corresponding device data datasheet.

33. Revision history

Table 33-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.17, 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